



Project Title	Wide – Impact cyber Security Risk framework
Project Acronym	WISER
Grant Agreement No	653321
Instrument	Innovation Action
Thematic Priority	Cybersecurity, Privacy & Trust, Risk Management, Assurance Models
Start Date of Project	01.06.2015
Duration of Project	30 Months
Project Website	<a href="http://www.cyberwiser.eu">www.cyberwiser.eu</a>

## <D5.2 - WISER REAL-TIME ASSESSMENT INFRASTRUCTURE>

Work Package	WP 5, WISER Assessment & Decision Support
Lead Author (Org)	Antonio Álvarez (ATOS)
Contributing Author(s) (Org)	Susana González (ATOS), Rodrigo Díaz (ATOS), Carlos Hernán Arce (ATOS), Dawid Machnicki (ATOS), Ales Cernivec (XLAB), Anze Zitnik (XLAB), Atle Refsdal (SINTEF), Alberto Luca Biasibetti (AON), Sara Poidomani (AON), Jan Bastiaensens (ENERVALIS), Roberto Cascella (TRUST-IT)
Due Date	30.09.2016
Date	18.10.2016
Version	1.0

### Dissemination Level

<input checked="" type="checkbox"/>	PU: Public
<input type="checkbox"/>	PP: Restricted to other programme participants (including the Commission)
<input type="checkbox"/>	RE: Restricted to a group specified by the consortium (including the Commission)
<input type="checkbox"/>	CO: Confidential, only for members of the consortium (including the Commission)



## Versioning and contribution history

Version	Date	Author	Notes
0.0	26.07.2016	Susana González (ATOS), Antonio Álvarez (ATOS)	Table of Contents and production guidelines
0.1	24.08.2016	Antonio Álvarez (ATOS), Rodrigo Díaz (ATOS)	Contribution to Section 2
0.2	25.08.2016	Susana González (ATOS)	Contribution to Section 3.2, 3.3, 3.4.
0.3	25.08.2016	Antonio Álvarez (ATOS), Susana González (ATOS)	Contribution to Section 3.1. Refinement of Section 3.2, 3.3, 3.4
0.4	26.08.2016	Anže Žitnik (XLAB)	Contribution to section 4.1, detailed skeleton of 4.2, 4.3.
0.5	01.09.2016	Anže Žitnik (XLAB)	Contribution to section 4 and subsections.
0.6	05.09.2016	Anže Žitnik (XLAB)	Contribution to section 4.4.
0.7	07.09.2016	Antonio Álvarez (ATOS)	Refinement of sections 3.1.1 and 3.1.2.1. Section 4 layout
0.8	08.09.2016	Antonio Álvarez (ATOS)	Refinement of section 3.1.2.1
0.9	08.09.2016	Antonio Álvarez (ATOS)	Input to sections 3.1.2.4 and 3.1.3.1
0.10	08.09.2016	Antonio Álvarez (ATOS)	Input to section 3.1.4
0.11	12.09.2016	Antonio Álvarez (ATOS)	Input to sections 3.1.4 and 5.1
0.12	13.09.2016	Antonio Álvarez (ATOS)	Input to sections 3.1.3.2.1, 5.1.1. Refinement of section 3.4.1
0.13	13.09.2016	Antonio Álvarez (ATOS)	Input to sections 5.1.2, 5.1.3, 5.1.4, 5.1.5
0.14	13.09.2016	Anže Žitnik (XLAB)	Input to section 3.1.2.3, added appendix I (detailed DWH API description).
0.15	13.09.2016	Antonio Álvarez (ATOS)	Some edition guidelines added
0.16	13.09.2016	Antonio Álvarez (ATOS)	Refinement of sections 3.1.2.1, 3.1.3.1.2. Input for section

			3.1.3.2.1, 3.1.3.2.2
0.17	14.09.2016	Antonio Álvarez (ATOS)	Some edition guidelines added.
0.18	14.09.2016	Atle Refsdal (SINTEF)	Additions to section 5.1.2
0.19	15.09.2016	Atle Refsdal (SINTEF)	Minor updates in sections 2, 3.1.1, 3.1.2.4, 3.2.4
0.20	15.09.2016	Antonio Álvarez (ATOS)	Input to section 5.1.6
0.21	15.09.2016	Antonio Álvarez (ATOS)	Consolidation of sections 2 and 3.1.1. Refinement of Section 3.4.1
0.22	15.09.2016	Antonio Álvarez (ATOS)	Input to section 3.1.5 Refinement of sections 3.3.2, 4.3.2, 4.4.2. Layout of sections 4.5.1, 4.5.1.3, 4.5.1.4 and 5.1.2
0.23	15.09.2016	Atle Refsdal (SINTEF)	Update in section 3.2.4
0.24	15.09.2016	Alberto Luca Biasibetti (AON), Sara Poidomani (AON)	Refinements to sections 2, 3.1, 3.2
0.25	15.09.2016	Antonio Álvarez (ATOS)	Document layout
0.26	16.09.2016	Anže Žitnik (XLAB)	Minor changes in appendix I, addressed comments in section 3.2.3.
0.27	16.09.2016	Antonio Álvarez (ATOS), Anze Zitnik (XLAB)	Refinement of sections 3.1.3.2.1 and 5.1.5
0.28	19.09.2016	Susana González (ATOS)	Refinements in section 3.1.2.4, 3.2.1 and 3.2.4.
0.29	21.09.2016	Antonio Álvarez (ATOS), Anze Zitnik (XLAB)	Refinement of sections 3.1.1, 3.1.2.1 and 5.1.2
0.30	21.09.2016	Antonio Álvarez (ATOS), Anze Zitnik (XLAB)	Refinement of section
0.31	21.09.2016	Carlos Hernan Arce (ATOS)	Input to sections 3.2.5 and 3.2.6
0.32	21.09.2016	Aleš Černivec (XLAB)	Input to sections 5.2, 5.3 and 5.4.
0.33	22.09.2016	Antonio Álvarez (ATOS)	Input to section 1
0.34	22.09.2016	Atle Refsdal (SINTEF)	Minor update to section 3.2.7
0.35	22.09.2016	Antonio Álvarez (ATOS)	Refinement of sections 3.1.3.2.2 and 3.2.7.

			General document layout
0.36	22.09.2016	Dawid Machnicki (ATOS), Antonio Álvarez (ATOS)	Contributions to sections 5.2, 5.3, 5.4.
0.37	22.09.2016	Antonio Álvarez (ATOS)	Executive Summary. Refinement of sections 1.2 and 1.4.
0.38	22.09.2016	Antonio Álvarez (ATOS)	General layout
0.39	23.09.2016	Dawid Machnicki (ATOS), Susana González (ATOS), Antonio Álvarez (ATOS)	Refinement of sections 3.4.1 and 5.3.1.2 Version ready for internal review
0.40	09.10.2016	Jan Bastiaesens ( Enervalis ), Roberto Cascella (TRUST-IT)	First round of reviews
0.41	10.10.2016	Antonio Álvarez (ATOS), Susana González (ATOS)	Answers to first review
0.42	17.10.2016	Jan Bastiaensens (ENERVALIS), Roberto Cascella (TRUST-IT), Antonio Álvarez (ATOS)	Second round of reviews and answers
1.0	18.10.2016	Antonio Álvarez (ATOS)	Submission to the EC

## Disclaimer

**This document contains information which is proprietary to the WISER Consortium. Neither this document nor the information contained herein shall be used, duplicated or communicated by any means to a third party, in whole or parts, except with the prior consent of the WISER Consortium.**

## Table of Contents

---

Executive Summary .....	1
1 Introduction .....	2
1.1 Purpose and Scope .....	2
1.2 Structure of the document .....	2
1.3 Relationship to other project outcomes .....	3
1.4 Methodology .....	3
2 The WISER Real-time Assessment Infrastructure in a Nutshell .....	4
3 Risk Assessment Engine .....	10
3.1 Functional design.....	10
3.1.1 Overview.....	10
3.1.2 Inputs.....	13
3.1.3 Outputs .....	22
3.1.4 Triggering cases.....	35
3.1.5 Other considerations .....	40
3.2 Implementation .....	41
3.2.1 Indicator Value Generator .....	41
3.2.2 Triggering detector .....	45
3.2.3 DEXi Model Rules Executor .....	46
3.2.4 DEXi Model instantiator .....	48
3.2.5 R Model Rules Executor.....	50
3.2.6 R Model Instantiator .....	51
3.2.7 Aggregator.....	52
3.3 Deployment.....	53
3.3.1 Pre-requisites .....	53
3.3.2 Installation procedure .....	53
3.3.3 How to verify the installation.....	54
3.4 Operation .....	54
3.4.1 User Manual .....	55
3.4.2 Example of usage.....	56
4 Data Warehouse .....	57
4.1 Overview.....	57
4.2 Data modelling.....	57
4.2.1 Relational database.....	57
4.2.2 Document-based store .....	58
4.3 Implementation .....	58
4.3.1 Communication.....	59
4.3.2 Security.....	59
4.3.3 Notifications about data changes .....	59
4.4 Deployment.....	61
4.4.1 Pre-requisites .....	61
4.4.2 Installation Procedure.....	61
4.4.3 How to verify that the installation is correct.....	63
4.5 Operation .....	63
4.5.1 User Manual .....	63
5 Decision Support System .....	67
5.1 Functional design.....	67
5.1.1 Overview.....	70
5.1.2 Mitigation measures .....	71
5.1.3 Cost-benefit analysis .....	72
5.1.4 Societal impact analysis .....	74
5.1.5 Support to dashboard visualization .....	79

5.1.6	Triggering cases .....	81
5.2	Implementation .....	83
5.2.1	Presentation layer .....	83
5.2.2	Business layer .....	84
5.3	Deployment.....	85
5.3.1	Pre-requisites .....	85
5.3.2	Installation Procedure.....	86
5.3.3	How to verify the installation.....	86
5.4	Operation .....	86
5.4.1	User Manual .....	86
6	Summary and Conclusions .....	88
Appendix I	API Endpoints .....	90

## List of Tables

Table 1.	Structure of a qualitative report for CyberWISER-Essential .....	28
Table 2.	Structure of a qualitative report for CyberWISER-Plus.....	29
Table 3.	Structure of a quantitative report for CyberWISER-Essential.....	31
Table 4.	Structure of a quantitative report for CyberWISER-Plus.....	32
Table 5.	Some examples of mitigation measures .....	71
Table 6.	Societal impact questionnaire .....	76
Table 7.	Societal impact evaluation. Example of configuration of the utility functions and weights to process the answers to the questionnaire. ....	77

## List of Figures

Figure 1.	WISER abstraction levels .....	5
Figure 2.	Interplay of the Real-Time Assessment Infrastructure with the rest of the WISER Framework in CyberWISER-Light.....	7
Figure 3.	Interplay of the Real-Time Assessment Infrastructure with the rest of the WISER Framework in CyberWISER-Essential .....	8
Figure 4.	Interplay of the Real-Time Assessment infrastructure with the rest of the framework in CyberWISER-Plus.....	9
Figure 5.	Risk Assessment Engine internal detail for CyberWISER-Light.....	12
Figure 6.	Risk Assessment Engine internal detail for CyberWISER-Essential and CyberWISER-Plus	13
Figure 7.	Inputs and output of the Risk Assessment Engine .....	14
Figure 8.	Questionnaire interface in CyberWISER-Light .....	15
Figure 9.	Target importance in terms of confidentiality, integrity and availability .....	16
Figure 10.	Configuration questionnaire, target setup and risk model selection user interface (CyberWISER-Essential).....	17
Figure 11.	Interface to specify elements of the target infrastructure .....	18
Figure 12.	Editing the targets within the client infrastructure (CyberWISER-Essential) .....	19
Figure 13.	Configuration questionnaire, target setup and risk model selection user interface (CyberWISER-Plus) .....	20
Figure 14.	Editing targets within client infrastructure (CyberWISER-Plus).....	20
Figure 15.	Risk model selection by means of the configuration dashboard .....	22
Figure 16.	Summary of the risk report produced by CyberWISER-Light .....	24
Figure 17.	CyberWISER-Light PDF report. Page sample.....	25
Figure 18.	CyberWISER-Light vulnerability report shown on the user interface .....	26
Figure 19.	CyberWISER-Light vulnerability report. Page sample.....	27
Figure 20.	CyberWISER-Essential risk report sample.....	30
Figure 21.	CyberWISER-Plus risk report sample.....	30
Figure 22.	Example of quantitative risk report .....	33
Figure 23.	General view of the dashboard with widgets for the current qualitative and quantitative	

---

assessments and the lists of previous assessments .....	34
Figure 24. Risk Assessment Engine operation case 1. Triggering demanded by the user .....	35
Figure 25. Risk Assessment Engine operation case 2. Change in model .....	37
Figure 26. Risk Assessment Engine operation case 3. Change in business variables .....	38
Figure 27. Risk Assessment Engine operation case 4. Change in monitoring variables .....	39
Figure 28. Risk Assessment Engine operation case 5. Change in testing variables .....	40
Figure 29. Example of risk report visualization in CyberWISER Dashboard. ....	56
Figure 30. Relational database model in WISER Data Warehouse .....	60
Figure 31. Decision Support System internal detail (CyberWISER-Light) .....	68
Figure 32. Decision Support System internal detail (CyberWISER-Essential) .....	69
Figure 33. Decision Support System internal detail (CyberWISER-Plus) .....	70
Figure 34. Suggestion of mitigation measures. Short and detailed description of the measure .....	72
Figure 35. Decision Support System. Cost-benefit analysis. Filling out the template .....	73
Figure 36. Decision Support System. Cost-benefit analysis. Result inspection .....	74
Figure 37. Decision Support System. Societal impact analysis. Questionnaire .....	78
Figure 38. Decision Support System. Societal impact analysis. Results .....	78
Figure 39. Older risk report visualization feature .....	79
Figure 40. Older report visualization. Report detail .....	80
Figure 41. Decision Support System Triggering Case 1: the user requests information about mitigation measures.....	81
Figure 42. Decision Support System Triggering Case 2: cost-benefit analysis of mitigation measures	82
Figure 43. Decision Support System Triggering Case 3. Comparison of mitigation measures in cost- benefit terms. ....	82
Figure 44. Decision Support System Triggering Case 4: Societal Impact Analysis of risks .....	83
Figure 45: DSS presentation layer.....	87
Figure 46: Widgets in edit mode. ....	87

---

## Executive Summary

---

This deliverable documents the implementation, deployment and operation of the Real-Time Assessment Infrastructure. It is composed of three modules, namely: 1) The Risk Assessment Engine; 2) The Data Warehouse and 3) The Decision Support System.

The document provides an overview of the modules and then goes deeper into the description of each of them and how they interplay with the rest of the Framework.

The functional design provided in Deliverable D2.3 and Deliverable D5.1 is updated to account for feedbacks of three different sources:

1) the needs detected when the implementation phase started in M10 – March 2016. Most of them have to do with changes in the data model, which is properly updated in this document, splitting the internal flow of the Risk Assessment Engine into two threads: for the qualitative and quantitative models respectively, and some minor modifications in the design of the Decision Support System to better accommodate the cost-benefit analysis and societal impact analysis features.

2) the early outcomes of Task 2.3 on solution visioning, which led to a more detailed definition of the dashboard for the sake of a more user-centric approach for the implementation of the technical solution of the user interface and

3) the feedback obtained from the WP6 Full-Scale Pilots. The functional design, as already discussed in previous deliverables, is documented in the light of the definition of the WISER Service Portfolio, with its three products, namely: CyberWISER-Light, CyberWISER-Essential and CyberWISER-Plus.

Relevant technical details are documented concerning the implementation of each module so the reader can gain good insights about the approach adopted to materialize what was envisioned during the requirement and design phases. Subsequently, the deployment is presented using the same sequence: 1) Pre-requirements, 2) Installation and 3) Verification of the installation. Finally, some remarks are made regarding the actual operation of each of the three elements.

The main takeaways of this document are the following:

- The final version of the Real-Time Assessment Infrastructure: general overview of the modules, interfaces and data exchanged among them.
- Functional design, implementation, deployment and operation of the Risk Assessment Engine, Data Warehouse, and Decision Support System.

WP5 activities, as scheduled in the DoA document will finish in M20 (January 2017). D5.2 is the last deliverable of the work package, and the remaining time will be used to strengthen and upgrade the first stable version of the Real Time Assessment Infrastructure documented in this deliverable.



---

## 1 Introduction

---

### 1.1 Purpose and Scope

Deliverable D5.2 'WISER Real-Time Assessment Infrastructure' covers the details of the implementation, deployment, and operation of the components providing the real-time feature of the WISER Framework. The goal is to report and document the work performed from M10 (March 2016) to M16 (September 2016) in the context of WP5.

The WISER Real-Time Assessment Infrastructure allows shifting away from a purely technical assessment of cyber risk, only considering the information obtained from the target infrastructure by means of testing and monitoring techniques, to an assessment which takes into consideration the business aspect linked to the threatened elements of such infrastructure.

This deliverable gives an overview of the technical solution provided for this important part of the framework (it is the brain of WISER as a matter of fact, providing the needed intelligence to consider the risk from the business point of view), included an in-depth presentation of each one of the components, namely: the Risk Assessment Engine (covered in Task 5.1), the Data Warehouse (addressed in Task 5.2) and Decision Support System (which is the subject of Task 5.3). Within the sections devoted to each component, the functional design is summarised, including updates produced during the period M10-M16, and then the implementation, the deployment and the operation are further explained.

The Risk Assessment Engine uses the input of the configuration, testing, modelling and monitoring modules to calculate the cyber risk assessment. The outcome of the assessment is sent to the Decision Support System to inform the user and support him when deciding the most appropriate countermeasures to implement in order to diminish such risk. In addition, the Data Warehouse is the central storage element, decisive to perform the exchange of data among the different components of the framework.

### 1.2 Structure of the document

The document follows the structure explained below:

- Section 1 introduces the document and its context as well as the relationship of this deliverable with other project outcomes. Finally, the methodology adopted to produce this deliverable is presented.
- Section 2 offers an overview of the real-time assessment infrastructure, describing briefly its components, the inputs received, and the outputs produced. This is done for the three products of the WISER Portfolio: CyberWISER-Light, CyberWISER-Essential and CyberWISER-Plus. Diagrams are provided to illustrate this explanation.
- Section 3 presents the Risk Assessment Engine, updating the functional design and reporting on the implementation, the deployment, and the operation of this module.
- Section 4 presents the details about the Data Warehouse, addressing its main characteristics, the data model solution adopted, its implementation, deployment and finally relevant insights about its operation.
- Section 5 presents the Decision Support System in detail, updating the functional design, reporting about its internal implementation, addressing its deployment and giving relevant insights about its operation.
- Section 6 concludes the document.
- Finally, an appendix contains the detailed description of the API endpoint of the Data Warehouse

---

### 1.3 Relationship to other project outcomes

The delivery of this part of the WISER Framework is the main goal towards which all the efforts of WP5 are oriented. Thus, providing this set of functionalities entails an important breakthrough within the project timeline. This will impact WP2, which will document the integration of WP5 outcomes with the rest of the WISER framework in deliverable D2.4 (due November 2016), WP7, which will take care of the market validation of the results of the technical work packages, and WP8, which will use this output to feed the activities of dissemination, communication, stakeholder engagement, community building, exploitation, and business models elaboration.

The work, herein presented, has been accomplished from M10 (March 2016) to M16 (September 2016) in close coordination with work packages WP2 (Coordination of design and solution visioning), WP3 (interfaces with modelling module), WP4 (interfaces with sensors and monitoring modules) and WP6 (continuous bidirectional feedback flow with Full-Scale Pilots taskforces). Deliverable D5.2 provides an update of the final design of the Real-Time Assessment Infrastructure which is reported in D5.1 (released in M9, February 2016). These updates come from three different sources:

- 1) the needs detected when the implementation phase started in M10 – March 2016 –, Most of them have to do with changes in the data model, which is properly updated in this document, splitting the internal flow of the Risk Assessment Engine into two threads: for the qualitative and quantitative models respectively, and some minor modifications in the design of the Decision Support System to better accommodate the cost-benefit analysis and societal impact analysis features.
- 2) the early outcomes of Task 2.3 on solution visioning, which led to a more detailed definition of the dashboard for the sake of a more user-centric approach for the implementation of the technical solution of the user interface.
- 3) the feedback obtained from the WP6 Full-Scale Pilots activities.

The implementation of the Real-Time Assessment Engine has been carried out in the context of the WP5 activities, and is coordinated from Task 2.2, which takes care of the integration with the outcomes coming from the WP3 (risk models), WP4 (monitoring infrastructure components) and WP2 (cross components such as configuration and user management). As with the functional design, a continuous communication flow takes place with WP6, to consider in the implementation relevant insights coming from the Full-Scale Pilots.

D5.2 leverages deliverables D5.1 (Design of the WISER Real-Time Assessment Infrastructure) and D2.3 (Framework Design, Final Version) to define the final design of the WISER real-time platform. D2.3 covers the design of WISER as a whole and offers a high-level approach of the final design, while D5.1 puts the focus on the real-time assessment infrastructure and the specific design of this module. Following a logic timeline, D5.2 is produced in parallel with D4.2 and their delivery entails an important breakthrough in the project. Despite being the last deliverable to be submitted within WP5, the work package will still last four more months, were refinements and upgrades of the implementation will take place. D2.4, to be released in M18 (November 2016) will document the complete integration of the different parts of the Framework coming from WP2, WP3, WP4 and WP5 activities.

Finally, the technical solution provided for the WISER challenge is being validated thanks to a continuous feedback with WP7 activities and is a major input for all the WP8 activities: dissemination, communication, community building, exploitation, business models, etc.

### 1.4 Methodology

A detailed implementation plan, with a thorough breakdown of tasks, responsible partners, and estimated deadlines was elaborated in M10 (March 2016). The implementation started effectively on 01/04/2016. From April to August, fortnightly follow-up calls were scheduled to monitor the progress of the work, discuss likely problems, and define following steps with concrete action points. From August onwards the frequency was switched to weekly. The workplan was continuously adjusted to account for the actual progress of the work. In coordination with WP1, risks have been monitored periodically.

An appropriate implementation environment has been put in place. Specific code repository (GITLab<sup>1</sup>), project management tool (Trac<sup>2</sup>) have been made available to ease work. Internal rules to be fulfilled by all developers have been defined (with a special focus on versioning, deployment, and regression tests). For communication purposes, a specific mailing list<sup>3</sup> has been made available to ease an agile communication flow.

As for the elaboration of the deliverable, the approach is the same for each element of the Real-Time Assessment Infrastructure:

- Provision of an update of the functional design.
- Explanation of the technical implementation of the internal components.
- Documentation of the deployment method, following always the same sequence: 1) Identification of pre-requisites, 2) Installation and 3) Verification of the installation.
- High-level description of the actual operation of each element.

## 2 The WISER Real-time Assessment Infrastructure in a Nutshell

The Real-Time Assessment Infrastructure provides a real-time assessment of the risk faced by the client. To do this, it leverages inputs from both the technical and the business aspects of an organisation. By combining these two aspects and putting the result of the evaluation in the context of the company business, the Real-Time Assessment Infrastructure delivers a risk assessment report that can be understood by the company's top management positions. This is of paramount importance since these are the key people in the company who have the responsibility to decide how to tackle cyber threats and what kind of mitigation measures should be adopted. In a nutshell, the Real-Time Assessment Infrastructure:

- Leverages the information obtained in real-time from the cyber climate (events, alarms, vulnerabilities).
- Correlates it to the business value of the services provided by the infrastructure elements and the business and ICT profile of the company by means of modelling techniques.
- Obtains an evaluation of the risk faced by the client company which considers not only the ICT level, but also the business level. It presents the information in a meaningful way in order to support the decision-making and mitigation process.

From a conceptual point of view, the Real-Time Assessment Infrastructure is placed on top of the Monitoring Infrastructure and provides an extra abstraction level which allows a cyber risk assessment understandable for top level managers, easing the decision-making process. This is shown in Figure 1, which already appears in D4.1 and D5.1.

---

<sup>1</sup> <https://git-scm.com>

<sup>2</sup> <https://trac.edgewall.org>

<sup>3</sup> [development-wiser@lists.atosresearch.eu](mailto:development-wiser@lists.atosresearch.eu)

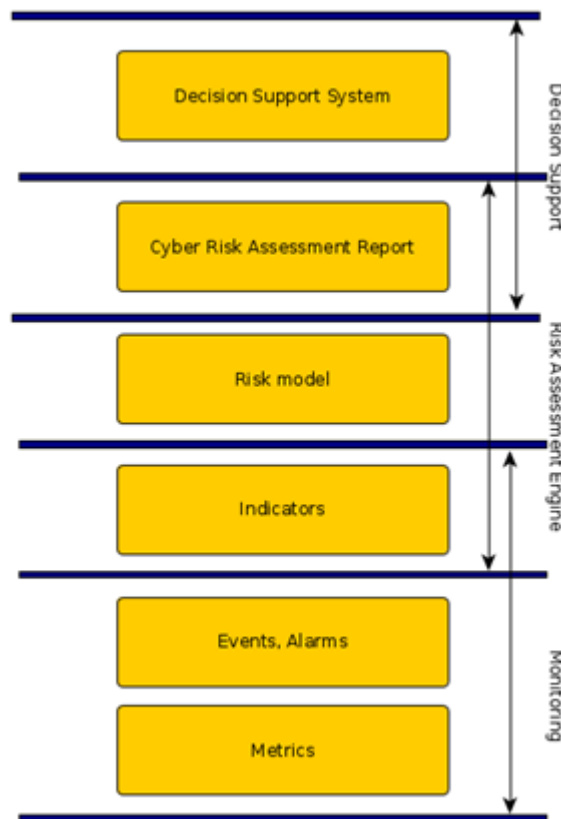


Figure 1. WISER abstraction levels

Deliverable D4.1 describes how, at the level of the monitoring infrastructure, the sensors and vulnerability scanners deployed within the client target infrastructure collect very useful data, used to describe the cyber climate in which the infrastructure and its elements are operating. This information is used to obtain metrics. Metrics are measures/quantifications referred to certain system properties being important for the purpose of a cyber security risk assessment. The next level of analysis is used to obtain events and alarms. Events are generated from the metrics, and alarms are reports generated from the collected events, describing situations taking place in the infrastructure being relevant for its cyber resilience. This information is enough to monitor the infrastructure from a purely technical point of view and to allow a quick reaction in case of a major event.

However, this is not enough for WISER purposes. The goal is to provide a cyber risk assessment report thought and addressed to top management profiles, taking into account the business implications of the cyber climate in which the company ICT infrastructure operates. To obtain this report, the following sequence takes place:

- The events and alarms have to be transformed into something understandable by the algorithm producing the cyber risk assessment, which we refer to as indicators. They are obtained by following certain transformation rules using the events and alarms themselves as inputs, also the business indicators related to the company business and ICT profile, and finally the business value of the company's infrastructure elements. These indicators mark the border between the Monitoring Layer and the Real-Time Assessment Infrastructure. For a short introduction about the concept of indicator, this is deeply addressed and developed in D3.1.
- The indicators are used to feed the risk assessment algorithm. This algorithm is based on a risk model and a set of associated model rules, which are executed by the algorithm itself.
- The result of executing this algorithm is a Cyber Risk Assessment Report. The whole process

starting with the indicators' generation and the issuing of the report happens in the Risk Assessment Engine, which will be further discussed.

- Finally, the Cyber Risk Assessment Report information is fed into the Decision Support System in order to help the user to make informed decisions on mitigation measures aimed at improving the company's cyber resilience.

The Real-Time Assessment Infrastructure is composed by three modules:

- 1) **Risk Assessment Engine:** Implements the logic needed in order to obtain a report on the assessment of the risk faced. It puts the cyber incidents in the necessary context to evaluate their impact in the company business process. It uses modelling techniques with an associated algorithm which executes machine-readable model rules. As part of the report, it suggests mitigation measures according to the risks evaluated. The algorithm is executed in near real-time.
- 2) **Decision Support System:** Includes a web-based dashboard where the results of the computations carried out by the Risk Assessment Engine are shown. This includes both the qualitative and quantitative assessment of the risk. It performs the societal impact assessment of cyber risks based on quality criteria. It performs the cost-benefit analysis of implementing a mitigation measure, enabling the decision maker to compare all direct and indirect positive and negative effects of the proposed decisions. As for the mitigation measures, it uses the suggestions received from the Risk Assessment Engine based on the risk level, and then prioritizes the mitigation measures by using the result of the cost-benefit analysis. It helps the operator in the decision process by providing an interface to compare the mitigation measures in terms of cost-benefit.
- 3) **Data Warehouse:** It is the central data-interchange and storage facility, being the access point to the information managed across the WISER Framework. Most relationships and data exchanges among the WISER components are based on read and write operations within this Data Warehouse, as described in deliverables D2.3 and D5.1. It has two parts: a document-based storage and a relational-database storage.

The features provided by WISER rely on the WISER Service the user utilizes. The different services are widely discussed in subsequent deliverables D8.7 and D2.5.

The case of CyberWISER-Light is depicted in Figure 2. A 3-tier-architecture can be identified. The components belonging to the Real-Time Assessment Infrastructure are highlighted in yellow. The risk assessment is the result of the analysis of the business and ICT profile of the company (1)(3), specified by the user via the configuration module, the business value of the company infrastructure elements, fed also thanks to the configuration module (2)(3) and of the results of vulnerability scanners (4). These two inputs are correlated offering a cyber risk assessment understandable from the point of view of management positions, empowered for decision-making (5). It offers, on the one hand, the evaluation of the risk (6) and, on the other hand, the proposed mitigation measures (7). To obtain this, the user invokes the Risk Assessment Engine on demand (8). This is done in two steps: first the Risk Assessment Engine only computes the risk taking into account the information provided in (1)(3). This produces an early version of the report (5)(6)(7). On a second step, once the user provides the information in (2)(3), the vulnerability scanners of the testing module are called (9). These scanners look for vulnerabilities in the target infrastructure (10) and obtain the results (11) which are fed into the Risk Assessment Engine (4) to calculate the refined version of the report (5)(6)(7). Besides, the user is made available a dedicated interface to consult the technical information about the vulnerabilities found (12)(13). Nevertheless this is not part of the Real-Time Assessment Infrastructure itself.

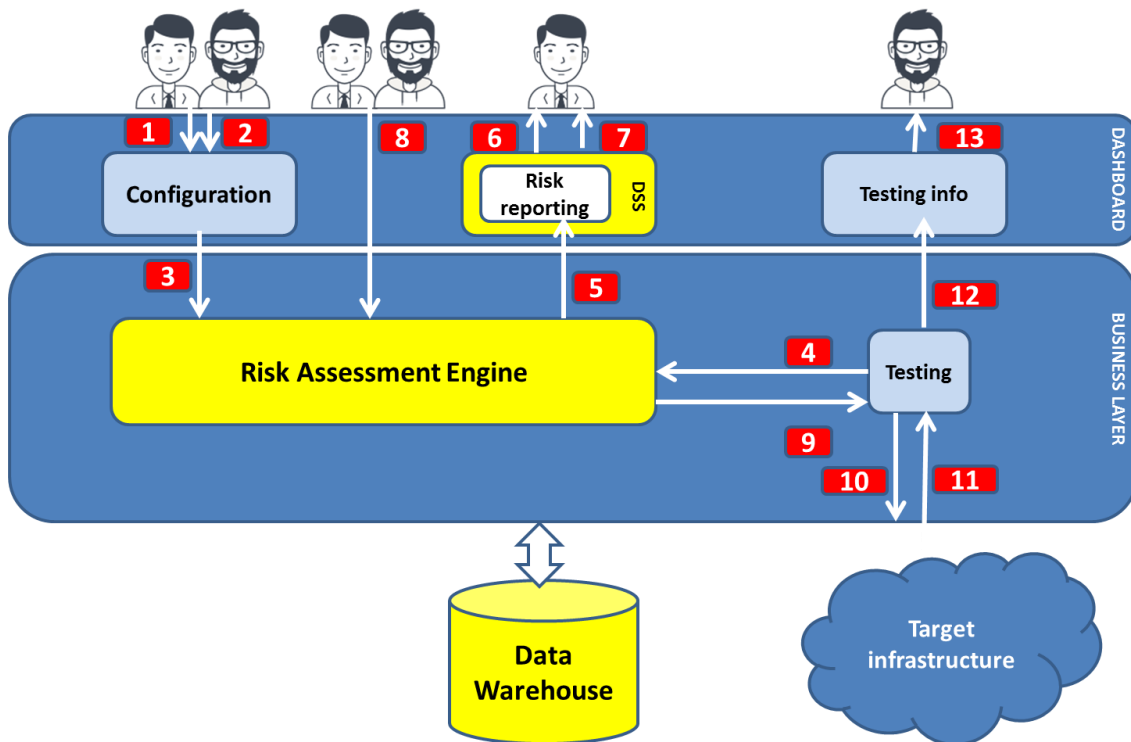


Figure 2. Interplay of the Real-Time Assessment Infrastructure with the rest of the WISER Framework in CyberWISER-Light

The inputs and outputs of the Real-Time Assessment Infrastructure for CyberWISER-Light are the following:

- Inputs
  - Business and ICT profile of the company, provided by the user by filling a questionnaire.
  - Business value of the infrastructure elements being analysed, measured in terms of the evaluation of the confidentiality, integrity, and availability of the stored data.
  - Vulnerabilities detected by the vulnerability scanners
  - Command to launch the Risk Assessment Engine (in CyberWISER-Light it works on demand, unlike in CyberWISER-Essential and CyberWISER-Plus).
- Outputs
  - Report with the risk evaluation including the business and technical aspects.

There is a meaningful leap between CyberWISER-Light and CyberWISER-Essential. The latter incorporates the real-time monitoring of the target infrastructure and a more sophisticated evaluation method based on modelling techniques. Figure 3 shows how the components of the Real-Time Assessment Infrastructure (highlighted in yellow) are related to the rest of the WISER Framework.

The Real-Time Assessment Infrastructure receives input from the Modelling Module. This module takes care of the management of the model repository, including adding, removing, and editing models. The modelling module takes information provided by the user via the configuration dashboard (1)(7)(8) to suggest one or more predefined models to assess the organization cyber risk exposure (2)(3). Using the model selection dashboard, the user chooses one or more of these models and the modelling module collects, processes and passes them to the Risk Assessment Engine (4)(5)(6). The user may edit models, as well as create new models from scratch.

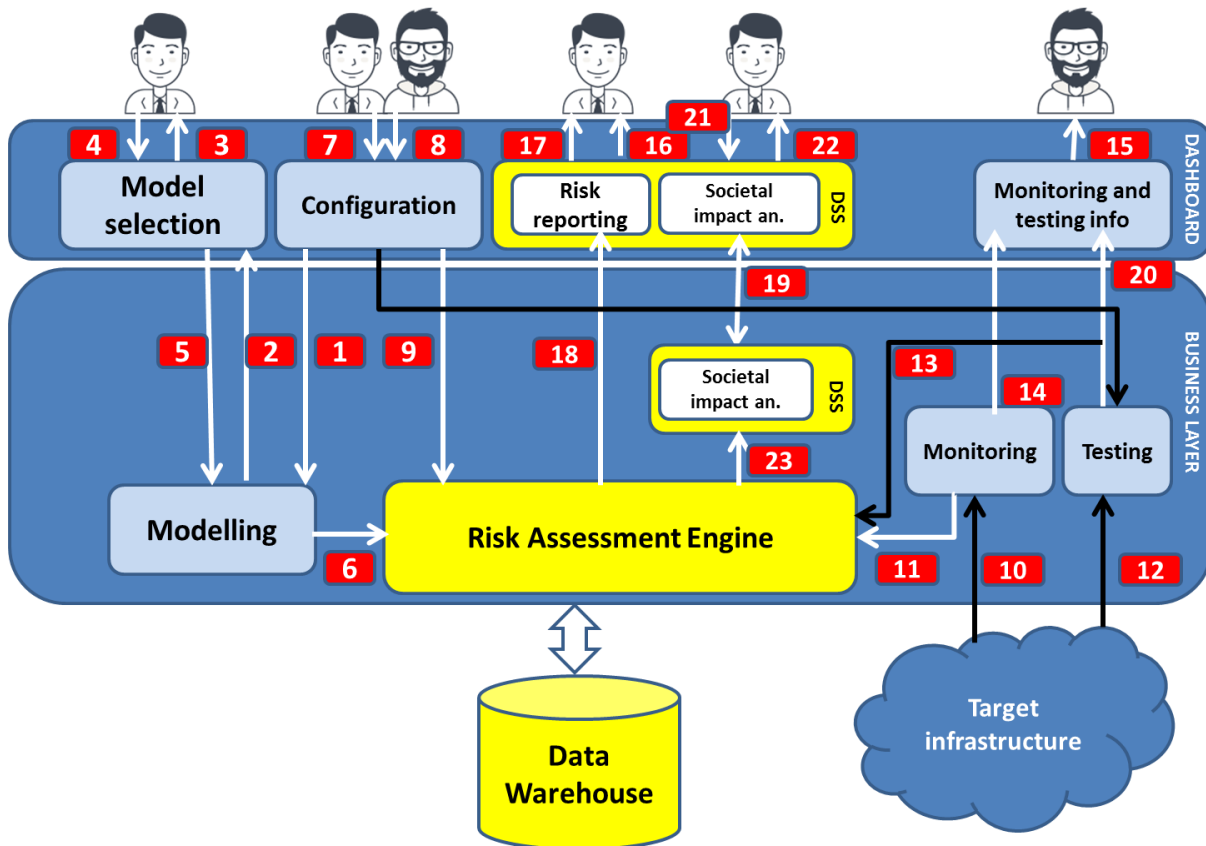


Figure 3. Interplay of the Real-Time Assessment Infrastructure with the rest of the WISER Framework in CyberWISER-Essential

The Risk Assessment Engine assesses the cyber risk exposure of the organization. The result of this evaluation considers relevant business information (18) and it is shown in the Decision Support System dashboard (16). The cyber risk assessment comes along with a set of suggested mitigation measures, which are also shown in the dashboard (17), and whose type is defined in accordance to the results of the cyber risk evaluation.

The cyber risk assessment is performed by using algorithms corresponding to (and derived from) the models chosen by the user in the modelling module. The algorithms take as input the following data, in order to obtain the indicators, populate the models and carry out the cyber risk assessment:

- Information regarding the company business configuration and the company's ICT profile, provided via a questionnaire during the configuration steps (7)(8)(9). In particular, the company daily IT practices have a direct impact in the risk faced by the company.
- Alarms coming from the target infrastructure monitoring module. This information is gathered by sensors deployed within the client infrastructure (10) and processed by the monitoring module. Then, the monitoring module triggers alarms, which are direct inputs to the Risk Assessment Engine (11). Collected events and alarms are also sent to the dashboard (14) to be shown to the user (15). Users having a technical profile are more likely to understand and interpret this information.
- Vulnerabilities found by the vulnerability scanners on the target infrastructure. The scanners detect the vulnerabilities and inform the testing module (12), and then the testing module processes this information and sends it to the Risk Assessment Engine as input for the evaluation algorithm (13). The information obtained by the vulnerability scanner is also sent to

the dashboard (20) to be shown to the user (15). Again, tech users are the most appropriate ones to take advantage of this information.

The business layer of the Decision Support System contains the business logic corresponding to the societal impact analysis feature offered by the presentation layer of the Decision Support System. This societal impact analysis is done taking as starting point the mitigation measures suggested by the Risk Assessment Engine, being part of the issued risk report (23). The user leverages the dashboard to introduce the information needed to evaluate the societal impact of the measures (21), this is processed within the business layer of the societal impact analysis module, which responds once the processing is done (19) and the user is shown the results (22).

The exchange of information is done by means of the Data Warehouse. In order to allow the interaction of the different components of the WISER platform with the Data Warehouse, an HTTP (REST) API and communication bus, based on RabbitMQ (applying the AMQP protocol) are implemented. The latter provides push notifications about data changes, triggering risk assessment after an indicator value changes, and provides direct message passing between components.

Finally, CyberWISER-Plus adds to the schema the functionality of the cost-benefit analysis of the mitigation measures suggested by the Risk Assessment Engine. The user introduces the necessary data to carry out the analysis, which is executed by the business layer component of the functionality, which responds and the result is shown to the user by means of the dashboard.

The complete schema showing the interaction of the Real-Time Assessment Infrastructure modules (highlighted in yellow) with the rest of the WISER Framework, for CyberWISER-Plus can be seen in Figure 4.

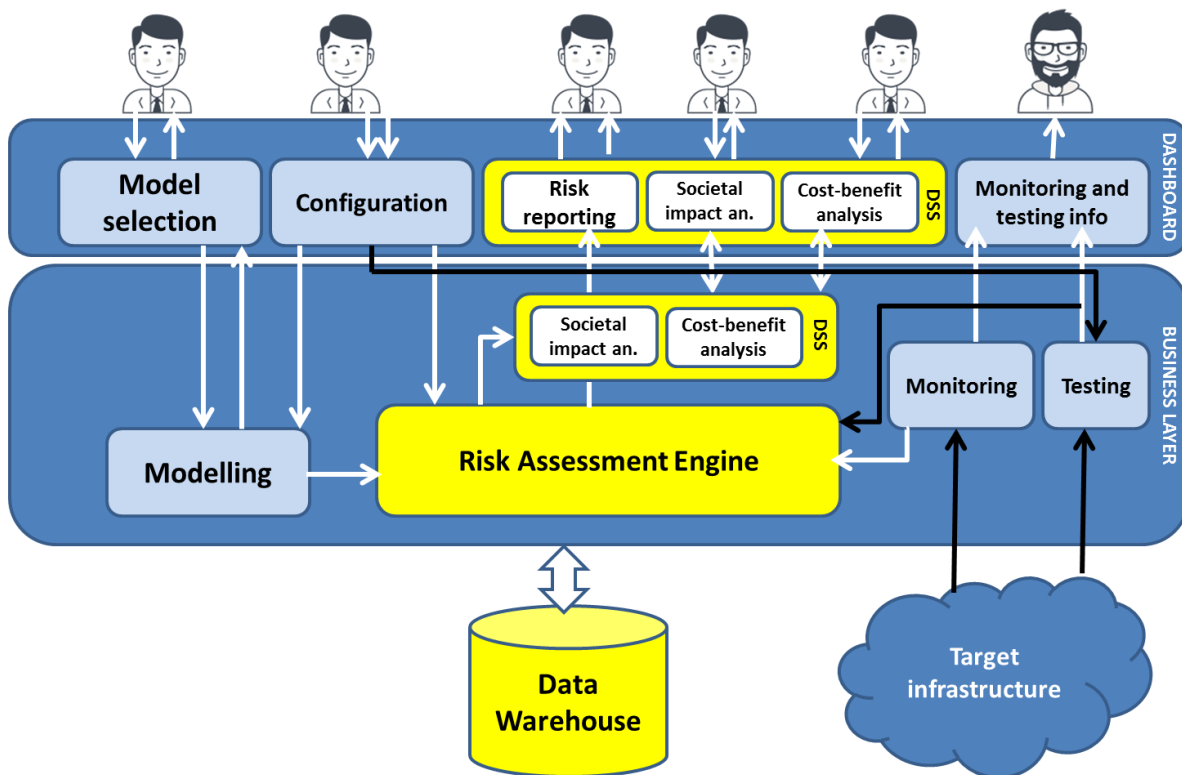


Figure 4. Interplay of the Real-Time Assessment infrastructure with the rest of the framework in CyberWISER-Plus

As a summary, below the inputs and outputs of the Real-Time Assessment Infrastructure for the



---

cases of CyberWISER-Essential and CyberWISER-Plus are enumerated:

- Inputs
  - Configuration information, composed of the following elements, namely: company business profile, company ICT profile and target infrastructure information. For the latter, this information involves not only technical aspects, but also the business value each infrastructure element has. This is paramount to correlate the cyber incidents to their business impact and make possible the multi-layer cyber risk assessment brought by WISER.
  - Selected model/s: This/these models is/are selected by the user; by default WISER suggest a model that can be changed by the user.
  - Vulnerabilities detected by the monitoring scanners, coming from the testing module.
  - Events and alarms providing the status of the cyber climate, thanks to the Monitoring Module
- Outputs
  - Risk assessment report, containing the evaluation of the risks addressed in the risk model/s considered, and related mitigation actions, obtained also based on the modelling rules.

This section ends with the summary of the main innovation streams brought thanks to the WISER Real-Time Assessment Infrastructure. A very important novelty is the fact of providing a second layer of cyber risk assessment. The market needs are changing in such a way that companies do not settle for a purely technical treatment of cyber security issues, analysing the potential threats, on the one hand, and monitoring and detecting incidents, on the other hand. They want to go further, they manifest that it is more and more necessary to understand the cyber climate in which the company operates from the perspective of the company business. This means that the top managers need to have a very clear and understandable picture of the cyber resilience status of the company, and the weaknesses that might be exploited by an attacker in his own benefit, damaging the interests of the company. A business interpretation of cyber risk puts the company in a good position to make decisions of high importance with impact in the long term (rather than simple patches), aiming at improving the cyber resilience of the company.

The second main innovation stream is the capability of supporting the decision-making process. The user will not feel alone to make the decision: WISER will suggest mitigation measures and will provide a methodology to rank the range of possible actions and to prioritize them. This way, the top management positions will make informed decisions based on a solid rationale to draw the company cyber security strategy. Finally, for those made decisions, a novel approach to the analysis of the societal impact is put in place. The societal impact, despite not being visible, may be harmful for the company. In such sense, it is very positive for the company to analyse the possible societal consequence a certain mitigation action could have.

### 3 Risk Assessment Engine

---

#### 3.1 Functional design

##### 3.1.1 Overview

The Risk Assessment Engine is the brain of the WISER Framework. This module executes a risk model-based algorithm in order to evaluate the cyber risk of the company. This evaluation is done in near real-time. It is a key component capable of putting the cyber incidents detected in the client infrastructure in the necessary context to evaluate their direct impact in the company business processes. Along with the evaluation of the cyber risk, the Engine suggests mitigation measures, if any. This information is sent to the Decision Support System, where it is represented in a user-friendly way.

The internal operation and composition of the Risk Assessment Engine depends on the WISER Service being used: CyberWISER-Light, CyberWISER-Essential and CyberWISER-Plus

CyberWISER-Light is addressed in depth in the context of Deliverable D2.3. The details about the operation of the Risk Assessment Engine for CyberWISER-Light are provided in Deliverable D5.1. Therefore, the reader is recommended to consult these documents for further details. In this deliverable, a summary of the operation of the Risk Assessment Engine is provided, using Figure 5 to support the explanation. This figure shows the internal components of the Risk Assessment Engine for CyberWISER-Light and its relationships with other modules. CyberWISER-Light evaluates the risk in two steps, with the execution of the Risk Assessment Engine in both steps happening on demand (4 in Figure 5). The first step is about collecting the company profile via a questionnaire which is used to evaluate the risk being inherent to the company business and ICT profile of the client. Algorithm 1 takes the answers provided by the user (1) and gives each an associated score and weight. Then, the score is aggregated by grouping the answers in six different groups and summing up the points. Finally, the result is transferred to the module in charge of producing the report (2). The user can consult the report by means of the dashboard (3).

In the second step, Algorithm 2 uses the information provided by the user (5) with respect to the importance certain security aspects have for the different categories of infrastructure elements. The security aspects considered are the confidentiality, integrity and availability of the information (more information about these security aspects can be obtained by consulting deliverables D2.2 and D5.1).

This information is combined with the vulnerabilities detected by executing vulnerability scanners, which are launched when the Algorithm 2 is triggered (6)(7). This is done by means of the testing module. The scanners detect the vulnerabilities and obtain the information from the target infrastructure (8) and make the result available to the Algorithm 2 (9).

With the information about the business value of confidentiality, integrity, and availability of each category, and knowing which security aspect/s is/are impacted by for each vulnerability detected, Algorithm 2 establishes a link between the vulnerabilities found (9) and the business impact they have. When the algorithm finishes the computation, sends the results to the module in charge of producing the report (10), which sends it to the dashboard to be visualized by the user (3).

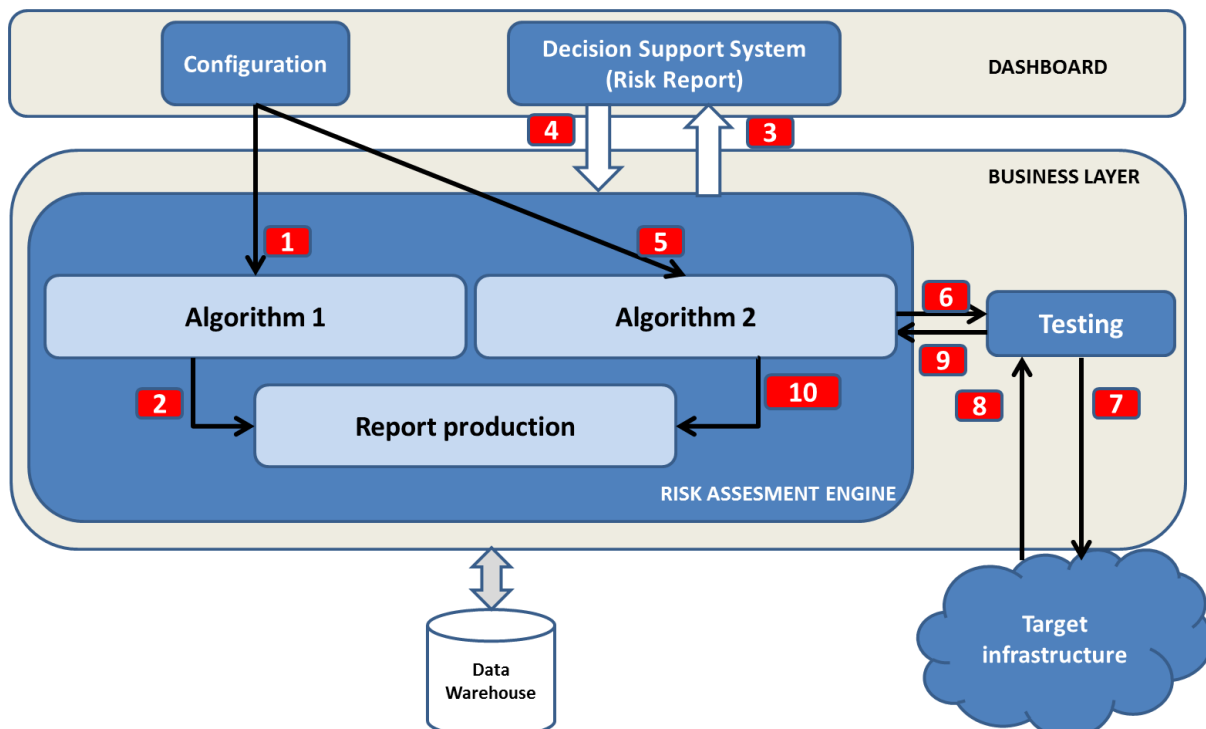


Figure 5. Risk Assessment Engine internal detail for CyberWISER-Light

Figure 6 shows the internal composition of the Risk Assessment Engine implemented in CyberWISER-Essential and CyberWISER-Plus. From the functional point of view there is no difference between them, the actual difference lies on the complexity of the models evaluated and the quantity and complexity of indicators used. The description of each block follows:

- 1) Indicator value generator:** Transforms the information coming from the configuration, the monitoring and the testing modules into inputs (called indicators) understandable by the models. It feeds these indicators to the triggering detector and the DEXi and R model instantiators.
  - a. Inputs
    - i. Information regarding the company business and ICT profile coming from the Configuration Module (1).
    - ii. Information about the elements of the infrastructure, with their technical characteristics and their importance for the business in terms of the confidentiality, integrity and availability of the data stored. This also comes from the Configuration Module (2).
    - iii. Events and alarms coming from the Monitoring Module (3).
    - iv. Vulnerabilities coming from the Testing Module (4).
  - b. Outputs
    - i. Indicators values: They are grouped per target and fed into the DEXi model instantiator (5), the R model instantiator (6) and to the triggering detector (7)
- 2) Triggering detector:** detects changes in the inputs for the models and, according to certain rules, launches a new execution of the model rules. In such case, a new risk report will be generated.
  - a. Inputs
    - i. Model used, in order to detect whether it changed (8).
    - ii. The indicators generated, in order to detect if some change took place (7).
  - b. Outputs
    - i. Command to launch the DEXi model rules executor (9)
    - ii. Command to launch the R model rules executor (10)
- 3) DEXi model instantiator and R model instantiator:** Create an instance of the models using the current values of the indicators they use as inputs.
  - a. Inputs
    - i. Model: this is the model to instantiate as many times as elements to be evaluated the risk (13)(14)
    - ii. Indicators produced by the Indicators Value Generator (5)(6)
  - b. Outputs
    - i. A model instance per target, with the corresponding model rules, to be used in the algorithm to be executed (11)(12)
- 4) DEXi model rules executor and R model rules executor:** execute the model rules to evaluate the model, and send the results to the Aggregator
  - a. Inputs
    - i. Command to launch the algorithm (9)(10)
    - ii. Model instances to use (11)(12)
  - b. Outputs
    - i. Risk Assessment per target within the client infrastructure (15)(16), fed into the Aggregator.
- 5) The Aggregator** aggregates the results at the different levels foreseen in the risk report and sends the output to the Decision Support System.
  - a. Inputs
    - i. Risk Assessment and mitigation proposals per target within the client infrastructure (15)(16)
  - b. Outputs

i. Risk Assessment of the infrastructure as a whole and mitigation measures proposed (17).

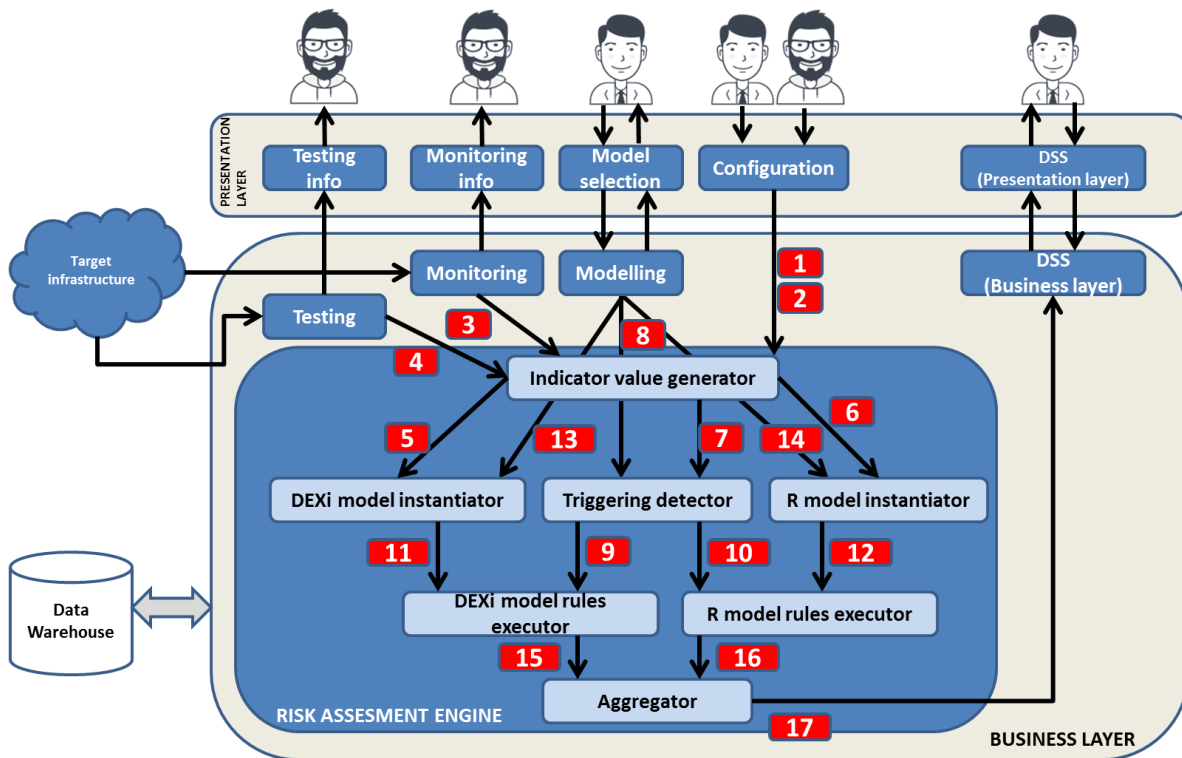


Figure 6. Risk Assessment Engine internal detail for CyberWISER-Essential and CyberWISER-Plus

### 3.1.2 Inputs

As it was introduced in D5.1, the Risk Assessment Engine receives inputs from four main sources (see Figure 7):

- configuration provided by the user through the graphical dashboard,
- real-time events and alerts received from the monitoring infrastructure,
- vulnerabilities detected in the user infrastructure when a vulnerabilities scan is performed and
- risk pattern modelling provided by WISER.

All these inputs are received by the **Indicator Value Generator** (as reflected in Figure 6) module, which evaluates them and generates a new indicator or updates an existent one when required. These indicators will be also stored in the Data Warehouse<sup>4</sup> (/rae/indicator\_values) for its usage by the Triggering Detector module.

<sup>4</sup> The Data Warehouse is addressed in section 4 and, in particular, the data model of the relational part and the involved tables are addressed in Section 4.2.1.



Figure 7. Inputs and output of the Risk Assessment Engine

### 3.1.2.1 Configuration inputs

In CyberWISER-Light, the user will complete a questionnaire (Figure 8 shows a screenshot of the user interface with one of the questions posed to the user) to obtain information related to the company to draw the profile that will be used by the Risk Assessment Engine. This company profile is stored in the DWH (/config/company\_profiles). The details about the information requested to the user can be found in the table 1 of D5.1. In addition, this is supplemented with information also provided by the user with respect to the importance in terms of the confidentiality, availability, and integrity of information of different categories of infrastructure elements (see Figure 9). These categories are enumerated in Section 3.1.1.

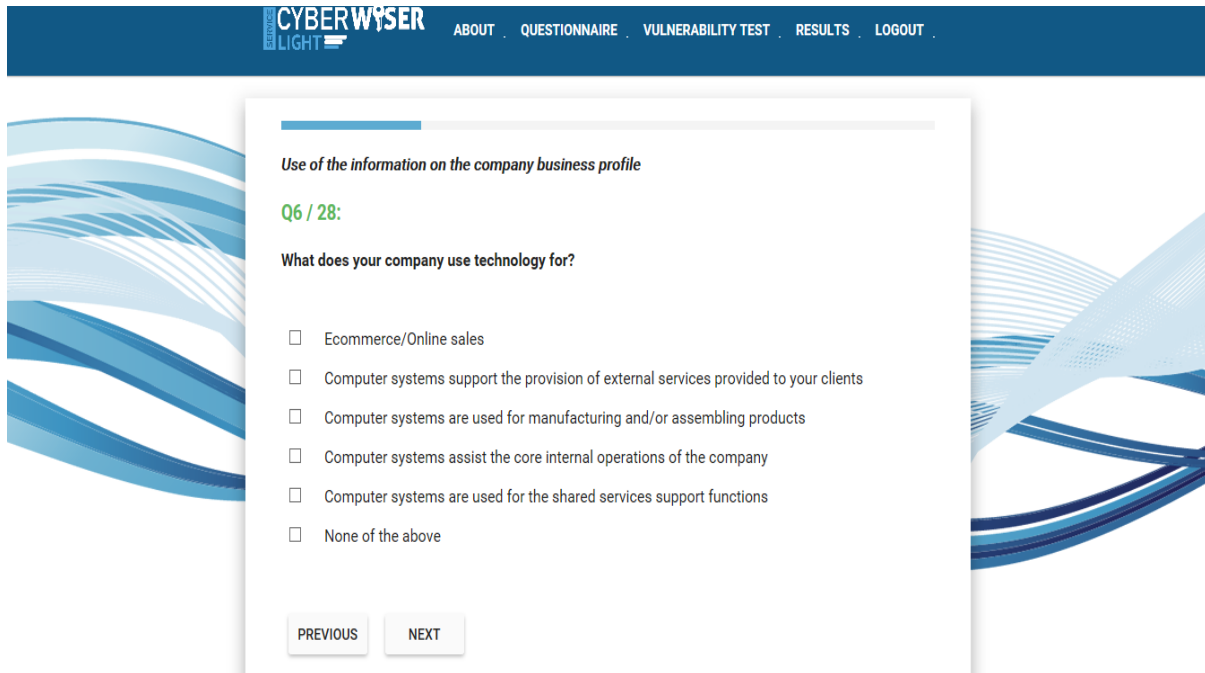
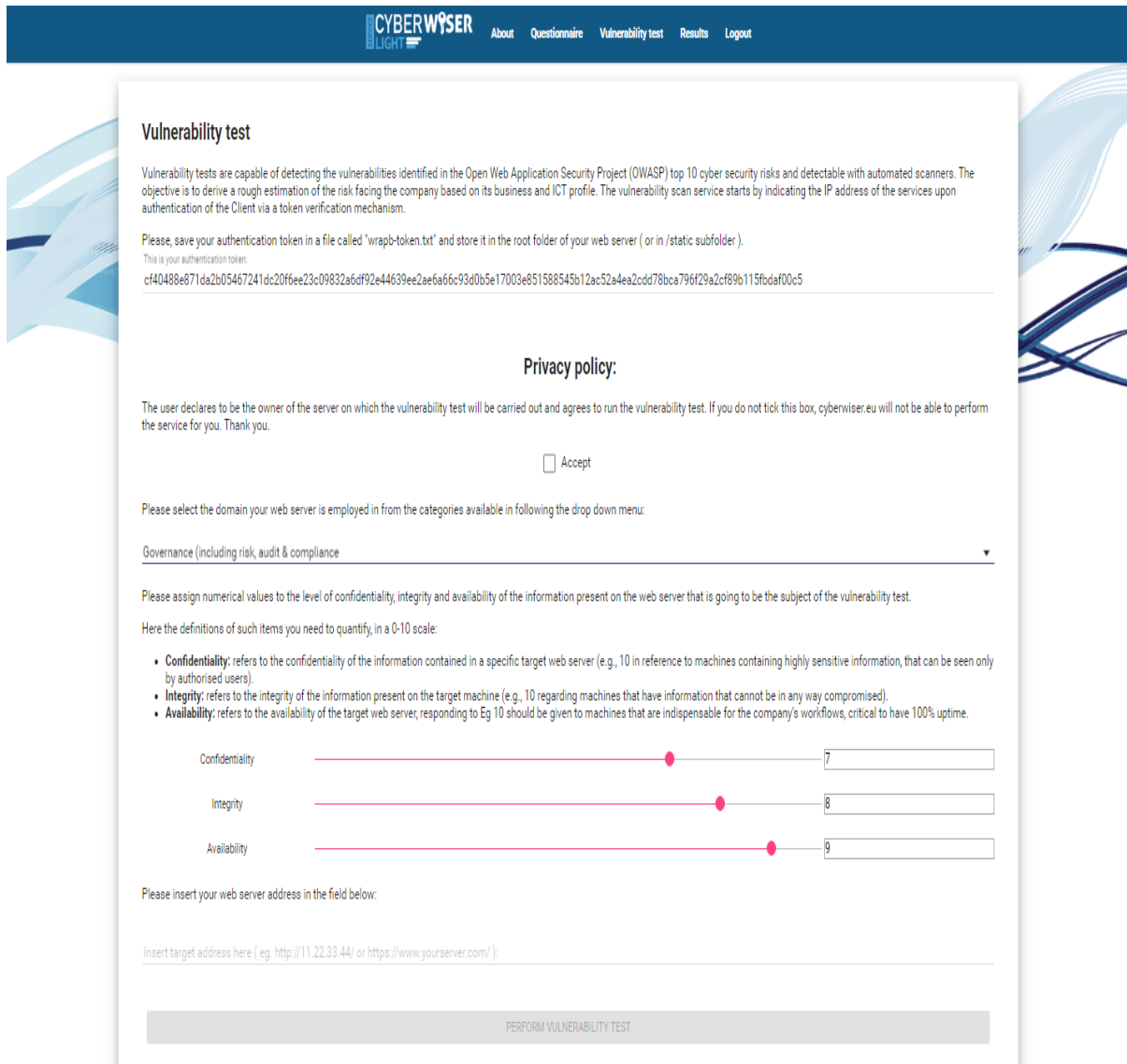


Figure 8. Questionnaire interface in CyberWISER-Light



**Vulnerability test**

Vulnerability tests are capable of detecting the vulnerabilities identified in the Open Web Application Security Project (OWASP) top 10 cyber security risks and detectable with automated scanners. The objective is to derive a rough estimation of the risk facing the company based on its business and ICT profile. The vulnerability scan service starts by indicating the IP address of the services upon authentication of the Client via a token verification mechanism.

Please, save your authentication token in a file called "wrapb-token.txt" and store it in the root folder of your web server ( or in /static subfolder ).

This is your authentication token:  
cf40488e871da2b05467241dc20f6ee23c09832a6df92e44639ee2ae6a66c93d0b5e17003e851588545b12ac52a4ea2cdd78bca796f29a2cf89b115fbdaf00c5

**Privacy policy:**

The user declares to be the owner of the server on which the vulnerability test will be carried out and agrees to run the vulnerability test. If you do not tick this box, cyberwiser.eu will not be able to perform the service for you. Thank you.

Accept

Please select the domain your web server is employed in from the categories available in following the drop down menu:

Governance (including risk, audit & compliance)

Please assign numerical values to the level of confidentiality, integrity and availability of the information present on the web server that is going to be the subject of the vulnerability test.

Here the definitions of such items you need to quantify, in a 0-10 scale:

- **Confidentiality:** refers to the confidentiality of the information contained in a specific target web server (e.g., 10 in reference to machines containing highly sensitive information, that can be seen only by authorised users).
- **Integrity:** refers to the integrity of the information present on the target machine (e.g., 10 regarding machines that have information that cannot be in any way compromised).
- **Availability:** refers to the availability of the target web server, responding to Eg 10 should be given to machines that are indispensable for the company's workflows, critical to have 100% uptime.

Confidentiality

Integrity

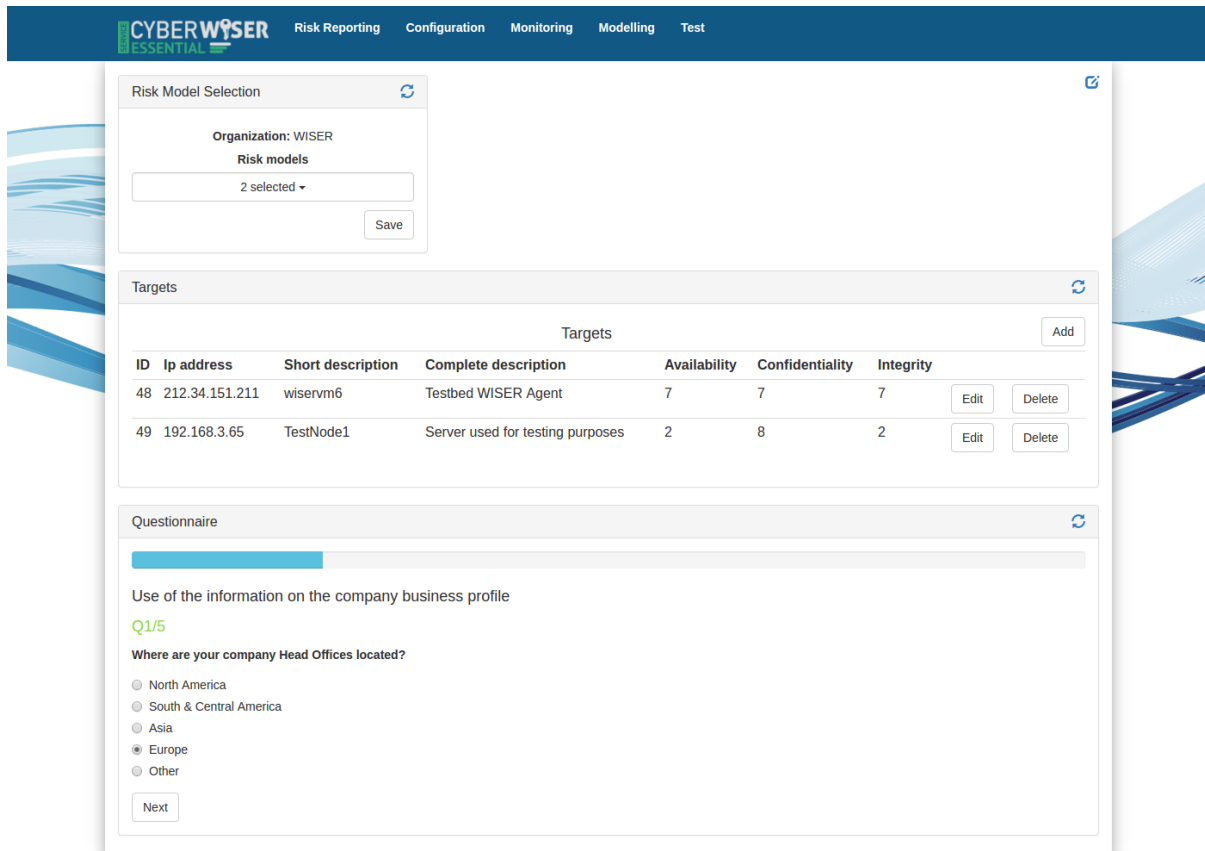
Availability

Please insert your web server address in the field below:

Insert target address here ( eg. http://11.22.33.44/ or https://www.yourserver.com/ ):

Figure 9. Target importance in terms of confidentiality, integrity and availability

In CyberWISER-Essential, the user interacts with the configuration dashboard not only to answer the questionnaire but also to specify the target infrastructure with more detail, indicating a descriptive name and the IP address of each element. Again, the confidentiality, integrity, and availability of each element is scored with a value between 0 and 10. Figure 10 shows how the questionnaire and the target setup are framed in the configuration area of the dashboard. Figure 11 and Figure 12 show some detail of the target configuration process in CyberWISER-Essential.



**CYBERWISER ESSENTIAL** Risk Reporting Configuration Monitoring Modelling Test

**Risk Model Selection**

Organization: WISER

**Risk models**

2 selected ▾

Save

**Targets**

ID	Ip address	Short description	Complete description	Availability	Confidentiality	Integrity		
48	212.34.151.211	wiservm6	Testbed WISER Agent	7	7	7	Edit	Delete
49	192.168.3.65	TestNode1	Server used for testing purposes	2	8	2	Edit	Delete

**Questionnaire**

Use of the information on the company business profile

Q1/5

Where are your company Head Offices located?

- North America
- South & Central America
- Asia
- Europe
- Other

Next

Figure 10. Configuration questionnaire, target setup and risk model selection user interface (CyberWISER-Essential)



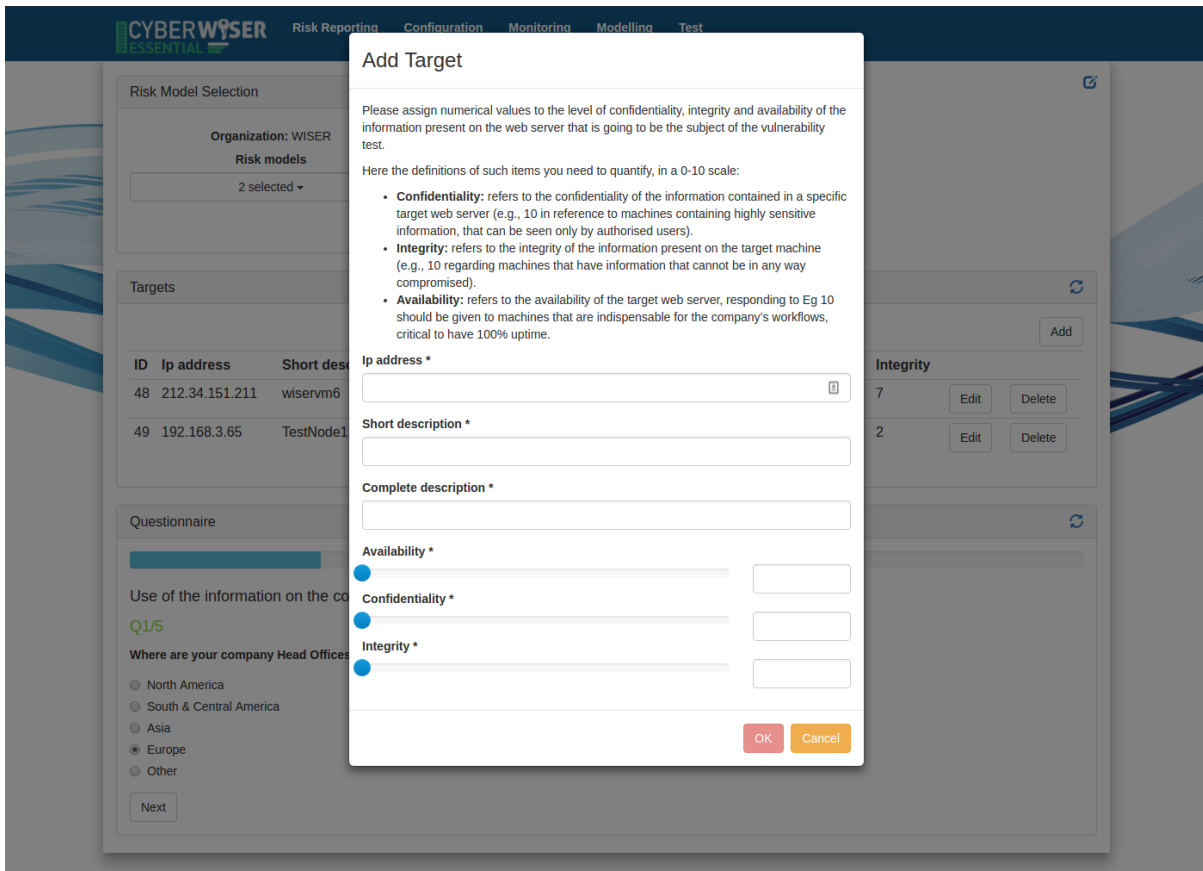


Figure 11. Interface to specify elements of the target infrastructure

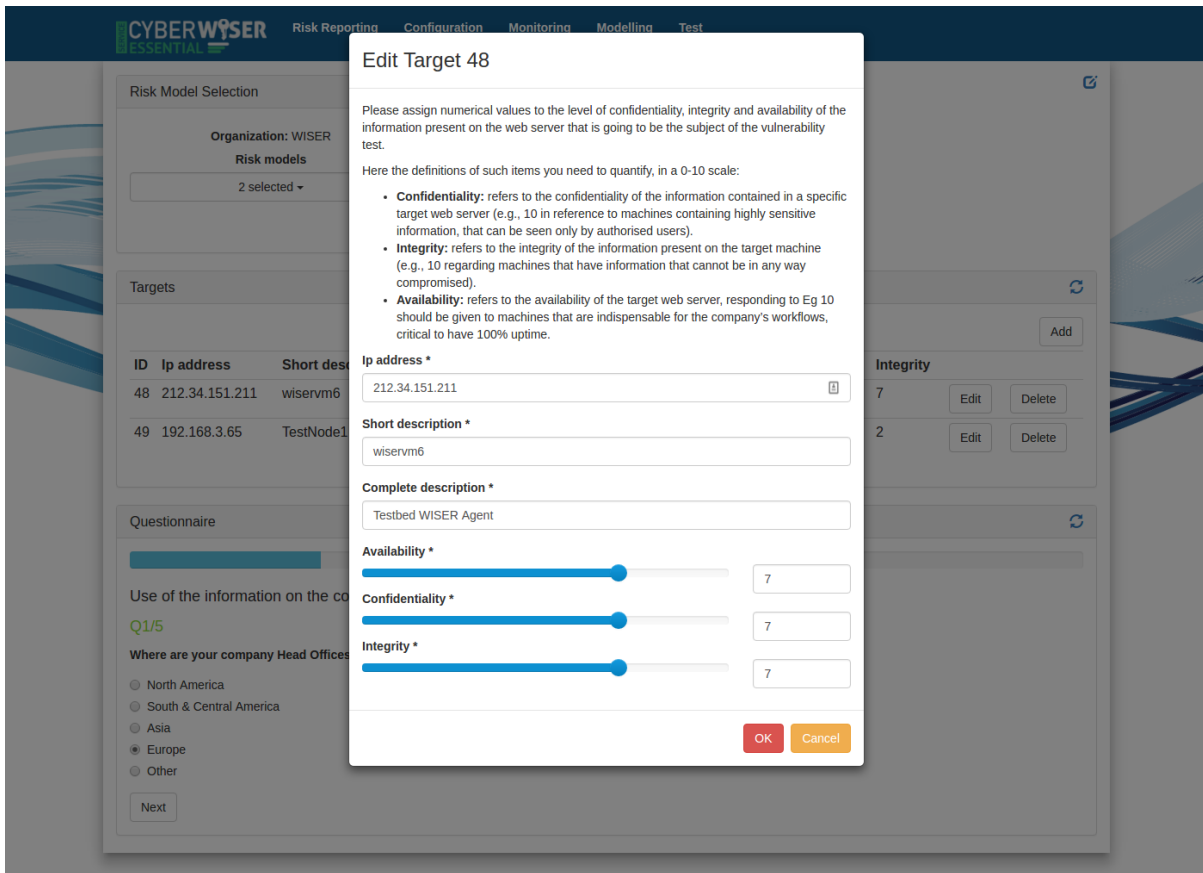


Figure 12. Editing the targets within the client infrastructure (CyberWISER-Essential)

In CyberWISER-Plus, it is also possible to specify which applications are running in each target indicating the ports used. All the information associated to the target infrastructure for a specific organization is also stored in the DWH (/rae/targets). Figure 13 and Figure 14 illustrate this.

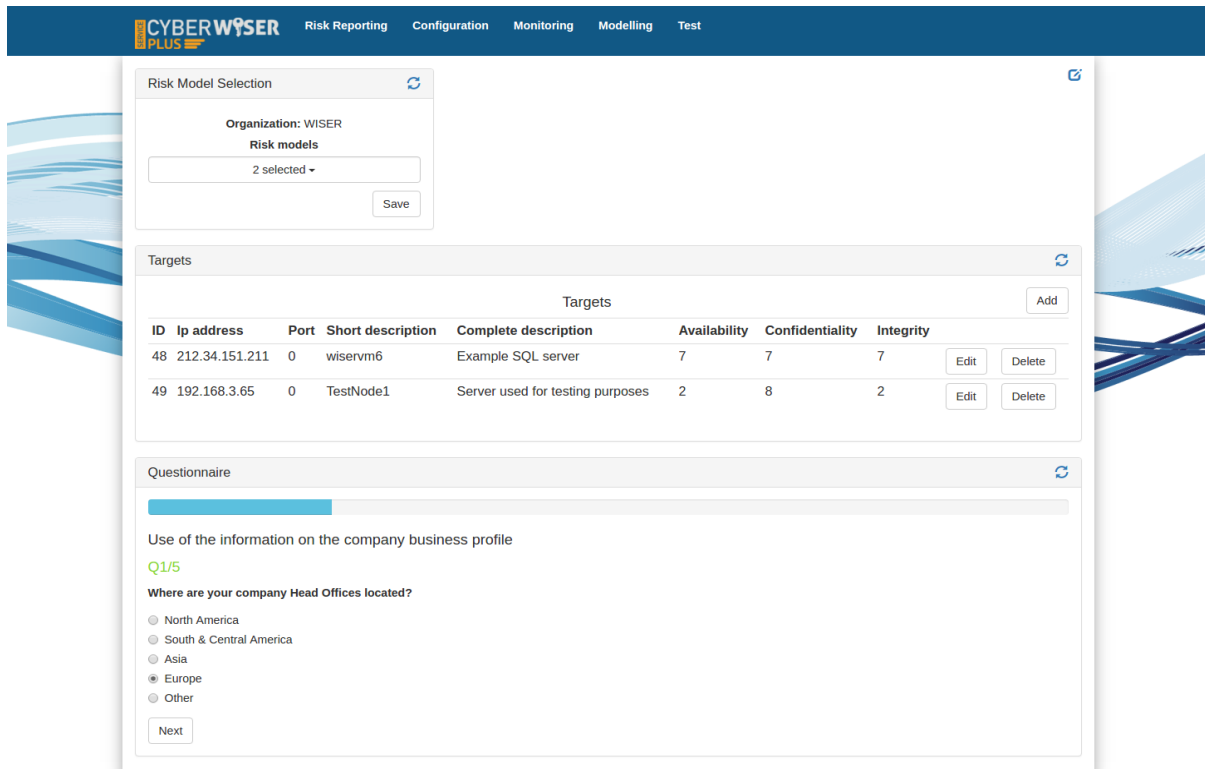


Figure 13. Configuration questionnaire, target setup and risk model selection user interface (CyberWISER-Plus)

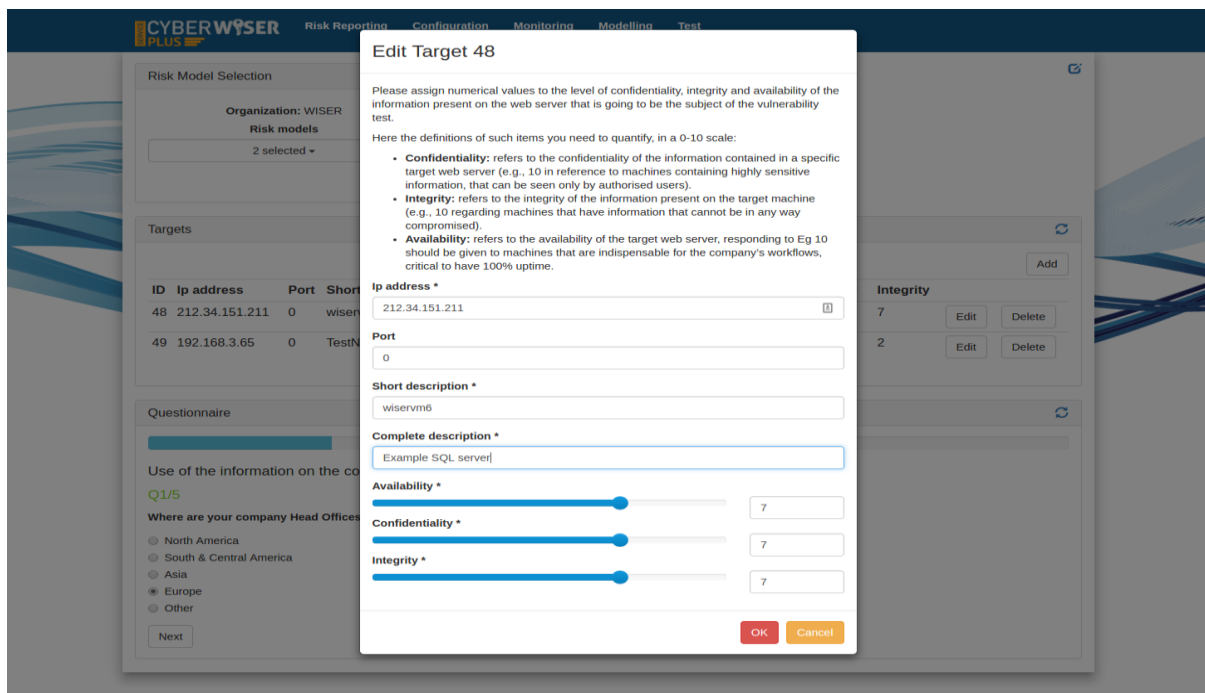


Figure 14. Editing targets within client infrastructure (CyberWISER-Plus)

Both inputs, the company profile and the target infrastructure configured by the user, are received by the Risk Assessment Engine from the Data Warehouse through notifications.

The Indicator Value Generator module of the Risk Assessment Engine will subscribe to the RabbitMQ queue called "**rae.ivg-dwhnotifs**" to receive a notification when a company profile or a target record has been created, updated, or deleted in the DWH. Below it is shown an example of a notification received:

```
{"object_ID": 1, "action": "CREATED", "object_class": "Target"}
```

Three actions can be notified in this way from the DWH to the Risk Assessment Engine: "CREATED", "DELETED" or "UPDATED". The object\_class parameter will be "Target" or "CompanyProfile" depending on the type of notification.

### 3.1.2.2 Monitoring inputs

The events and alarms generated by the monitoring infrastructure are used by the Risk Assessment Engine to get information about what is happening in the client infrastructure in real-time.

The Indicator Value Generator module of the Risk Assessment Engine receives the incoming events and alarms arriving to the DWH subscribing directly to the RabbitMQ queues "**rae.ivg-events**" and "**rae.ivg-alarms**".

The format of the events and alarms received are the same described in section 3.2.2 of D5.1.

### 3.1.2.3 Testing inputs

The vulnerability reports generated when a vulnerability scan is done in the client infrastructure are also used by the Risk Assessment Engine to update the value of the testing indicators related to the vulnerabilities affecting the client target system.

The Indicator Value Generator module of the Risk Assessment Engine receives the incoming vulnerability reports arriving to the DWH subscribing directly to the RabbitMQ queue "**rae.ivg-vulnreports**".

The format of the reports received is the following one:

```
{
  "target": <STRING target web application URL>,
  "target_id": <INTEGER target ID>,
  "task_status": <STRING scanning task status ("FINISHED" or "FAILED")>,
  "report": <JSON array of vulnerability info objects**>,
  "reason": <STRING details about the error***>,
  "log": <STRING vulnerability scanner tools logs>
}
```

The "report" section is only included if the vulnerability scan was successful. In case of an error, this section is replaced by "reason" containing an error message. The "report" section contains multiple objects with the following format (one per each vulnerability found):

```
{
  "desc": "Web Browser XSS Protection is not enabled, or is disabled by the
configuration of the 'X-XSS-Protection' HTTP response header on the web server\n\t",
  "reference":
"https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet\n
\thttps://blog.veracode.com/2014/03/guidelines-for-setting-security-headers/\n\t",
  "risk_level": "Low (Medium)",
  "short_desc": "Web Browser XSS Protection Not Enabled",
  "solution": "Ensure that the web browser's XSS filter is enabled, by setting
the X-XSS-Protection HTTP response header to '1'.\n\t",
}
```

```
"source_pentest": "OWASP ZAP",  
"w_risk_level": 20,  
"wascid": "14"  
}
```

### 3.1.2.4 Modelling inputs

The user interacts with the configuration dashboard of CyberWISER-Essential and CyberWISER-Plus to select the risk model/s to be considered from the ones available in the WISER Framework (see Figure 15). Those selected are stored in the DWH<sup>5</sup>.

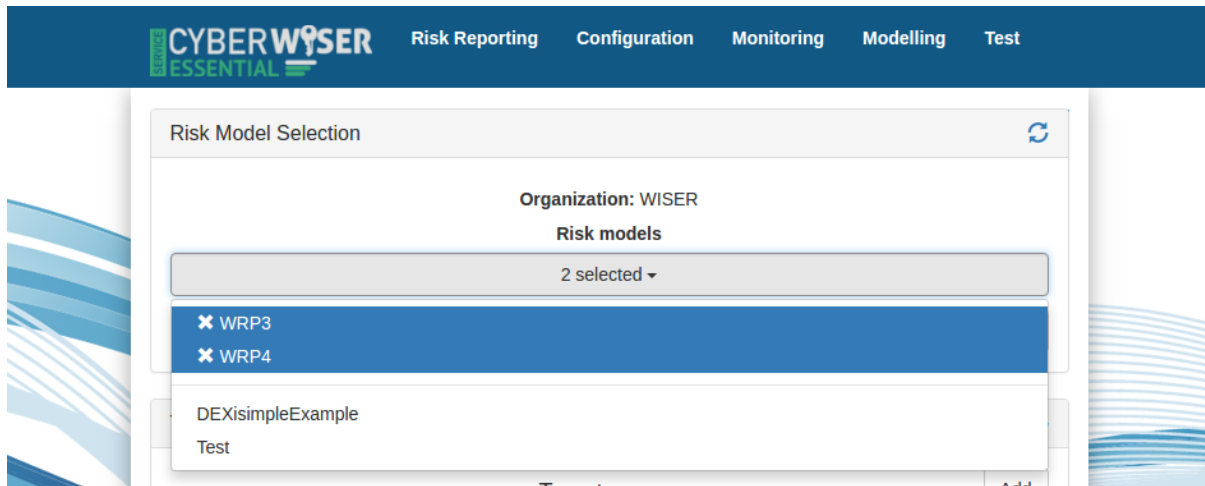


Figure 15. Risk model selection by means of the configuration dashboard

When a risk model is selected or created by the user or an existent one is modified, the DWH sends a notification to the Risk Assessment Engine. This notification is received by the Triggering Detector module subscribed to the RabbitMQ queue called "rae.td-dwhnotifs". Below it is shown an example of a notification received:

```
{"object_ID": 1, "action": "UPDATED", "object_class": " SelectedRiskModel"}
```

Three actions can be notified in this way from the DWH to the Risk Assessment Engine: "CREATED", "DELETED" or "UPDATED". The object\_class parameter will be "RiskModel" or "SelectedRiskModel" depending on the type of notification.

Section 3.1.4 addresses this in detail, explaining the different triggering cases by using sequence diagrams.

### 3.1.3 Outputs

This section is focused on describing the outputs produced by the Risk Assessment Engine. With this respect, a differentiation has to be made among the different operation modes.

First of all, the accuracy and quality of the information in the report clearly depends on the inputs made available to produce it. In CyberWISER-Light, the questionnaire and the vulnerabilities found

<sup>5</sup> In particular in the endpoint /config/selected\_risk\_models where the risk models used for risk assessment for each specific organization are stored. It is important not confusing this endpoint with the endpoints /modelling/risk\_models (with data related to the available risk models for its selection by the organizations) and /modelling/specific\_risk\_models (with specific risk models associated to a custom risk model that is only available to users of a same organization).

are the inputs used to produce the very first approach to the risk assessment of the client. CyberWISER-Essential adds information obtained from the cyber climate by means of monitoring techniques and modelling techniques to provide more advanced algorithms, with expected better results. The qualitative and quantitative (economic) assessments of risks take these inputs into consideration. Mitigation measures are proposed according to the results of the assessment. As for CyberWISER-Plus, it adds sensors working not only at the network layer, but also at the application layer. Moreover, the range of vulnerability scanners available and the catalogue of potentially detectable vulnerabilities are broader, thus improving the accuracy of the monitoring events. This leads to more sophisticated correlation rules and detection of more sophisticated attacks. The business information collected by questionnaires the user fills will be more accurate and complete. The models used are also more sophisticated.

In terms of format, the reports coming from CyberWISER-Light, on the one hand (see section 3.1.3.1), and CyberWISER-Essential and CyberWISER-Plus, on the other hand (see section 3.1.3.2) are quite different to each other. Reports generated for CyberWISER-Essential and CyberWISER-Plus follow the same format, what makes a difference is their accuracy and reliability, based on the quantity and quality of the inputs processed and analysed to produce such reports. CyberWISER-Plus, being the advanced mode of operation, will provide the most complete reports.

### 3.1.3.1 Report for CyberWISER-Light

As presented in Deliverable D2.3 and updated herein, in CyberWISER-Light, the report has two different parts, which are provided separately: the assessment of the risk derived from the company profile and the risk due to vulnerabilities found in the target infrastructure.

#### 3.1.3.1.1 Risk derived from company profile

This report is structured in 6 different sections, namely: 1) Company business profile, 2) Governance, 3) Data managed, 4) IT policies, 5) Outsourcers and 6) Past cyber risk episodes.

Each section is associated a set of questions, answered by the user. The report shows the answers the user provided along with some explanatory text including relevant information to the user, depending on the particular answer. A rationale behind the reasoning presented in each piece of text usually accompanies the assertions. For each section, the answers of the user determine a risk exposure specified in a qualitative manner, with 5 possible values: low, medium-low, medium, medium-high and high.

Figure 16 shows the summary of the report produced by CyberWISER-Light, where a spider web is used to express in a graphical manner the risk to which the user is exposed, plotting in such web the evaluations for each section.

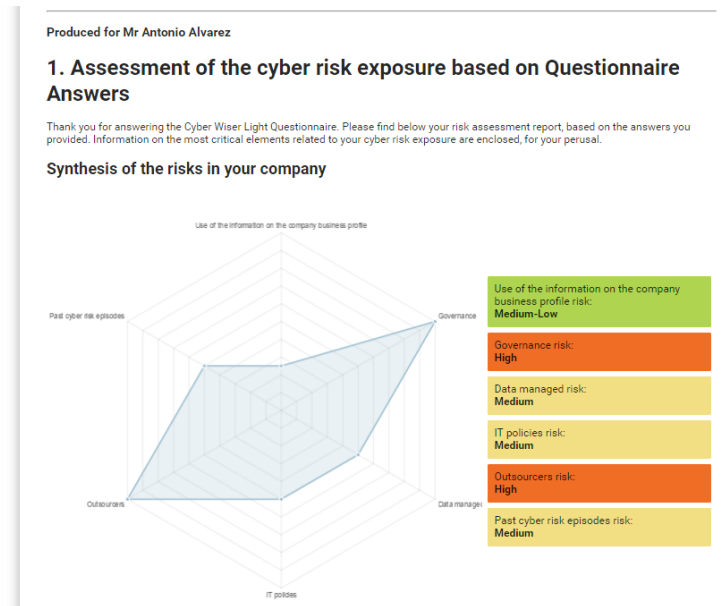


Figure 16. Summary of the risk report produced by CyberWISER-Light

The user can go into details and study the rationale of how each of the answers provided has been evaluated. A PDF file can be obtained with the detailed analysis of each question, leading to the assessment of each of the six parts and to the overall result. Figure 17 shows a screenshot of one of the pages of this report.

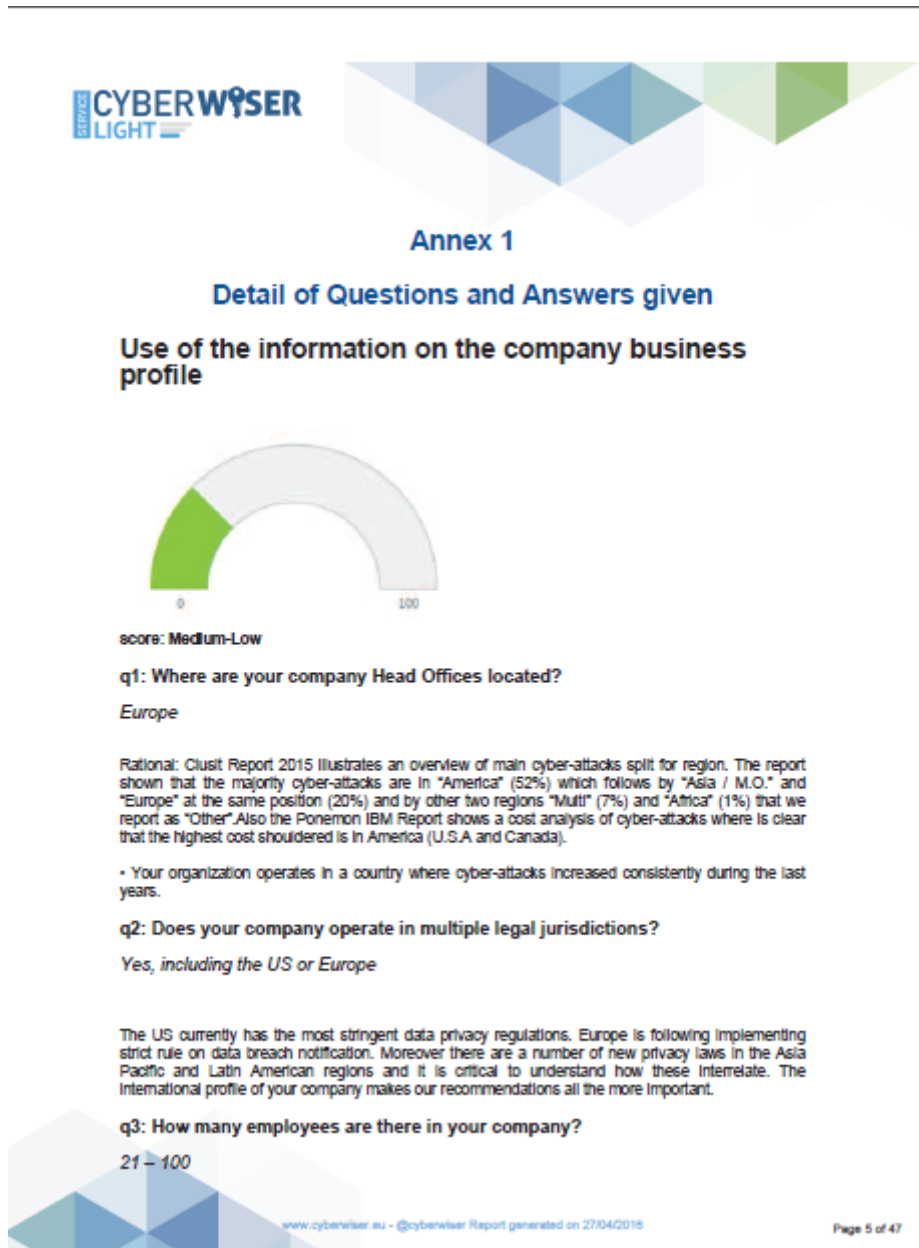


Figure 17. CyberWISER-Light PDF report. Page sample

### 3.1.3.1.2 Risk due to vulnerabilities found in the target infrastructure

This report enumerates the vulnerabilities found, specifying: 1) short description of the vulnerability, 2) detailed description of the vulnerability, 3) risk level and 4) solution.

Besides, a risk exposure level, derived from the impact the detected vulnerabilities have in the business, is presented. This risk level is specified in a qualitative manner and ranges from low to high, through medium-low, medium and medium-high. The report can be consulted both by means of the user interface (see Figure 18) and by producing a PDF report (see Figure 19).



Report Details:

Save PDF to DB | Download PDF without saving

Target Category	Business Impact
media_and_communication	Medium ( 0.433 )

Detected Vulnerabilities			
Short Description	Description	Risk Level	Solution
Private IP disclosure vulnerability	A total of 30 HTTP responses contained the private IP address 127.0.0.1 in the "via" response header. The first ten matching URLs are: - http://target01-wiser.xlab.si/icons/image2.gif - http://target01-wiser.xlab.si/dwa/images/lock.png - http://target01-wiser.xlab.si/dwa/css/ - http://target01-wiser.xlab.si/icons/folder.gif - http://target01-wiser.xlab.si/ - http://target01-wiser.xlab.si/icons/back.gif - http://target01-wiser.xlab.si/dwa/css/login.css - http://target01-wiser.xlab.si/dwa/js/ - http://target01-wiser.xlab.si/dwa/includes/dwaPage.inc.php - http://target01-wiser.xlab.si/dwa/images/RandomStom.png	Low	Private, or non-routable, IP addresses are generally used within a home or company network and are typically unknown to anyone outside of that network. Cyber-criminals will attempt to identify the private IP address range being used by their victim, to aid in collecting further information that could then lead to a possible compromise. The tool discovered that the affected page returned a RFC 1918 compliant private IP address and therefore could be revealing sensitive information. This finding typically requires manual verification to ensure the context is correct, as any private IP address within the HTML body will trigger it.
Directory indexing	The URL: "http://target01-wiser.xlab.si/dwa/" has a directory indexing vulnerability.	Low	Web servers permitting directory listing are typically used for sharing files. Directory listing allows the client to view a simple list of all the files and folders hosted on the web server. The client is then able to traverse each directory and download the files. Cyber-criminals will utilise the presence of directory listing to discover sensitive files, download protected content, or even just learn how the web application is structured. The tool discovered that the affected page permits directory listing.
Directory indexing	The URL: "http://target01-wiser.xlab.si/dwa/images/" has a directory indexing vulnerability.	Low	Web servers permitting directory listing are typically used for sharing files. Directory listing allows the client to view a simple list of all the files and folders hosted on the web server. The client is then able to traverse each directory and download the files. Cyber-criminals will utilise the presence of directory listing to discover sensitive files, download protected content, or even just learn how the web application is structured. The tool discovered that the affected page permits directory listing.
Directory indexing	The URL: "http://target01-wiser.xlab.si/dwa/includes/" has a directory indexing vulnerability.	Low	Web servers permitting directory listing are typically used for sharing files. Directory listing allows the client to view a simple list of all the files and folders hosted on the web server. The client is then able to traverse each directory and download the files. Cyber-criminals will utilise the presence of directory listing to discover sensitive files, download protected content, or even just learn how the web application is structured. The tool discovered that the affected page permits directory listing.
Directory indexing	The URL: "http://target01-wiser.xlab.si/dwa/js/" has a directory indexing vulnerability.	Low	Web servers permitting directory listing are typically used for sharing files. Directory listing allows the client to view a simple list of all the files and folders hosted on the web server. The client is then able to traverse each directory and download the files. Cyber-criminals will utilise the presence of directory listing to discover sensitive files, download protected content, or even just learn how the web application is structured. The tool discovered that the affected page permits directory listing.
Blank http response body	The URL: "http://target01-wiser.xlab.si/dwa/includes/dwaPhpIds.inc.php" returned an empty body, this could indicate an application error.	Information	
Blank http response body	The URL: "http://target01-wiser.xlab.si/dwa/includes/dwaPage.inc.php" returned an empty body, this could indicate an application error.	Information	
Directory indexing	The URL: "http://target01-wiser.xlab.si/dwa/includes/DBMS/" has a directory indexing vulnerability.	Low	Web servers permitting directory listing are typically used for sharing files. Directory listing allows the client to view a simple list of all the files and folders hosted on the web server. The client is then able to traverse each directory and download the files. Cyber-criminals will utilise the presence of directory listing to discover sensitive files, download protected content, or even just learn how the web application is structured. The tool discovered that the affected page permits directory listing.
Blank http response body	The URL: "http://target01-wiser.xlab.si/dwa/includes/DBMS/PGSQL.php" returned an empty body, this could indicate an application error.	Information	
Blank http response body	The URL: "http://target01-wiser.xlab.si/dwa/includes/DBMS/MySQL.php" returned an empty body, this could indicate an application error.	Information	
Click-jacking vulnerability	The whole target has no protection (X-Frame-Options header) against Click-Jacking attacks	Medium	Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages. The server didn't return an "X-Frame-Options" header which means that this website could be at risk of a clickjacking attack. The "X-Frame-Options" HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.

Figure 18. CyberWISER-Light vulnerability report shown on the user interface



### 3.1.3.2 Report for CyberWISER-Essential and CyberWISER-Plus

#### 3.1.3.2.1 Structure of a qualitative risk report

It is necessary to differentiate between the structure of the report produced for CyberWISER-Essential and that produced for CyberWISER-Plus.

In CyberWISER-Essential the structure is the one shown below. The possible qualitative levels are: very low, low, medium, high and very high.

Overall profile							Qualitative assessment [VL,L,M,H,MH]
	Section 1: name						Qualitative assessment [VL,L,M,H,MH]
		Risk WRPx-Ry: name					Qualitative assessment [VL,L,M,H,MH]
			Target T1: name	IP address T1	Short description T1	Complete description T1	Qualitative assessment [VL,L,M,H,MH]
			Target Tz: name	IP address Tz	Short description Tz	Complete description Tz	Qualitative assessment [VL,L,M,H,MH]
		Risk WRPx-Ry: name					
			Target T1: name	IP address T1	Short description T1	Complete description T1	Qualitative assessment [VL,L,M,H,MH]
			Target Tz: name	IP address Tz	Short description Tz	Complete description Tz	Qualitative assessment [VL,L,M,H,MH]
	Section 2: name						Qualitative assessment [VL,L,M,H,MH]
		Risk WRPx-Ry: name					Qualitative assessment [VL,L,M,H,MH]
			Target T1: name	IP address T1	Short description T1	Complete description T1	Qualitative assessment [VL,L,M,H,MH]
			Target Tz: name	IP address Tz	Short description Tz	Complete description Tz	Qualitative assessment [VL,L,M,H,MH]

Table 1. Structure of a qualitative report for CyberWISER-Essential

For CyberWISER-Plus, the structure adds the field specifying the port, since not only machines, but also applications can be assessed. The possible qualitative values are the same.

Overall profile								Qualitative assessment [VL,L,M,H,MH]
	Section 1: name							Qualitative assessment [VL,L,M,H,MH]
		Risk WRPx-Ry: name						Qualitative assessment [VL,L,M,H,MH]
			Target T1: name	IP address T1	Port T1	Short description T1	Complete description T1	Qualitative assessment [VL,L,M,H,MH]
			Target Tz: name	IP address Tz	Port Tz	Short description Tz	Complete description Tz	Qualitative assessment [VL,L,M,H,MH]
		Risk WRPx-Ry: name						Qualitative assessment [VL,L,M,H,MH]
			Target T1: name	IP address T1	Port T1	Short description T1	Complete description T1	Qualitative assessment [VL,L,M,H,MH]
			Target Tz: name	IP address Tz	Port Tz	Short description Tz	Complete description Tz	Qualitative assessment [VL,L,M,H,MH]
	Section 2: name							Qualitative assessment [VL,L,M,H,MH]
		Risk WRPx-Ry: name						Qualitative assessment [VL,L,M,H,MH]
			Target T1: name	IP address T1	Port T1	Short description T1	Complete description T1	Qualitative assessment [VL,L,M,H,MH]
			Target Tz: name	IP address Tz	Port Tz	Short description Tz	Complete description Tz	Qualitative assessment [VL,L,M,H,MH]

Table 2. Structure of a qualitative report for CyberWISER-Plus

As a sample, the figure below shows an example of report as it is displayed to the user on CyberWISER-Essential risk reporting dashboard.

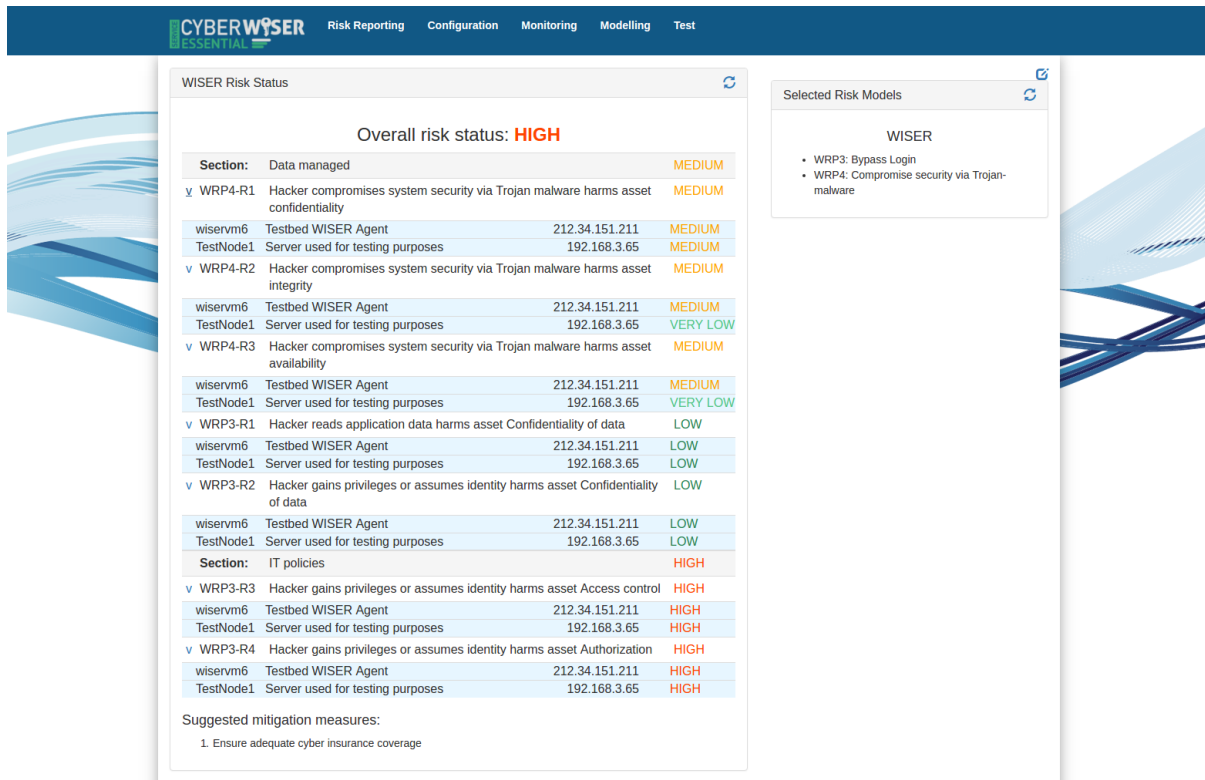


Figure 20. CyberWISER-Essential risk report sample

Next figure shows a sample of report displayed in CyberWISER-Plus<sup>6</sup>.

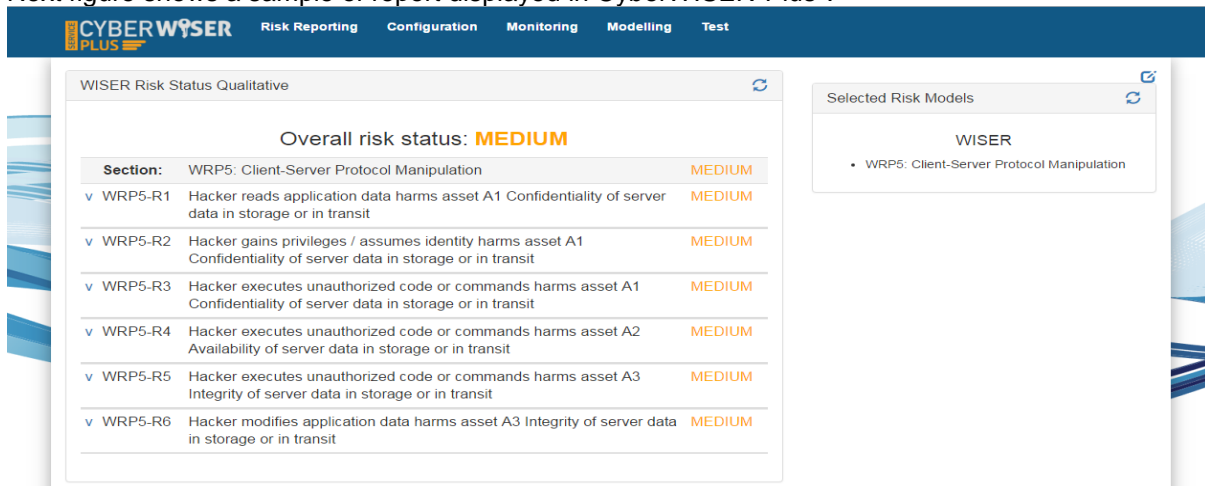


Figure 21. CyberWISER-Plus risk report sample

<sup>6</sup> What is expressed here is the risk exposure related to a hack. If the system detects (in the monitoring part) that the hack is actually happening the risk exposure calculated by the Risk Assessment Engine will increase

### 3.1.3.2.2 Structure of a quantitative risk report

In this case the structure follows a similar approach to the one described for the qualitative models. The difference lies in the way the risk assessment is performed. Quantitative models allow obtaining a more precise estimation of the risk, in terms of money. This is, the user is shown to which economic extent the cyber climate surrounding the company exposes the company.

Similarly, the structure changes slightly depending on working with CyberWISER-Essential or CyberWISER-Plus.

In the case of CyberWISER-Essential the structure is the one shown below.

Overall profile							Quantitative assessment [xxxxxx €]
	Section 1: name						Quantitative assessment [xxxxxx €]
		Risk WRPx-Ry: name					Quantitative assessment [xxxxxx €]
			Target T1: name	IP address T1	Short description T1	Complete description T1	Quantitative assessment [xxxxxx €]
			Target Tz: name	IP address Tz	Short description Tz	Complete description Tz	Quantitative assessment [xxxxxx €]
		Risk WRPx-Ry: name					Quantitative assessment [xxxxxx €]
			Target T1: name	IP address T1	Short description T1	Complete description T1	Quantitative assessment [xxxxxx €]
			Target Tz: name	IP address Tz	Short description Tz	Complete description Tz	Quantitative assessment [xxxxxx €]
	Section 2: name						Quantitative assessment [xxxxxx €]
		Risk WRPx-Ry: name					Quantitative assessment [xxxxxx €]
			Target T1: name	IP address T1	Short description T1	Complete description T1	Quantitative assessment [xxxxxx €]
			Target Tz: name	IP address Tz	Short description Tz	Complete description Tz	Quantitative assessment [xxxxxx €]

Table 3. Structure of a quantitative report for CyberWISER-Essential

Again, for CyberWISER-Plus, the structure adds the field specifying the port, since not only machines, but also applications can be assessed.

Overall profile								Quantitative assessment [xxxxxx €]
	Section 1: name							Quantitative assessment [xxxxxx €]
		Risk WRPx-Ry: name						Quantitative assessment [xxxxxx €]
			Target T1: name	IP address T1	Port T1	Short description T1	Complete description T1	Quantitative assessment [xxxxxx €]
			Target Tz: name	IP address Tz	Port Tz	Short description Tz	Complete description Tz	Quantitative assessment [xxxxxx €]
		Risk WRPx-Ry: name						Quantitative assessment [xxxxxx €]
			Target T1: name	IP address T1	Port T1	Short description T1	Complete description T1	Quantitative assessment [xxxxxx €]
			Target Tz: name	IP address Tz	Port Tz	Short description Tz	Complete description Tz	Quantitative assessment [xxxxxx €]
	Section 2: name							Quantitative assessment [xxxxxx €]
		Risk WRPx-Ry: name						Quantitative assessment [xxxxxx €]
			Target T1: name	IP address T1	Port T1	Short description T1	Complete description T1	Quantitative assessment [xxxxxx €]
			Target Tz: name	IP address Tz	Port Tz	Short description Tz	Complete description Tz	Quantitative assessment [xxxxxx €]

Table 4. Structure of a quantitative report for CyberWISER-Plus

The figures below show the visualization of a quantitative risk report. Figure 22 show the detail of the report and Figure 23 show a general view of the dashboard where the widget showing the quantitative risk report appears along with other widgets.

WISER Risk Status Quantitative <span style="float: right;">↻</span>			
<b>Overall risk status: 280000 €</b>			
<b>Section:</b>	Data managed		200000 €
v WRP4-R1	Hacker compromises system security via Trojan malware harms asset confidentiality		40000 €
wiservm6	Testbed WISER Agent	212.34.151.211	20000 €
TestNode1	Server used for testing purposes	192.168.3.65	20000 €
v WRP4-R2	Hacker compromises system security via Trojan malware harms asset integrity		40000 €
v WRP4-R3	Hacker compromises system security via Trojan malware harms asset availability		40000 €
v WRP3-R1	Hacker reads application data harms asset Confidentiality of data		40000 €
v WRP3-R2	Hacker gains privileges or assumes identity harms asset Confidentiality of data		40000 €
<b>Section:</b>	IT policies		80000 €
v WRP3-R3	Hacker gains privileges or assumes identity harms asset Access control		40000 €
v WRP3-R4	Hacker gains privileges or assumes identity harms asset Authorization		40000 €
<b>Suggested mitigation measures:</b>			
1. Ensure adequate cyber insurance coverage ⓘ			

Figure 22. Example of quantitative risk report



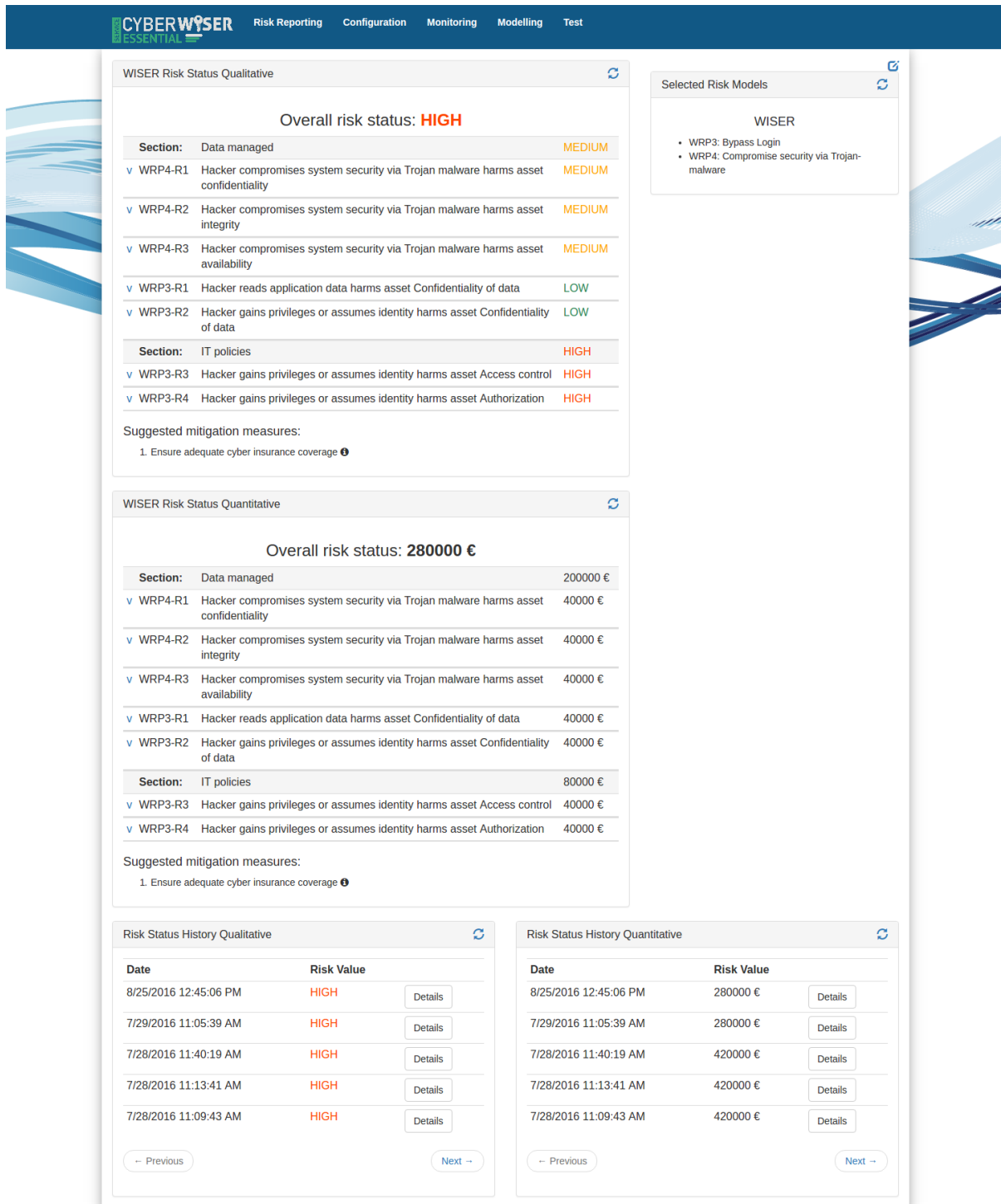


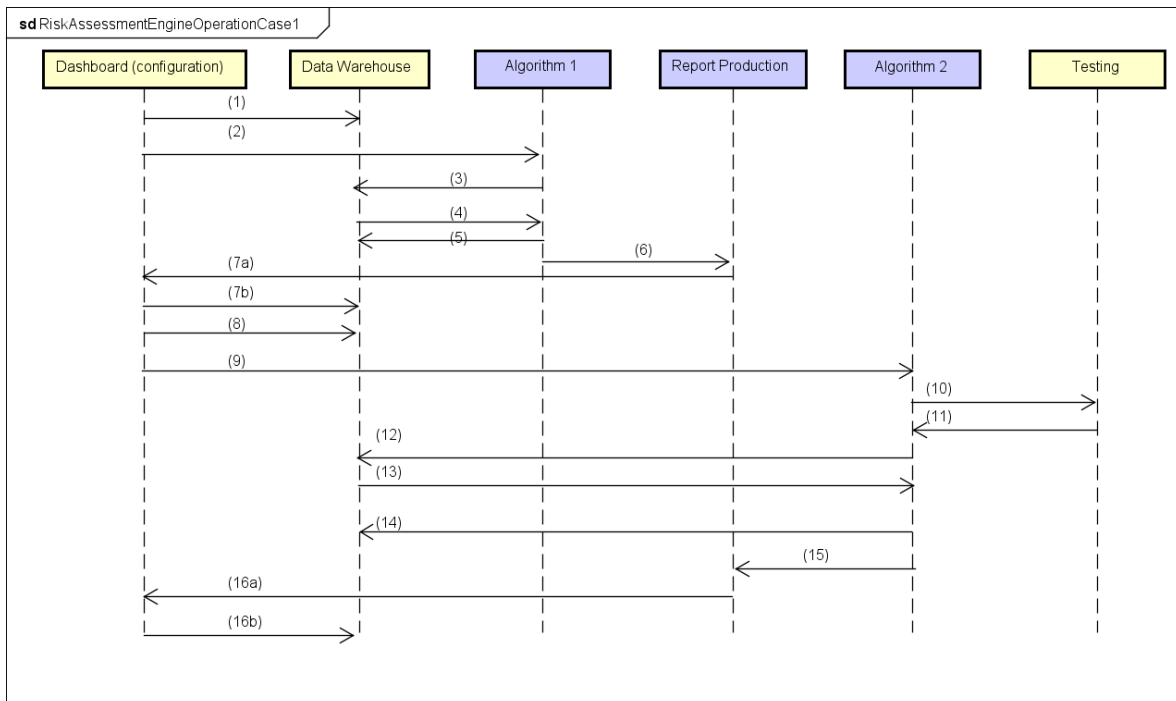
Figure 23. General view of the dashboard with widgets for the current qualitative and quantitative assessments and the lists of previous assessments

### 3.1.4 Triggering cases

The Risk Assessment Engine does not continuously evaluate the risk assessment of an organisation, but this is triggered only when something relevant happens. In the following, we explain how the triggering works. This description is an updated version of that given in section 3.2.4 of D5.1. The possible triggering events are enumerated below:

- 1) **CyberWISER-Light.** This is triggered on demand by the user, when the questionnaire is completed of a new vulnerability scan is launched.
- 2) **CyberWISER-Essential and CyberWISER-Plus.** Four cases are distinguished here:
  - a. There is a change in the business indicators. The user changes the answer provided in some question/s, thus changing in the value of some business indicator/s (see Figure 6, arrows (1)(2) and (7)).
  - b. There is a change in the selected model/s. The user goes to the modelling selection interface and selects different model/s to work with, and confirms such selection. This is detected, what launches a new assessment of the risk with the algorithm (see Figure 6, arrow (8)).
  - c. There is a change in the vulnerability indicators. After performing a vulnerability scan and interpreting the results, the value of some indicator/s produced by the 'Indicators value generator' changes and this is detected, meaning that the algorithm is launched (see Figure 6, arrows (4) and (7)).
  - d. There is a change in events and alarms. This may mean a change in the value of network layer indicators or application layer indicators. When this is detected, the algorithm is launched (see Figure 6, arrows (3) and (7)).

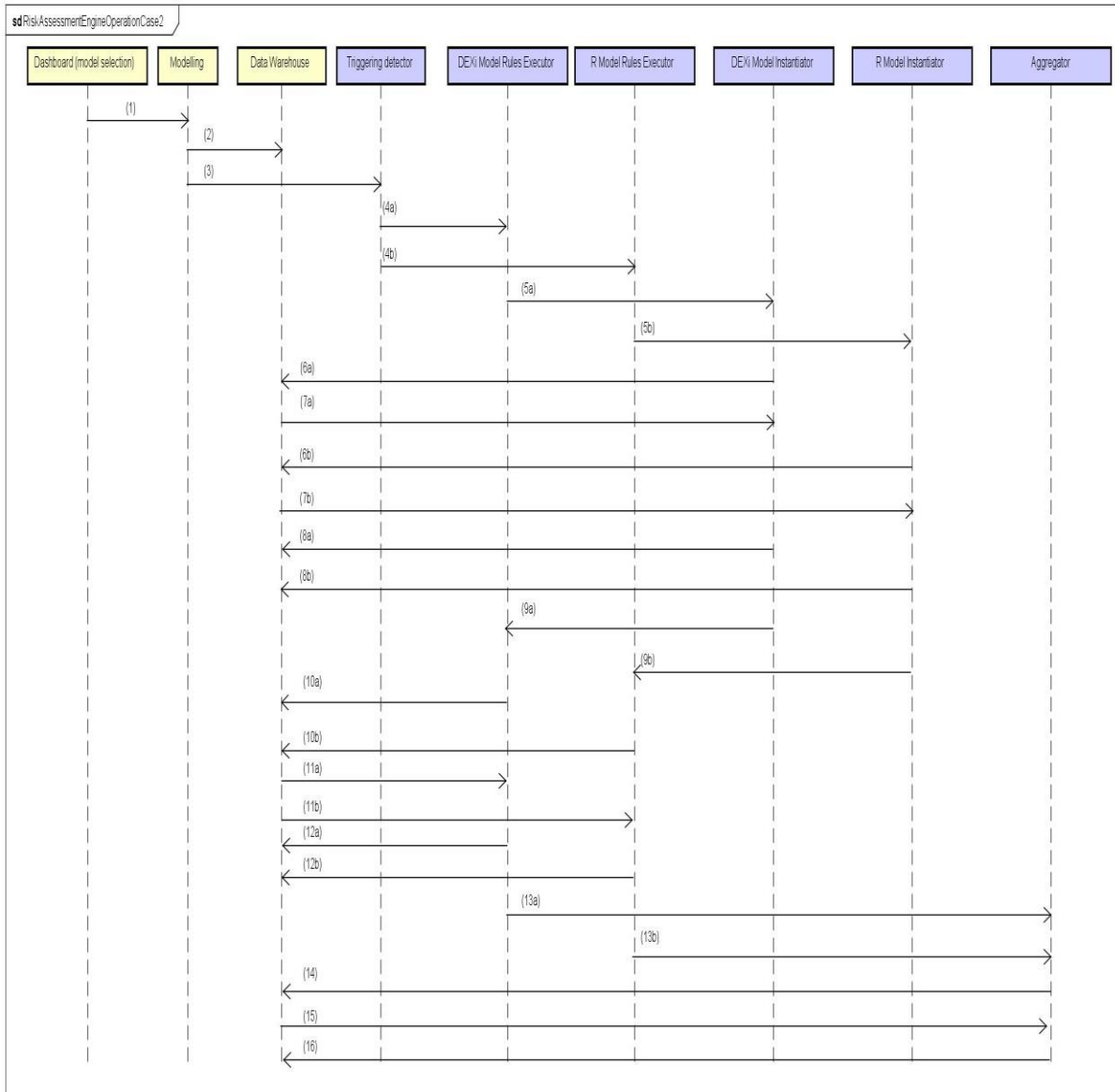
In CyberWISER-Essential, as depicted in Figure 6, the component called 'Triggering detector' checks whether there was any relevant change in the indicators and, if there was, launches the algorithm associated to the model.



powered by Astah

Figure 24. Risk Assessment Engine operation case 1. Triggering demanded by the user

Figure 24 reflects the sequence diagram related to the operation of the Risk Assessment Engine in the case it is triggered on demand by the user. This is specific for the CyberWISER-Light. The inner components of the Risk Assessment Engine appear marked in blue, while the outers are marked in yellow. The user, by means of the configuration section of the dashboard, uses the questionnaire to introduce the business variables, which are stored in the Data Warehouse (1). Once the questionnaire is filled out, the user launches the Algorithm 1 (2). When the algorithm is launched, it retrieves from the Data Warehouse the variables introduced by the user (3)(4). It goes through the scoring and weighting algorithm and, basing on that, produces the results which are stored in the Data Warehouse (5). Then, the Report Production module is called (6), which uses the output of the algorithm to issue the report which is shown to the user by means of the dashboard (7a). The user will have the chance to store it on the Data Warehouse (7b). Then, the user introduces the information related to the infrastructure elements and their value in terms of confidentiality, integrity, and availability, which is stored in the Data Warehouse (8), and calls the Algorithm 2 (9). This calls in turn to the Vulnerability Scanners (10) and collects the results of executing them (11). Then, Algorithm 2 retrieves from the Data Warehouse (12)(13) the information the user provided in (8). Then, this information and that obtained with the vulnerability scanners are correlated and the result of the algorithm is stored in the Data Warehouse (14). The report production module is then called (15) and the report is shown to the user (16a). The user has the chance to store the report in the Data Warehouse (16b).

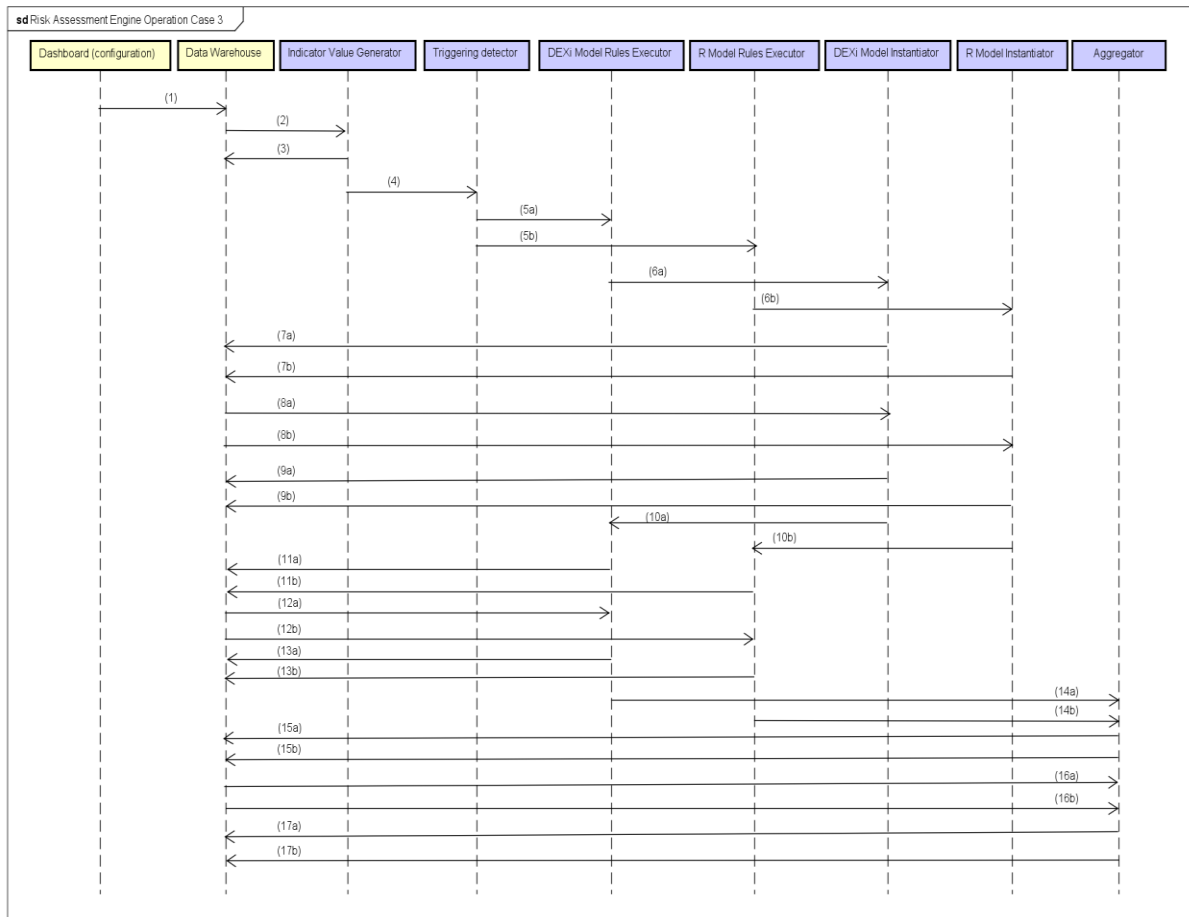


powered by Actix

Figure 25. Risk Assessment Engine operation case 2. Change in model

Figure 25 represents the functioning of the Risk Assessment Engine when there is a change in the model/s used to produce the risk assessment report. This is specific of CyberWISER-Essential and CyberWISER-Plus. This explanation is based on the assumption that a risk model is composed of both a quantitative and a qualitative model. First, the user selects new model/s to base the risk evaluation on. To do so, he interacts with the Modelling module by means of the Dashboard (1). The reference to the active model is stored in the Data Warehouse (2). The Modelling module informs the triggering detector, by means of an event, about the change in the model/s (3). This module passes the control to the DEXi Model Rules Executor and the R Model Rules Executor (4a)(4b), in charge of carrying out the qualitative and quantitative evaluations respectively. Then, the DEXi Model Instantiator and the R Model instantiator are respectively called (5a)(5b). Both instantiators retrieve from the Data Warehouse the model/s being used and the indicators to populate these models. The former was provided by the Modelling module and the latter by the indicator values generator (6a)(7a)(6b)(7b). Then, the instantiators produce the different model instances and store them in the Data Warehouse (8a)(8b). Once this is done, each instantiator gives back the control to the pair

Model Rules Executor (9a)(9b). Both Model Rules Executors retrieve the model instances (10a)(10b)(11a)(11b), perform the calculations aimed at obtaining the risk assessment report, and store the results in the Data Warehouse (12a)(12b). A report per infrastructure element is produced. Then, both DEXi and R Model Rules Executor call the Aggregator (13a)(13b). The Aggregator retrieves the reports produced for each of the infrastructure elements (14)(15) and produces the global reports (both qualitative and quantitative) where the risk associated to the infrastructure as a whole and the likely mitigation measures are reflected (16).

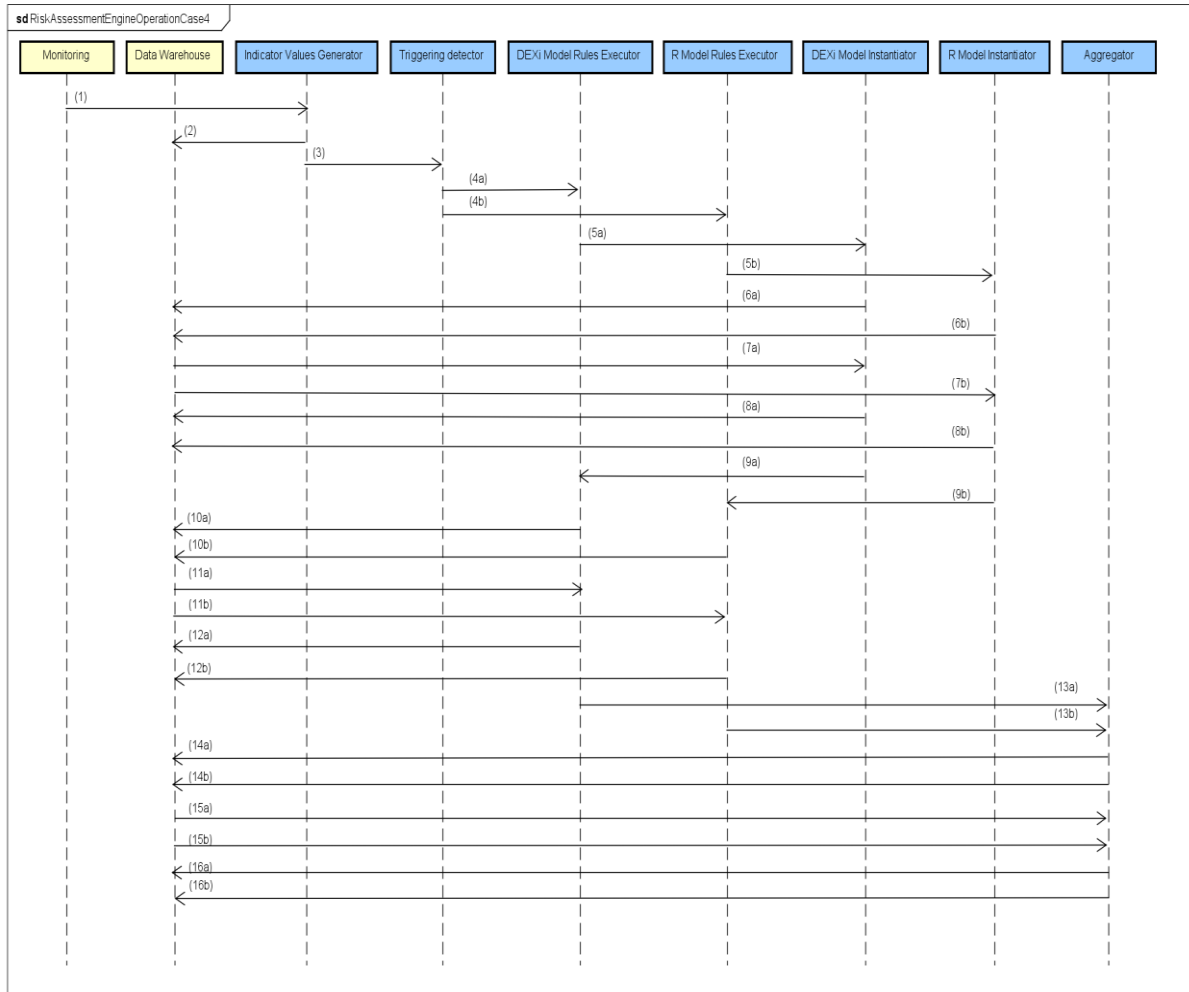


powered by Astah

Figure 26. Risk Assessment Engine operation case 3. Change in business variables

Figure 26 represents the functioning of the Risk Assessment Engine when this is triggered because of a change in the business variables. The user, by means of the Dashboard, changes something in the configuration and this change is stored in the Data Warehouse (1). By means of the events message queue, the indicator values generator is informed about the change in the configuration (2). The indicator values generator is executed to produce the values of the indicators and they are stored in the Data Warehouse (3). Once this is done, the indicator value generator notifies the triggering detector (4) to which the control is passed. In turn, the triggering detector passes the control to the DEXi Model Rules Executor and the R Model Rules Executor (5a)(5b). In turn, they respectively call the DEXi and R model instantiators (6a)(6b). Both instantiators retrieve from the Data Warehouse the current indicator values along with the models to populate (7a)(7b)(8a)(8b) and produce the instances of the qualitative and quantitative models, which are in turn stored in the Data Warehouse (9a)(9b). Then, both instantiators return the control to their respective Model Rules Executors (10a)(10b). Both executors retrieve their respective model instances (11a)(11b)(12a)(12b) and compute the qualitative

and quantitative risk evaluation per infrastructure target, storing this information in the Data Warehouse (13a)(13b). The aggregator is then called by each if the executors (14a)(14b). It retrieves the qualitative and quantitative risk reports per target (15a)(15b)(16a)(16b), and obtains the aggregated reports addressing the infrastructure as a whole, and stores them in the Data Warehouse (17a)(17b).

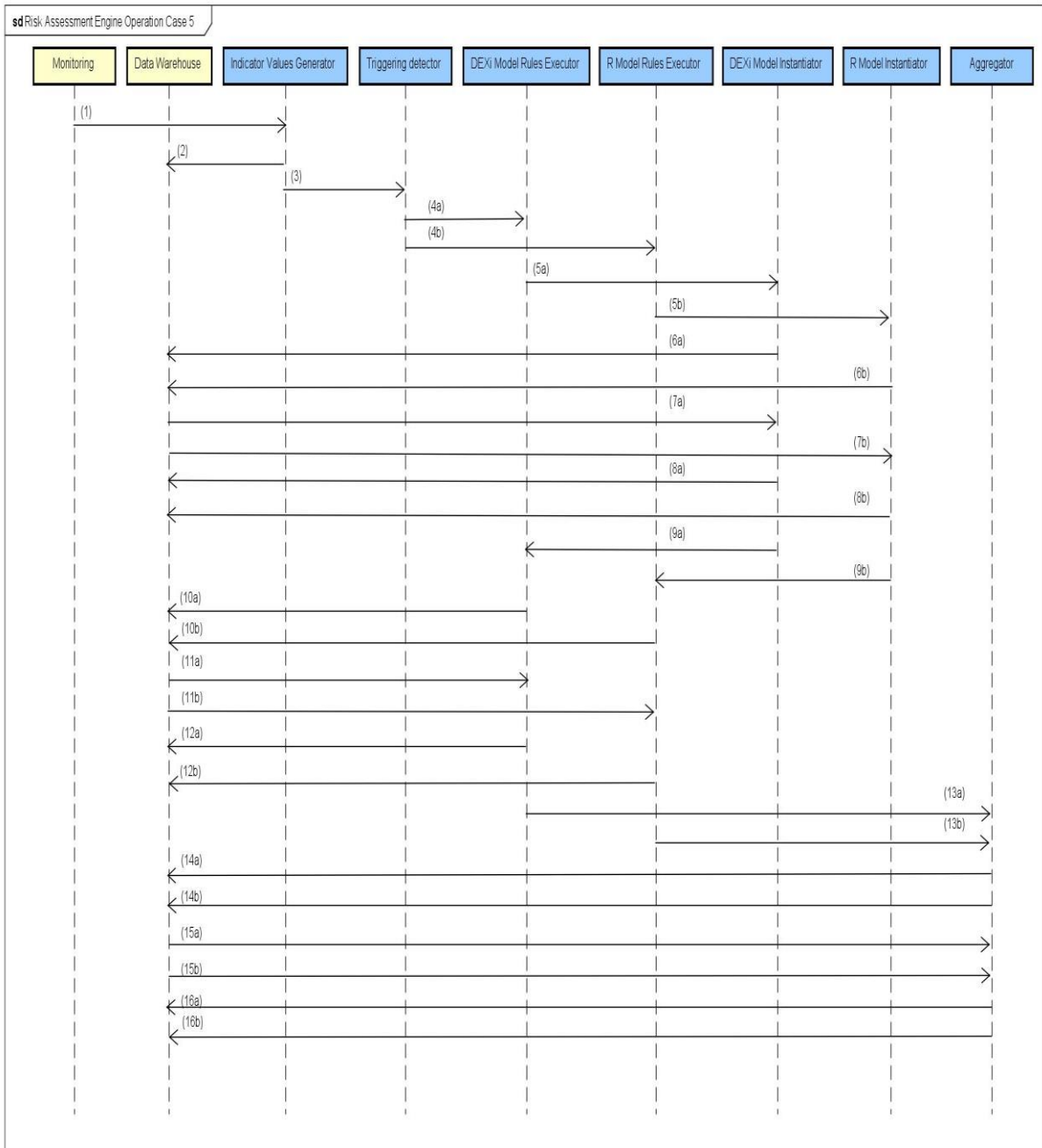


powered by Astah

Figure 27. Risk Assessment Engine operation case 4. Change in monitoring variables

Figure 27 shows the operation when the Risk Assessment Engine detects a change in the monitoring variables. The sequence is the same except for the fact that this time the information triggering the Engine comes from the Monitoring module, as a consequence of the evolution of the cyber climate. This time the Dashboard (the user interface) does not play a relevant role.

The same case applies when the change comes from the testing variables, i.e., vulnerabilities detected lead to a change in the indicators which are evaluated by the triggering detector. This is illustrated in the figure below.



powered by Astah

Figure 28. Risk Assessment Engine operation case 5. Change in testing variables

### 3.1.5 Other considerations

Indicators serve as input to risk models to support the automated assessment of risk levels. Indicators are obtained from questions posed through the Configuration module, from Monitoring of the target infrastructure through the Monitoring module, and from vulnerability testing through the testing module. D5.1, section 3.5 provides details on how the indicators are obtained for each case.

These indicators feed the inner algorithm of the Risk Assessment Engine. This algorithm represents the model rules that define how to compute the risk levels and the mitigation proposals from the

different indicators received. These rules are executed in run-time by both the DEXi and the R Model Rules Executors (presented in Section 3.1.1 and its implementation being addressed in Sections 3.2.3 and 3.2.5). The DEXi and R Model Rules Executors take, as input, the instances of the risk models (Model rules) associated with each target, including the updated indicator values obtained through the business configuration, monitoring and testing of the target in question. As output, they provide qualitative and quantitative risk level assessments for each of the risks identified in the relevant models, as well as the corresponding proposals for mitigation options. The Triggering detector determines when the Model rule executor runs the algorithms. More details about the Inner algorithm can be found in D5.1, Section 3.6.

## 3.2 Implementation

The Risk Assessment Engine is implemented as a Python application with the following modules that will be described in detail below:

- IndicatorValueGenerator
- TriggeringDetector
- DEXIModelInstantiator
- DEXIModelRulesExecutor
- RModelRulesExecutor
- RModelInstantiator
- Aggregator

Initially, the Risk Assessment Engine requests an authentication token to be used by the different modules to communicate with the Data Warehouse and retrieve or store data. This token will be refreshed periodically when it expires.

### 3.2.1 Indicator Value Generator

The Indicator Value Generator module is in charge of receiving the different inputs required for the generation of the indicators values used in the Risk Assessment Engine and storing those indicator values in the Data Warehouse for its usage by the rest of modules.

For this reason, the Indicator Value Generator establishes connection with the following RabbitMQ Server queues using a certificate file created specifically for this module and that we will identify with the prefix “wiser-rae-ivg”:

- **rae.ivg-dwhnotifs**: this queue is used to receive notifications from the Data Warehouse when a record is created, updated or deleted in the tables related to the configuration inputs. These tables are /config/company\_profiles for the business company profile as a whole and /rae/targets for the specific targets in the infrastructure.
- **rae.ivg-events**: this queue is used to receive in real-time events from the monitoring infrastructure.
- **rae.ivg-alarms**: this queue is used to receive in real-time alarms from the monitoring infrastructure.
- **rae.ivg-vulnreports**: this queue is used to receive vulnerabilities reports from scans done in targets of the monitored infrastructure.

The indicator values generated by the Indicator Value Generator module will be stored in the Data Warehouse in the table /rae/indicator\_values.



On the other hand, initially the Indicator Value Generator will retrieve from the Data Warehouse the information related to all the organizations and targets registered as well as all the indicators stored in DWH Catalogue (table /rae/indicators/) to have this data already available for the generation of the indicator values, instead of accessing to the DWH each time they are required. Besides, this information is used to initialize the network and testing indicators values associated to the existent targets to its default value ('No', to indicate that initially no events/alerts/vulnerabilities have been detected in the target).

A different thread is used to deal with each of the tasks performed by the Indicator Value Generator:

### 1. ConsumerCompanyInfo

This thread is responsible for the generation of the business indicators. It receives notifications from the Data Warehouse using the queue *rae.ivg-dwhnotifs* when there is a change in a Target (/rae/targets) or in a CompanyProfile (/config/company\_profiles). The format of the notifications received has been already described in section 3.1.2.1.

Each target defined and configured by the user will have associated three business indicators related to the security assets defined in WISER: Confidentiality, Integrity, and Availability. The score values assigned to each of the assets will be mapped by the Indicator Value Generator module to the following indicator values: Very Low (1-2), Low (3-4), Medium (5-6), High (7-8) or Very high (9-10). These indicators do not have a rule associated and they can be identified from other business indicators because the means is "asset". Below it is shown how these indicators are defined in the indicator catalogue stored in the DWH (/rae/indicators):

```
{ "id":1,"question":"IN-C7:Confidentiality","data_type":"string","motivation":"Impact of Confidentiality security asset","indicator_type":1,"means":"asset","rule":""}  
{ "id":2,"question":"IN-C8:Integrity","data_type":"string","motivation":"Impact of Integrity security asset","indicator_type":1,"means":"asset","rule":""}  
{ "id":3,"question":"IN-C9:Availability","data_type":"string","motivation":"Impact of Availability security asset","indicator_type":1,"means":"asset","rule":""}
```

Consequently, each time it is notified that a new target has been created, the Indicator Value Generator will store in the DWH a new entry in the table /rae/indicator\_values/ by each of those assets with the value of the score assigned to it. And when it is received a notification due to a change in some of the assets, it will be checked the stored indicator value and updated with the new one if required. Additionally, when it is received the notification for a new target created, it is also initialized with a default value the network and testing indicator values associated to that target.

The information about the company profile, completed by the user through the questionnaire, will have associated different business indicators, each of them related to some of answers provided by the user. The field 'means' of these business indicators in the DWH will have the value 'questionnaire'. The answers provided by the user will be mapped by the Indicator Value Generator to a Boolean value according to the rules defined in the business indicators. Below it is included an example to show how these indicators are defined in the indicator catalogue stored in the DWH (/rae/indicators):

```
{ "id":10,"question":"IN-C1:Does your company maintain and process restricted information?","data_type":"boolean","motivation":"Evaluation the sensitivity level of the information your company maintains and processes.","indicator_type":1,"means":"questionnaire","rule":"q14=0"}
```

In this example, a value of 'Yes' (True) associated to the business indicator IN-C1 is stored in the DWH when the user selects the first available option (0) in the q14 of the questionnaire.

NOTE: In the current version of the Risk Assessment Engine, the rules for the business indicators have to be defined with the format "q<question\_number>=<option\_number>".

Each time a new company profile is created, the Indicator Value Generator will store a new entry in the DWH table `/rae/indicator_values/` corresponding to each of those business indicators. When a notification, due to a change in an existent company profile, is received, then, it will be checked which questions have been updated in order to determine the associated rules and update the corresponding indicator values. Since each indicator value is associated to a target, these business indicators will be associated to a generic target with ip address "0.0.0.0" created automatically for each organization. This generic target will not be considered when the generation of business indicators is associated to the security assets.

## 2. ConsumerAlarms

This thread is responsible for the generation of the network indicators when an alarm is received from the monitoring infrastructure.

The network indicators associated to alarms are identified in the DWH indicator catalogue (`/rae/indicators`) from the ones associated to events because the means is "alarm". These indicators have a rule associated with the condition to be checked by the Indicator Value Generator to determine the indicator value. It will be 'Yes' (True) in case the incoming alarm matches the condition included in the rule. Below it is shown an example of network indicator associated to an alarm stored in the DWH indicator catalogue:

```
{ "id":7, "question": "IN-11: Is there a malware-Trojan inside the infrastructure?", "data_type": "boolean", "motivation": "It has been detected malware signatures in the client's infrastructure network traffic.", "indicator_type": 3, "means": "alarm", "rule": "PLUGIN_ID=70000 AND PLUGIN_SID IN (40000:44999)" }
```

The alarms are received from the RabbitMQ queue `rae.ivg-alarms`. First, it is checked if it affects some of the targets registered in the DWH retrieving from the incoming message the fields `DST_IP` (destination ip address), `SRC_IP` (source ip address) and `ORGANIZATION`.

In case the alarm has been generated in the monitored infrastructure, the rules included in the network indicators associated to alarms will be evaluated. When the fields included in the alarms match the rule included in an indicator, the Indicator Value Generator will update the entry associated to that indicator for that specific target in the `/rae/indicator_values/` table with a value of 'Yes' (True).

In the current version of the Risk Assessment Engine, the rules for the network indicators associated to alarms can include the following fields:

- `PLUGIN_ID`: identifier of the data source generating the alarm (as defined in the Monitoring Engine)
- `PLUGIN_SID`: identifier of the type of event for a specific data source generating the alarm (as defined in the Monitoring Engine)
- `DST_IP`: destination ip address
- `SRC_IP`: source ip address
- `DST_PORT`: destination port
- `SRC_PORT`: source port

The conditions can be defined with "=" to indicate a specific value or "IN (value1:value2)" to define a range of values. It is also possible to include several conditions joined by the word "AND".

## 3. ConsumerEvents

This thread is responsible for the generation of the network indicators when an event is received from the monitoring infrastructure.

The network indicators associated to events will have their 'means' field assigned the value 'event' in the DWH indicator catalogue. They are defined in a similar way to the explained in the

ConsumerAlarms using the field “rule” to define the condition to be checked by the Indicator Value Generator. Below it is shown an example of network indicator associated to an event stored in the DWH indicator catalogue.

```
{"id":6,"question":"Network scan","data_type":"boolean","motivation":"It has been detected a network reconnaissance attempt","indicator_type":3,"means":"event","rule":"plugin_id=1001 AND plugin_sid IN (2010930:2010940)"}
```

The events are received from the RabbitMQ queue *rae.ivg-events*. It is checked if it affects some of the targets registered in the DWH retrieving from the incoming message the fields *dst\_ip* (destination ip address), *src\_ip* (source ip address) and organization.

#### 4. ConsumerVulnerabilities

This thread is responsible for the generation of the testing indicators when a vulnerability report has been received after the execution of a vulnerability scan.

The vulnerability reports are received from the RabbitMQ queue *rae.ivg-vulnreports*. First, the Indicator Value Generator has to recover the list of vulnerabilities which are included in the *binary\_data* (coded base64). For each vulnerability report, it will be checked if it has been generated in one of the targets registered to be processed.

Each vulnerability included in the report will be checked against the testing indicators stored in the DWH indicator catalogue (*/rae/indicators*). The means of these testing indicators is “vulnerability” and they also have a rule with the condition to be verified by the Indicator Value Generator. Below it is shown an example of testing indicator associated to a vulnerability stored in the DWH indicator catalogue.

```
{"id":9,"question":"IN-20:Does the server use outdated authentication schemes?","data_type":"boolean","motivation":"Using outdated authentication schemes or mechanisms may expose the target system to authentication attacks such as reflection attacks in authentication protocol.","indicator_type":2,"means":"vulnerability","rule":"w_risk_level=100 AND short_desc IN (Basic HTTP credentials,NTLM authentication,HTTP Basic authentication,Guessable credentials)"}
```

In the current version of the Risk Assessment Engine, the rules for the testing indicators associated to vulnerabilities can include the following fields:

- *W\_risk\_level*: risk level included in the vulnerability report
- *Short\_desc*: short description of the vulnerability included in the report

The conditions can be defined with “=” to indicate a specific value or “IN (value1:value2)” to define a range of values. It is also possible to include several conditions joined by the word “AND”.

When the fields included in one of the vulnerabilities received in the report received match the rule included in a testing indicator, the Indicator Value Generator will update the entry associated to that indicator for that specific target in the */rae/indicator\_values/* table with a value of ‘Yes’ (True).

#### 5. ResetIndicatorValues

Periodically, this thread checks the timestamp of the indicator values stored in the DWH which correspond to no business indicators and will reset them to the default value (‘No’) in case they are obsolete.

The parameter *resetProcess* included in the configuration file *config.json* indicates how often this checking is done. The timeout to determine when a timestamp can be considered obsolete is

---

defined by the parameter *timeoutIndicatorValues* (also configured in the file *config.json*).

### 3.2.2 Triggering detector

The Triggering Detector is responsible for launching the evaluation of the different risk assessment algorithms when it is detected a change in an indicator or in a risk model. Besides, before launching the algorithms, this module is in charge of registering the new risk assessment report to be produced storing a new entry in the DWH (*/rae/risk\_reports*). The id of this report will be later used by the Model Rules Executor and Aggregator modules to store the results of the risk assessment.

Consequently, this module starts establishing a connection with the RabbitMQ server to receive notifications about changes in the DWH. In particular, changes in the *IndicatorValues*, *RiskModel* and *SelectedRiskModel* will be notified to the Triggering Detector using “*rae.td-dwhnotifs*” queue.

When a notification is received, depending on the type of message received a different thread is started for its processing. There are two types of threads:

#### 1. ConsumerIndicatorChange

This thread will deal with the notifications received related to a change in an indicator value. First, when an indicator change is received, it is recovered the new indicator value from the Dataware House (*/rae/indicator\_values*). The generic information about that indicator is searched in DWH indicator catalogue (*/rae/indicators*) using as ide the field ‘indicator’ retrieved from the */rae/indicator\_values*.

Depending on the type of indicator and the type of action notified, it will be done a different processing to determine if the algorithms need to be triggered. Testing, network, and application indicators only will be processed for specific targets (different from the generic target with ip address “0.0.0.0” only associated to the company profile business indicators). The different triggering cases, depending on the inputs received, have been already described with more detail in section 3.1.4.

The Triggering Detector module makes use of two timeouts configured in the file *config.json* to wait for more indicator change notifications before starting the processing: *timeoutTargets* and *timeoutTriggeringDetector*.

The first one (*timeoutTargets*) is used to wait for the three notifications associated to the business indicators related to the security assets defined in WISER (Confidentiality, Integrity and Availability). When it is received a notification with the action “CREATED” for an object Target, the ConsumerIndicatorChange thread will wait *timeoutTargets* seconds for new notifications associated to the same specific target before launching the algorithms. If the target in the notification has the ip address “0.0.0.0”, it will not be processed since it represents that a new company profile has been registered and consequently it will be also received a notification with the action “CREATED” but for an object CompanyProfile.

In any other case, it will be waited for new notifications the time indicated in the parameter *timeout TriggeringDetector*, through the use of flags associated to each organization. In this way, it is avoided the successive execution of the risk assessment algorithms when it is expected several indicator notifications can arrive in a short period of time (for example in case of changes in the company profile

The risk models for the specific organization where the indicator has been changed are retrieved from DWH using the endpoint */config/selected\_risk\_models*. If there is no selected risk model for the organization where the indicator notification has been received, no processing will be done by the Triggering Detector.

For each risk model selected by the organization, the Triggering Detector checks if the updated indicators received from the DWH notifications are included in the model using the field

'indicators' included in the DWH risk models catalogue (*/modelling/risk\_models*). The *risk\_model* recovered from the DWH is a JSON object like the following one:

```
"risk_model":{
  "coras_model": "cm",
  "dexi_model": "<DEXi> .... </DEXi>",
  "r_model": "rm",
  "relevance_criteria": "test",
  "indicators":[1,2,3,7],
  "risks":[3,4,5]}
```

The Triggering Detector will launch, for each of the risk models selected in the organization where some indicator has been changed, the execution of the qualitative (DEXi) and quantitative (R) algorithms invoking the method *execute\_DEXI\_model(data)* provided by the *DEXIModelRulesExecutor* module and *execute\_R\_model(data)* provided by the *RModelRulesExecutor*, respectively.

The algorithms will be launched by each of the specific targets associated to the indicator values modified. If the ip address of the target associated to the indicator is the "0.0.0.0" (infrastructure as a whole), the model will be evaluated by each of the specific targets in the infrastructure.

## 2. ConsumerModelChange

This thread will deal with the notifications received related to a change in a risk model. If it is a "**RiskModel**" change notification, then it is recovered from the DWH (*/config/selected\_risk\_models*) the list of risk models selected by each of the organizations and it is checked which ones are using that updated risk model. If it is a "**SelectedRiskModel**" change notification, it is directly recovered from the DWH (*/config/selected\_risk\_models*) the specific selected risk model created or updated.

For each risk model selected by an organization affected by the change, the Triggering Detector checks if the indicators included in that risk model in the DWH risk models catalogue (field "indicators" in */modelling/risk\_models*) already have an input with the indicator value stored in the DWH (*/rae/indicator\_values*). If not, the Triggering Detector will store them with the default value ('No' for network, testing and application indicators).

Finally, the Triggering Detector will invoke, for each selected model affected selected by an organization and for each target in that organization (different from the generic target with ip address "0.0.0.0"), the evaluation of the risk assessment qualitative (DEXi) and quantitative (R) algorithms.

### 3.2.3 DEXi Model Rules Executor

The DEXi Model Rules Executor is the Risk Assessment Engine module in charge of the execution the qualitative risk assessment algorithm through the evaluation of the DEXi model file defined for a specific risk model and a specific target in an organization infrastructure.

This module is invoked by the Triggering Detector module when it has been detected changes in the indicator values or in the risk models. The information provided by the Triggering Detector includes:

- the id of the risk assessment report where the result of the evaluation will be registered,
- the id of the organization,
- the id of the risk model to be evaluated and

- the id of the specific target whose indicator values will be used in the evaluation.

The JSON with this data provided is shown below:

```
data = {  
    'report': 1,  
    'organization':1,  
    'risk_model': 1,  
    'target': 5  
}
```

Each time the evaluation of a DEXi model is invoked by the Triggering Detector, it will be started a different thread *DEXiRulesExecutor* to do the processing. The steps followed by this thread are the following:

- First, the *DEXiModelInstantiator* is invoked to do the instantiation of the risk model for the specific target. See in section 3.2.4 how this module works. The output of that instantiation contains the DEXi model XML file to be evaluated and a string with the indicators required for that model and the values stored in the Data Warehouse for the specific target. For example:

```
dexi_instantiator_output = [{  
    'dexi_model': '<DEXi>....</DEXi>',  
    'indicators_instance':'IN-C7=High;IN-C8=High;IN-C9=High;IN-11_WRP4-R1=No;IN-11_WRP4-R2=No;IN-11_WRP4-R3=No;DummyNode=DummyValue1'  
}]
```

- Then it is executed the DEXi algorithm to perform the qualitative risk assessment using the open source Java library *JDEXiEval.jar* (<http://kt.ijs.si/MarkoBohanec/jdexi.html>). This library needs as arguments the dexi file (.dexi) with the model to be evaluated and the list with the set of indicators included in that model and their values for the target evaluated (a ';' separated list of name=value pairs). For example:

```
java -jar JDEXi3Eval.jar WRP-4.dexi "IN-11=No;IN-C7=High;IN-C8=High;IN-C9=Medium;DummyNode=DummyValue1"
```

It is important to remark that before running the Java DEXi evaluator it is required to have previously installed the Oracle JDK and correctly defined in the environment the variables *JAVA\_HOME* and *PATH*.

The execution of that DEXi java evaluator generates the following output:

```
'Output values': WRP4-M1=No;WRP4-R1=Medium;!_U1=Low;!_S1_to_U1=Low;!_S1=Low;cl_S1_to_U1=High;c_U1_A1=High;WRP4-
```

```
R2=Very low;l_U1=Low;l_S1_to_U1=Low;l_S1=Low;cl_S1_to_U1=High;c_U1_A2=Very  
low;WRP4-R3=Very  
low;l_U1=Low;l_S1_to_U1=Low;l_S1=Low;cl_S1_to_U1=High;c_U1_A3=Very low
```

- The DEXi files containing WISER risk models assess the risk level for one or more risks (up to 8). In the above example, the file WRP-4.dxi evaluates the qualitative risk level of the following three risks:
  - WRP4-R1: Hacker compromises system security via Trojan malware harms confidentiality asset;
  - WRP4-R2: Hacker compromises system security via Trojan malware harms integrity asset;
  - WRP4-R3: Hacker compromises system security via Trojan malware harms availability asset.

The risks included in each dexi model are defined in the field 'risks' for each risk model registered in the DWH risk models catalogue (*/modelling/risk\_models*). The information about each risk included in the risk model is stored in the DWH risks catalogue (*/rae/risks*) and can be retrieve using a specific risk id. Below it is shown an example of a risk stored in the DWH:

```
{"id":3,"short_description":"WRP4-R1","detailed_description":"Hacker reads application data  
harms asset confidentiality of data","sections":[3],"mitigation_measures":[]},
```

Each Risk has a *short\_description* (e.g. »WRP4-R1«) which is the name that will be searched by the DEXIRulesExecutor in the Output values to retrieve the qualitative assessment for that risk. Besides, each Risk has a field *sections* (related to the different parts of the risk report, registered in the DWH in */rae/sections*) and a field *mitigation\_measures* (that are also registered in the DWH in */rae/mitigation\_measures*).

- For each risk in the model evaluated, it is stored in the DWH (*/rae/risk\_per\_target*) what is the qualitative assessment associated to the evaluated target. The quantitative assessment will be stored by the R Model Rules Executor. Below it is shown an example of a risk per target qualitative assesement stored in the DWH:

```
{"id":40,"report":55,"risk_model":4,"risk":5,"target":48,"qualitative_assessment":"very high",  
"quantitative_assessment":"low"}
```

- Finally, it is invoked the Aggregator module (see section 3.2.7).

### 3.2.4 DEXi Model instantiator

DEXi Model Instantiator module of the Risk Assessment Engine offers a method *instantiate\_DEXi\_model(data)* which is invoked by the DEXi Model Rules Executor to do the instantiation of a risk model selected for a specific target.

The steps followed when this module is invoked are:

- All the indicator values stored in the Data Warehouse are retrieved for the specific target evaluated. These values are recovered using the endpoint `/rae/indicator_values` and filtering for the specific target id provided by the DEXi Rules Executor in the data of the request received.

In addition, all the indicator values associated to the organization where the specific target belongs to are retrieved. These indicator values correspond with business indicators not associated to a specific target but to the organization as a whole (ip address of the target is "0.0.0.0").

- The information about the risk model to be instantiated is retrieved from Data Warehouse using the endpoint `/modelling/risk_models` for the specific risk model id provided by the DEXi Rules Executor in the request.

From this risk model data, the following is obtained:

- the DEXi model to be evaluated (field "dexi\_model"),
  - the list of indicators included in the DEXi model to determine which one will need to be populated in the instantiation of the model,
  - the list of risks included in the DEXi model to be able of identifying the name of the nodes in the dexi model to be instantiated.
- For each indicator included in the DEXi model, the corresponding indicator value stored for the target or organization (in the case of business indicators) is retrieved. The objective is to build pairs with the format:

***indicator\_name=indicator\_value***

The indicator\_name can be composed by different parts. First, it has to include the id of the indicator as it is defined in D3.1. This indicator id is recovered from the data field "question" included for each indicator stored in the DWH indicators catalogue (`/rae/indicators`). The data field "question" follows this format: `<indicator_id>:<short_description>`.

For example, `<indicator_name>=IN-21` for the following indicator:

```
{"id":8,"question":"IN-21:ls      there      unusual      activity      in      the
infrastructure?","data_type":"boolean","motivation":"It has been detected unusual activity
against      IP's      in      the      client's
infrastructure.","indicator_type":3,"means":"alarm","rule":"PLUGIN_ID=70000      AND
PLUGIN_SID IN (50000:51999)"}

```

Besides, depending on the risk model, it can be necessary to instantiate a same indicator by each risk in the model. In order to identify these indicators for its evaluation, it is added the *short\_description* of the risk (e.g. "WRP4-R1") as a suffix adding the character "\_" after the indicator id.

For example: `<indicator_name>=IN-11_WRP4-R1`.

In order to know if it is required to instantiate by risks, the DEXi Model Instantiator does a first evaluation of the risk model without attributes and analyzes the basic attributes that comes out. This means that internally it is executed the equivalent to the following command line:

```
# java -jar JDEXi3Eval.jar WRP-4.dxi ""

```

Finally, depending on the risk model and to avoid having linked attributes, it can be necessary to add another suffix with the name of the attribute. This situation is also detected in the analysis of the basic attributes for the DEXi model.



For example: <indicator\_name>=IN-20\_WRP3-R4\_S1.

- Additionally, to the indicators defined for a risk model, a dummy value associated to a dummy node has to be added ("DummyNode=DummyValue1"). The dummy node allows us to simulate constant values in the model for nodes for which we do not have indicators. A node can be given a constant value by using the dummy node as its child and defining the utility function such that its output is the same (i.e. it equals the desired constant value) irrespective of the value of the child node. In this way, the customer who instantiates a model for an organization can fix the value of the constant simply by redefining the utility function before uploading the model back to the platform. A default utility function is defined in the original model, which will provide a default value in case the user directly selects the risk model without making changes for its organization.

The final string prepared by the DEXi Model Instantiator to be used in the DEXi evaluation of the model would be something like the following one:

```
IN-C7=High;IN-C8=Very low;IN-C9=Very low;IN-11_WRP4-R1=No;IN-11_WRP4-R2=No;IN-11_WRP4-R3=No;DummyNode=DummyValue1
```

### 3.2.5 R Model Rules Executor

This module is in charge of the execution of the quantitative risk assessment through the use of R scripts.

These scripts require a set of arguments for the evaluation that are provided as output of the R Model Instantiator. See section 3.2.6 for specifications of how the module works.

The output of that instantiation contains two variables. The first one represents the name of the file of the R scripts that will be executed by the module in order to do the quantitative assessment, and a string with the indicators required for that model and the values stored in the Data Warehouse, formatted with R syntax to be read as arguments. For example:

```
r_instantiator_output = [{  
  'r_model': 'WRP3',  
  'indicators_instance': ' cl_S1_to_S3 <- interval(0.5, 0.7);cl_S2_to_S3 <- interval(0.3, 0.5);cl_S3_to_U1 <-  
interval(0.7, 0.9);cl_S3_to_U2 <- interval(0.7, 0.9);IN_21 <- FALSE;IN_20 <- FALSE;IN_C1 <- FALSE;IN_C2  
<- FALSE;IN_C3 <- FALSE;IN_C4 <- FALSE;IN_C5 <- FALSE;IN_C6 <- FALSE;'  
}]
```

The argument “-e” receives R formatted expressions, so the model executor calls the Rscript with the content of the indicators\_instance received by the instantiator, and the corresponding r\_model to execute it in this way:

```
Rscript -e "library(sets);" + indicators_instance + "source(\"" + r_model + ".R\")"
```

Following the example, it will be evaluated as this:

```
Rscript -e "library(sets); cl_S1_to_S3 <- interval(0.5, 0.7);cl_S2_to_S3 <- interval(0.3,  
0.5);cl_S3_to_U1 <- interval(0.7, 0.9);cl_S3_to_U2 <- interval(0.7, 0.9);IN_21 <- FALSE;IN_20 <-  
FALSE;IN_C1 <- FALSE;IN_C2 <- FALSE;IN_C3 <- FALSE;IN_C4 <- FALSE;IN_C5 <- FALSE;IN_C6 <-  
FALSE; source(\"WRP3.R\")"
```

It's important to note that the R library 'sets' needs to be installed and prepend to the expression in order for the intervals to be correctly expanded and assigned before executing the script with the 'source' command.

The execution of the R script will generate a JSON style output like the following:

```
{ "R1" :[5600, 194400], "R2" :[5600, 194400], "R3" :[0, 12960], "R4" :[5600, 259200], "I_S1" :[100, 120], "I_S2" :[100, 120], "I_S3" :[80, 144], "I_S1_to_S3" :[50, 84], "I_S1_to_S3" :[30, 60], "I_S3_to_U1" :[56, 129.6], "I_S3_to_U2" :[56, 129.6], "I_U1" :[56, 129.6], "I_U2" :[56, 129.6] }
```

From that output, the R\_Rules\_Executor will parse the JSON and extract each of the quantitative assessments. These are represented as an array with the minimum and the maximum value in euros for each specific risk.

The result of the evaluation for the quantitative assessment is then stored in the DWH using the endpoint (/rae/risk\_per\_target). The qualitative assessment will be stored by the DEXi Model Rules Executor explained in section 3.2.3.

In the following example, you can see the quantitative\_assessment corresponding to the risk 1 (represented as R1 in output of the R script) that was stored for the target 48.

```
{"id":5,"report":60,"risk_model":3,"risk":1,"target":48,"qualitative_assessment":"very high",  
"quantitative_assessment":[5600, 194400]}
```

After all the corresponding quantitative assessments are stored in the DWH, the Aggregator module is invoked (see section 3.2.7).

### 3.2.6 R Model Instantiator

This module is the counterpart of the DEXi Model Instantiator, and provides a similar functionality, but adapted for the R scripts.

The R Model Instantiator is invoked by the R Model Rules Executor to gather the indicator values and the risk model selected for a specific target from the DWH.

The steps followed when this module is invoked are:

1. Read the configuration file containing the value for the constants to be used as arguments for the R scripts. These constants are expressed as intervals that represent the conditional probability of a secondary attack occurring, giving the case that a first attack has happened.
2. Retrieve all the indicator values stored in the DWH for the specific target evaluated. This process is similar to the one in the DEXi Model Instantiator, quering the DWH using the endpoint /rae/indicator\_values and filtering by the target, but the output is formatted to be read by R.
3. The instantiator module parses the list of indicator objects (name and value) into a string that can be used as an argument before executing the R scripts to do the assignment of variables used for the evaluation of the risks.

R cannot accept dashes in the names of variables, so the dashes in the indicators retrieved from the DWH are replaced by underscores, and also the “yes” and “no” values are changed into “TRUE” and “FALSE”.

The assignation in R is made by an arrow “<-”, so the name/value pair will have the format:

```
Indicator_name <- value;
```

The final string prepared by the R Model Instantiator will prepend the configurable parameters:

```
cl_S1_to_S3 <- interval(0.5, 0.7);cl_S2_to_S3 <- interval(0.3, 0.5);cl_S3_to_U1 <- interval(0.7,  
0.9);cl_S3_to_U2 <- interval(0.7, 0.9);IN_21 <- FALSE;IN_20 <- FALSE;IN_C1 <- FALSE;IN_C2 <-  
FALSE;IN_C3 <- FALSE;IN_C4 <- FALSE;IN_C5 <- FALSE;IN_C6 <- FALSE
```

### 3.2.7 Aggregator

In the following we explain the module for aggregation when using qualitative assessment. The Aggregator module is invoked by the Triggering Detector module after the DEXi Model Executor has been invoked for the risk models selected by the organization and for the different targets. The goal of this component is to do the aggregation of the different risk assessment values stored in the DWH by targets (*/rae/risk\_per\_target*).

The Aggregator module makes use of a timeout configured in the file *config.json* (parameter *timeoutAggregator*) to wait those seconds once it has been invoked for a specific risk report id before starting the aggregation. In that way, it can be done considering the different risk models selected that will be processed by different DEXi Rules Executor threads.

Currently, there are two types of aggregation algorithms implemented: “average” and “maximum”. A parameter “*aggregation\_algorithm*” has been included in the *config.json* to establish which one to apply. By default, it is configured the maximum algorithm. In the presentation of results, the number of risks included in the aggregation will be shown together with the aggregated risk level. The reason is that there is clearly a big difference between having, for example, 10 risks with average value medium and 30 risks with average value medium, even if the average value is the same.

There are different aggregation levels:

- **Overall Aggregation for the organization:**

In this case, it is retrieved all the targets belonging to the organization received by the Aggregator module in the request. For each target, it is recovered from the DWH (*/rae/risk\_per\_target*) the risk assessments stored by risk and target by the DEXi Model Rules Executor modules. For each risk per target, it is recovered the id of the risk and the qualitative\_assessment.

There are 5 potential qualitative\_assessment values: very low, low, medium, high, very high. The Aggregator will assign to each of them a value in the range 1 to 5. Depending on the aggregation algorithm selected, the overall aggregated value will be its average value or its maximum value.

The result will be stored in DWH in */rae/risk\_reports* using the risk report id provided the Triggering Detector when the execution of the risk assessment algorithms was launched. For example:

```
{ "id":82,"qualitative_assessment":"high","quantitative_assessment":"","organization":1,"timestamp":"2016-06-21T12:52:13.342694Z","mitigation_measures":[] }
```

- **Aggregation by sections:**

In this case, the Aggregator calculates the aggregated qualitative\_assessment taken into account the different sections that can appear in the risk reports. These sections are defined in the DWH in */rae/sections* and each risk stored in the DWH risk catalogue (*/rae/risks*) has the list of the sections where that risk should be considered.

For each section available in the risk report, it will be aggregated the qualitative\_assessment for the risk belonging to that section.

The result will be stored in DWH in */rae/risk\_reports\_section* using the risk report id provided the Triggering Detector when the execution of the risk assessment algorithms was launched. For example:

```
{ "id":49,"report":104,"section":3,"qualitative_assessment":"medium","quantitative_assessment":"low" }
```

- **Aggregation by risks:**

In this case, the Aggregator calculates the aggregated qualitative\_assessment taken into account the different risks registered in the DWH risk catalogue (*/rae/risks*).

The Aggregator recovers for each risk in the catalogue the entries per target available in */rae/risk\_per\_target* for the organization and report considered, and the aggregation algorithm is performed with those qualitative assessment values.

The result will be stored in DWH in */rae/risk\_reports\_risk* using the risk report id provided the Triggering Detector when the execution of the risk assessment algorithms was launched. For example:

```
{"id":34,"report":67,"section":3,"risk":4,"qualitative_assessment":"medium","quantitative_assessment":"low"}
```

For the quantitative assessment, the aggregation method is straightforward since it is only about summing the amount of money associated per each risk and target, which gives the amount of money per section of the report, and in turn the results per section are summed to produce the overall result.

### 3.3 Deployment

#### 3.3.1 Pre-requisites

- Oracle JDK 1.7 or later: When installed, make sure you have *JAVA\_HOME* pointing to the Java folder installation and included in the PATH. They are also included in the script */etc/init.d/wiser-rae*.
- Python 2.7 or later. The following python modules are required: pika, ssl
- Regarding the execution of the R components, it is necessary to install some dependencies previously:

```
sudo sh -c 'echo "deb http://cran.rstudio.com/bin/linux/ubuntu trusty/"  
>> /etc/apt/sources.list'  
gpg --keyserver keyserver.ubuntu.com --recv-key E084DAB9  
gpg -a --export E084DAB9 | sudo apt-key add -  
sudo apt-get update  
sudo apt-get install r-base
```

Once R is installed, open the interface executing the command 'R' and install the following package:

```
install.packages("sets")
```

- From the server where the Risk Assessment Engine is installed, it is necessary to have access to the RabbitMQ Server and the DWH endpoint.
- The Risk Assessment Engine has been tested on linux Ubuntu.

#### 3.3.2 Installation procedure

The Risk Assessment Engine is provided as a python application that, once unzipped, can be installed executing with root user<sup>7</sup>:

```
# python setup.py install --prefix=/usr
```

<sup>7</sup> This will be done by the entity providing the WISER Service, This installation process is not to be done by the client.

All the files required will be installed in the folder `{prefix}/share/wiser-rae`.

The configuration can be done editing the file `{prefix}/share/wiser-rae/config.json`. See the description of the parameters in section 4.5.1.

The Risk Assessment Engine is provided as a service called “**wiser-rae**” (`/etc/init.d/wiser-rae`) and it can be added to the services in the machine so it can be automatically started executing:

```
# update-rc.d wiser-rae start 35 3 4 5.
```

### 3.3.3 How to verify the installation

The Risk Assessment Engine is deployed as a daemon called **wiser-rae**. In order to verify the installation, you can check the status of the service with the following command:

```
root@wiservm4:~# service wiser-rae status
Wiser Risk Assessment Engine is running (pid is 11338).
root@wiservm4:~#
```

You can also verify the installation checking the following script python is running as a daemon:

```
wiseruser@wiservm4:~$ ps -ef|grep wiser-rae
root  23511 21214  0 14:06 pts/3    00:00:00 /usr/bin/python -OOt /usr/bin/wiser-rae
```

If there is no response, this means that the Risk Assessment Engine is not running and you must start the service manually by typing:

```
root@wiservm4:~# service wiser-rae start
```

The logs generated by the Risk Assessment Engine are stored in the file `/var/log/wiser/wiser-rae.log`

```
root@wiservm4:~# tail -f /var/log/wiser/wiser-rae.log
RAE: Recovered dwh access token:g4YmYA05BWTObSWJMygxd5hyDbzIUy
IVG: Starting thread to refresh DWH Access Token
IVG: Starting thread to reset indicator values periodically
IVG: Checking timestamp of indicator values
Starting processing notifications received from DWH
Starting processing VulnReports
IVG: Starting processing incoming Events
IVG: Starting processing Alarms
Triggering Detector: Starting processing IndicatorChanges received from DWH
```

## 3.4 Operation

---

### 3.4.1 User Manual

This user manual will support the administrator of the WISER framework<sup>8</sup> to configure and run the Risk Assessment Engine.

The Risk Assessment Engine can be set up using the file **config.json**. The following parameters can be configured:

- *dwh\_url*: dataware House URL.
- *client\_id* and *client\_secret*: credentials issued specifically to the Risk Assessment Engine component accessing the DWH API.
- *username* and *password*: user credentials used to get the authentication token.
- *rabbitmq\_server* and *rabbitmq\_port*: information to connect to the RabbitMQ Server.
- *ca\_certs*, *ivg\_certfile*, *ivg\_keyfile*, *td\_certfile* and *td\_keyfile*: certificate files required to the connection to the RabbitMQ Server using SSL from the Indicator Value Generator (ivg) and Triggering Detector (td) modules.
- *timeoutIndicatorValues*: this is the timeout used by the Indicator Value Generator module to check when a network indicator value (those ones related to events and alarms received from the monitoring infrastructure) stored in the DWH (*/rae/indicator\_value*) is obsolete and it needs to be set to its default value (*False*). Each time an indicator value is created or updated in the DWH, it is registered its timestamp. In this way, it is prevented that once an event or alarm is detected and the Indicator Value Generator has set it to *True*, it stays in this status indefinitely for the risk assessment. This timeout is indicated in seconds. By default, 3600.
- *resetProcess*: this value indicates how often the Indicator Value Generator module will check the timestamps of the current indicator values stored in the DWH in order to reset them if required. This reset time is indicated in seconds. By default, 3600.
- *timeoutTriggeringDetector*: this is the timeout used by the Triggering Detector module once it has been received a notification due to a change in an indicator value stored in the DWH to launch the qualitative and quantitative evaluation of the risk models. In this way, instead of executing the DEXi and R algorithms for each time a single indicator value has been updated, the evaluation will be done taken into account all the indicator changes received in the timeout configured. For example, we can think about a change in the questionnaire affecting several business indicators related to the company profile. Thanks to this timeout, the evaluation of the risk assessment algorithms will be done only once considering all the questions modified instead of several times, each of them adding one of the indicators. This timeout is indicated in seconds. By default, 10.
- *aggregation\_algorithm*: this is the algorithm used by the Aggregator module to perform the aggregation of the qualitative assessments of the different risks involved in the risk models selected. Currently there are two available algorithms: average and maximum. By default, it is configured to "maximum".
- *timeoutAggregator*: this is the timeout used by the Aggregator module once it has been invoked to start the aggregation algorithm with the risk assessment values stored in the DWH and update the results in the risk report. In this way, if there are several risk models selected for a same organization, the aggregation will be done only once including the qualitative and quantitative evaluations done for all the models. This timeout is indicated in seconds. By default, 30.

The Risk Assessment Engine is deployed as a Linux daemon called "**wiser-rae**" and consequently it can be managed using the command *service*.

- Start the Risk Assessment Engine:

---

<sup>8</sup> This will be done by the entity providing the WISER Service. This setup process is not to be done by the client.

```
# service wiser-rae start
```

- Check the status of the Risk Assessment Engine:

```
# service wiser-rae status
```

- Stop the Risk Assessment Engine:

```
# service wiser-rae stop
```

The Risk Assessment Engine is running in background and the logs generated are stored in the file **/var/log/wiser/wiser-rae.log**.

The results of the risk assessments performed by the Risk Assessment Engine are stored directly in the Data Warehouse<sup>9</sup> and can be visualized by the user through the Cyberwiser Essential Dashboard.

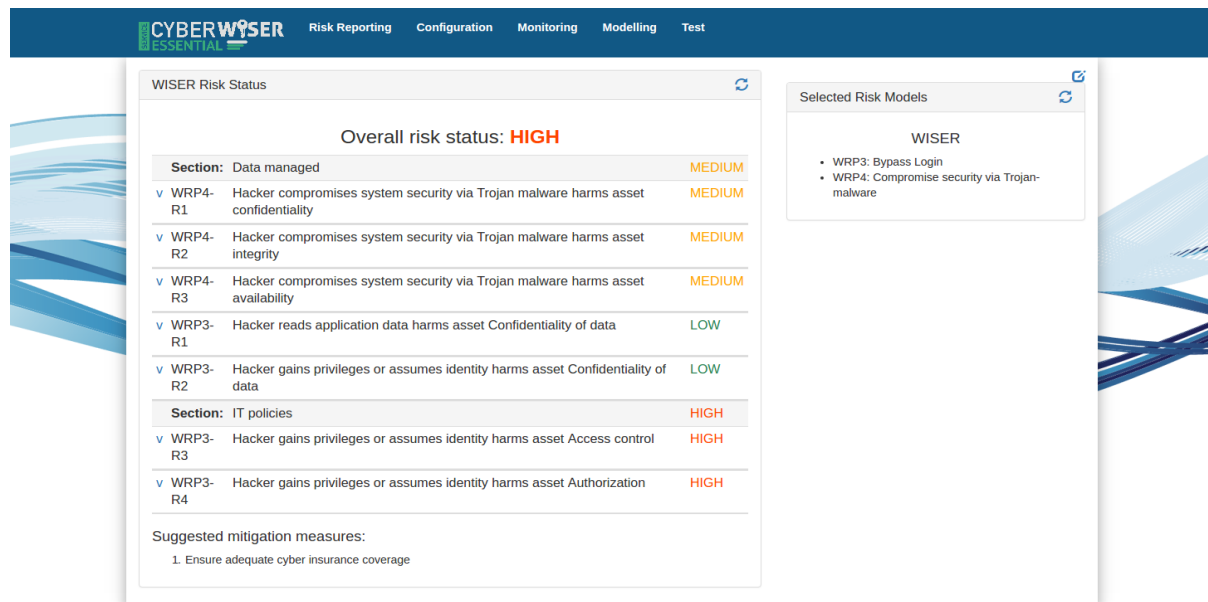


Figure 29. Example of risk report visualization in CyberWISER Dashboard.

### 3.4.2 Example of usage

1. Check the status of the Risk Assessment Engine running:

```
root@wiservm4:~# service wiser-rae status
Wiser Risk Assessment Engine is running (pid is 11338).
root@wiservm4:~#
```

2. Log generated when an alarm is received in the Risk Assessment Engine:

<sup>9</sup> The Data Warehouse and the specific API for the Risk Assessment Engine and the rest of components are addressed in Section 4 and in Appendix I

```
IVG: Received Alarm: {u'DST_IP_HOSTNAME': u'00000000', u'PRIORITY': 4, u'SRC_PORT': 17784, u'RISK': 3, u'RELATED_EVENTS': u'[69d511e6a3e600163e424823aebebe84,69d511e6a3e600163e424823d0b6fetc,69d511e6a3e600163e424823f0e6d076,69d611e6a3e600163e4248231117c8be,69d611e6a3e600163e424823327860c2,69d611e6a3e600163e424823520f4914,69d611e6a3e600163e42482371a6975a,69d611e6a3e600163e42482392269ee4,69d611e6a3e600163e424823b3875830,69d611e6a3e600163e424823d2868e0e,69d611e6a3e600163e424823f1ec2c86,69d711e6a3e600163e42482312a09822,69d711e6a3e600163e4248233197f900,69d711e6a3e600163e4248235260d440,69d711e6a3e600163e424823730cce88,69d711e6a3e600163e42482393d54b4a]', u'EVENT_ID': u'92f85e1a3dbe4b38b44ba42eff7f2da0', u'PROTOCOL': 6, u'PLUGIN_NAME': u'directive_sls', u'SRC_IP': u'221.229.172.116', u'BACKLOG_ID': u'57deda0d7c3c4571a3c8538aebac0cd', u'RELIABILITY': 10, u'DST_PORT': 22, u'ORGANIZATION': u'WISER', u'DATE': u'2016-08-24 08:48:49', u'DST_IP': u'212.34.151.211', u'PLUGIN_SID': u'50113', u'SRC_IP_HOSTNAME': u'00000000', u'PLUGIN_ID': u'70000', u'SID_NAME': u'directive_event: Bruteforce attack, SSH service authentication attack against DST_IP'}
IVG: Received Alarm: {u'DST_IP_HOSTNAME': u'00000000', u'PRIORITY': 4, u'SRC_PORT': 18711, u'RISK': 1, u'RELATED_EVENTS': u'[69d711e6a3e600163e4248235260d440,69d711e6a3e600163e424823730cce88,69d711e6a3e600163e42482393d54b4a]', u'EVENT_ID': u'52c0e19614484aaeacbd8386b2d577e9', u'PROTOCOL': 6, u'PLUGIN_NAME': u'directive_sls', u'SRC_IP': u'221.229.172.116', u'BACKLOG_ID': u'6c7aa9f9830d4d138d2a7c76bbf2f8ba', u'RELIABILITY': 4, u'DST_PORT': 22, u'ORGANIZATION': u'WISER', u'DATE': u'2016-08-24 08:48:49', u'DST_IP': u'212.34.151.211', u'PLUGIN_SID': u'50098', u'SRC_IP_HOSTNAME': u'00000000', u'PLUGIN_ID': u'70000', u'SID_NAME': u'directive_event: Bruteforce attack, SSH authentication attack against DST_IP'}
```

## 4 Data Warehouse

### 4.1 Overview

Data Warehouse is the central data storage component of the WISER Framework. It stores and provides information crucial to the operation of the Framework, including:

- client users and organizations,
- users' configuration parameters,
- risk models,
- catalogues of risks, mitigation measures and indicators,
- risk reports (results of finished risk assessment procedures),
- data about active deployed sensors,
- events reported by sensors,
- alarms reported by the Monitoring Engine,
- reports of vulnerabilities found by the vulnerability scanners.

The Data Warehouse consists of two database backend services (relational and document-based), their respective definitions of data models, and a common API, providing a single data retrieval point to other components of the WISER Framework.

The following subsections present the data models, the technologies used for storage and communication, and briefly describe the installation and operation of the Data Warehouse, highlighting the differences between the design presented in the Deliverable D5.1 and the current state of the implementation.

### 4.2 Data modelling

This section presents the data models in both the relational database and the document-based store, which form the storage component of the Data Warehouse.

#### 4.2.1 Relational database



---

The relational-database component of the Data Warehouse is intended for storage of data with a rigid schema and a low rate of input. The database tables are directly mapped to REST API endpoints, enabling a simple and generic interface to other components of the WISER Framework. Data stored in the relational Data Warehouse part includes:

- information about users and client organizations (input manually by administrators),
- user settings, business configuration variables, risk models' configuration (input by end users),
- indicator values and risk assessment results (computed and inserted by the Risk Assessment Engine),
- status of deployed sensors (input by sensors themselves by means of heartbeat messages),
- catalogues of static data, e.g. risks, mitigation measures, risk patterns.

The schema of the Data Warehouse relational database is shown in Figure 30. The data tables shown in the figure correspond directly to the API endpoints accessible via HTTP REST protocol. The fields correspond to the parameters which are be passed to and retrieved from the API.

A similar schema was already presented in the Deliverable D5.1. Since the delivery of D5.1, we have changed some parts of the schema to achieve the best fit for the actual deployment of the developing WISER Framework. The testing module has been removed from the relational schema with the vulnerability scan results now being saved to the document-based storage because of their similarity to monitoring events in terms of their delivery to the Data Warehouse, as well as large size and high expected rate of input. A configuration module has been added to provide storage for business configuration variables, monitoring and alerting policies, information about monitored networks and deployed WISER Agent instances and risk models selected by client organizations. A monitoring module has been added to store information about the status of deployed sensors. Minor changes have been introduced to other parts of the database schema, mostly regarding relations between data types.

The complete description of all API endpoints is included in Appendix I.

#### **4.2.2 Document-based store**

The document-based storage component of the Data Warehouse provides storage for less-structured documents with no relations to other documents and with a high rate of input. Data types stored in the document-based store are:

- monitoring events,
- monitoring alerts,
- vulnerability scan reports.

The documents can be retrieved by means of the same HTTP API as with the relational-backed store. A list of documents can be retrieved and filtered by any field present in the document, and a single document can be retrieved by its ID. Documents cannot be inserted via the HTTP API, but only by sending it to the Monitoring Communication Bus (RabbitMQ) via AMQP. Stored documents cannot be updated or deleted.

### **4.3 Implementation**

The Data Warehouse uses a variety of technologies to support storing different types of data while serving requests at a high frequency and keeping response times low as well as ensuring authenticity and privacy of data. As described in the previous sections, the Data Warehouse uses two backend databases; a MySQL database is used for the relational store and an Elasticsearch engine for the document-based storage. The API used for accessing Data Warehouse from other WISER components is based on HTTP REST technology and written using Django REST Framework in Python. Another interface used for communication with the Data Warehouse is the Monitoring Communication Bus, which is based on RabbitMQ, an implementation of AMQP.

### 4.3.1 Communication

The HTTP REST API is a layer above the storage components of the Data Warehouse. It enables a common interface to all the Data Warehouse's data models using generic HTTP REST calls: GET, POST, PUT and DELETE with JSON objects as payload format. The REST API enables writing and reading from the relational-based data resources.

Data types with a higher rate of input to the Data Warehouse are provided through the RabbitMQ service. A simple program runs alongside the Data Warehouse and listens to messages coming from the RabbitMQ to the Data Warehouse. Depending on the data type, it writes the document either directly to Elasticsearch (in case of events, alerts and vulnerability scan reports) or via the REST API to the relational store (in case of sensor heartbeats).

### 4.3.2 Security

The Data Warehouse REST API ensures security by using HTTPS for all requests and the OAuth2<sup>10</sup> authentication mechanism. Every request to the API must contain an access token, which can be acquired by a POST request to a special API endpoint, which also returns a refresh token. The access tokens expire after a certain period of time, after which a new access token can be obtained by means of the refresh token. There are three user authorization levels corresponding to three pre-defined user groups<sup>11</sup>:

1. "Wiser Coordinator": A representative of the WISER consortium, who has the highest level of access: he/she can read and write most objects and create new users of both lower-permission roles.
2. "Client Organization Administrator": A manager of the client organization, who can read and write objects that are associated with their organization.
3. "Client Organization Staff": A user belonging to the client organization with the most basic permission levels: he/she can only read objects that are associated with their organization.

The Monitoring Communication Bus (RabbitMQ) accepts connections encrypted with TLS v1.2. Authentication of both clients and server is assured using public-key certificates. The server accepts only clients that present a valid certificate issued by the WISER certificate authority.

### 4.3.3 Notifications about data changes

The Data Warehouse sends messages via RabbitMQ to other WISER components when any object instance of an observed data type is changed or added.

The observed data types are: Vulnerability scan report, Target, Indicator value, Company profile, Selected risk model, Risk model.

The RabbitMQ messages are in JSON format and include:

- "object\_ID": the ID of the object manipulated,
- "action": the action that was performed on the object (CREATED, UPDATED or DELETED),
- "object\_class": the type of the object (name of the data table).

An example of a message about a changed object:

```
{"object_ID": 13, "action": "CREATED", "object_class": "IndicatorValue"}
```

<sup>10</sup> <https://oauth.net/2/>

<sup>11</sup> Some details about user grouping can be found in Deliverables D2.2 and D2.3

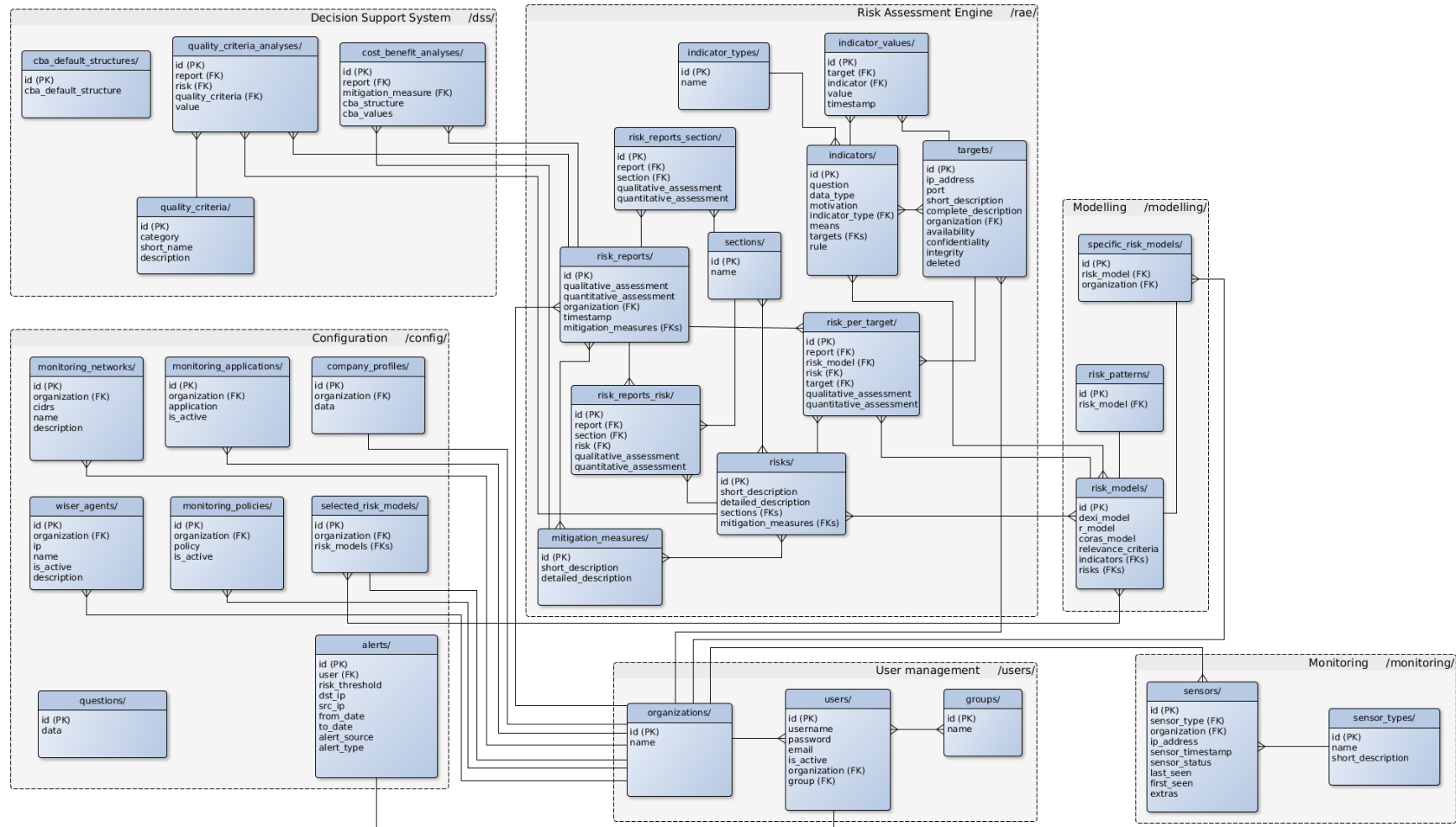


Figure 30. Relational database model in WISER Data Warehouse

## 4.4 Deployment

The deployment of the WISER Data Warehouse consists of the Python-based DWH API, a MySQL server and an ElasticSearch server. The procedure described herein considers the deployment of the API and the MySQL database on one machine, and the ElasticSearch engine on the other. However, it is possible to install all three components on the same machine or each of them separately. In that case, the hardware requirements for the machine(s) should be adjusted accordingly.

### 4.4.1 Pre-requisites

Recommended hardware specifications for

- machine 1 (DWH API and MySQL server): 2 CPU cores, 2 GB of RAM, 10GB disk space,
- machine 2 (ElasticSearch engine): 2 CPU cores, 4GB of RAM, 100GB disk space.

The described software has been tested on Ubuntu Linux 14.04.

Apart from the OS, all required software is either included in the installation packages or automatically downloaded from the public repositories.

### 4.4.2 Installation Procedure

#### 4.4.2.1 DWH API and MySQL

Install required dependencies<sup>12</sup>:

```
apt-get update
apt-get install mysql-server python3.4-dev libmysqlclient-dev gunicorn nginx supervisor
python3-pip
```

Copy data-warehouse folder to /opt/.

Install required Python libraries:

```
pip3 install -r requirements.txt
```

Create MySQL database objects:

```
mysql -uroot -p < dwh/dwh-create-db.sql
python3 manage.py migrate
python3 manage.py collectstatic
```

Create files and folders required for logging:

```
mkdir /var/log/dwh
chmod a+w /var/log/dwh/
mkdir /var/lib/dwh
chmod a+w /var/lib/dwh/
touch /var/log/dwh/dwh.log
chmod a+r /var/log/dwh/dwh.log
```

Copy keys and certificates for connection with RabbitMQ:

```
cp dwh/ca.crt dwh/wiser-dwh-rabbit.xlab.si.key.nopass /dwh/wiser-dwh-rabbit.xlab.si.crt
/var/lib/dwh/certificates/
```

Make sure to set the appropriate read permissions for the certificate files. Only the user running Gunicorn should be able to read the private key.

Review and set the settings in `dwh_api/settings.py`. Addresses of RabbitMQ and ElasticSearch servers have to be set.

<sup>12</sup> This will be done by the entity providing the WISER Service, this installation process is not to be done by the client.

Insert initial data into the Data Warehouse relational DB. Adapt the administrator's credentials by your choice:

```
echo "from django.contrib.auth.models import User;
User.objects.create_superuser('admin', 'admin@example.com', 'adminpass')" | python3
manage.py shell
python3 manage.py loaddata core/fixtures/initial_data.json
python3 manage.py loaddata --app configuration questions
python3 manage.py loaddata --app configuration comp_profiles
python3 manage.py loaddata --app modelling modelling
```

Configure Supervisor to start Gunicorn server with DWH API automatically. Review the settings in gunicorn.conf before issuing:

```
cp gunicorn.conf /etc/supervisor/conf.d/
service supervisor restart
```

Configure Nginx server:

```
cp nginx-dwh /etc/nginx/sites-available/dwh
ln -s /etc/nginx/sites-available/dwh /etc/nginx/sites-enabled/dwh
rm /etc/nginx/sites-enabled/default
service nginx restart
```

#### 4.4.2.2 ElasticSearch

Install required dependencies:

```
add-apt-repository -y ppa:webupd8team/java
apt-get -y install openjdk-7-jre
apt-get install oracle-java8-installer
```

Install ElasticSearch:

```
wget https://download.elastic.co/elasticsearch/elasticsearch/elasticsearch-2.1.0.deb
dpkg -i elasticsearch-2.1.0.deb
update-rc.d elasticsearch defaults
```

Configure ElasticSearch settings:

```
echo "cluster.name: elastic-cluster" > /etc/elasticsearch/elasticsearch.yml
echo "node.name: node-1" > /etc/elasticsearch/elasticsearch.yml
echo "path.data: /var/lib/elasticsearch" > /etc/elasticsearch/elasticsearch.yml
echo "path.logs: /var/log/elasticsearch" > /etc/elasticsearch/elasticsearch.yml
echo "bootstrap.mlockall: true" > /etc/elasticsearch/elasticsearch.yml
echo "network.host: 0.0.0.0" > /etc/elasticsearch/elasticsearch.yml
echo "http.port: 9200" > /etc/elasticsearch/elasticsearch.yml
echo "vm.swappiness = 1" > /etc/sysctl.conf
echo "ES_HEAP_SIZE=512m" > /etc/default/elasticsearch
echo "bootstrap.mlockall: true" > /etc/elasticsearch/elasticsearch.yml
service elasticsearch start
```

Install ElasticSearch-HQ plugin for monitoring and management of ElasticSearch engine (optional):

```
./plugin install royrusso/elasticsearch-HQ
```

Install and configure Nginx for proxying requests to ElasticSearch and enable basic HTTP authentication (set credentials by your choice):

```
apt-get install nginx apache2-utils
rm /etc/nginx/sites-enabled/default
htpasswd -b -c /etc/nginx/conf.d/admins.htpasswd wisser wisserpassword
htpasswd -b -c /etc/nginx/conf.d/users.htpasswd search searchpassword
cp nginx/default /etc/nginx/sites-enabled/default
```

```
service nginx restart
```

Create ElasticSearch indices:

```
/elastic/createindices.sh
```

Install the Rabbit-DWH-Interface program to connect ElasticSearch database with RabbitMQ:

```
mkdir /opt/xlab/  
cp -ar /wiser-rabbit-dwh-interface /opt/xlab/wiser-rabbit-dwh-interface  
apt-get install python-pip supervisor  
pip install -r /opt/xlab/wiser-rabbit-dwh-interface/requirements.txt  
cp /opt/xlab/wiser-rabbit-dwh-interface/config/supervisor/wiser-rabbit-dwh-  
interface.conf /etc/supervisor/conf.d/  
service supervisor restart
```

#### 4.4.3 How to verify that the installation is correct

The installation of the DWH API and its connection to the MySQL server can be verified by connecting to its REST API (see request examples in section 4.5). The “Django-admin” administration web site can be reached at [https://<address\\_of\\_the\\_server>/admin](https://<address_of_the_server>/admin). The associated log files are:

```
/var/log/dwh/dwh.log, /var/log/supervisor/gunicorn-stdout.log,  
/var/log/supervisor/gunicorn-stderr.log, /var/log/nginx/access.log,  
/var/log/nginx/error.log, /var/log/mysql/error.log.
```

The installation of the ElasticSearch engine can be verified by connecting to its management (HQ) plugin at [http://<address\\_of\\_the\\_server>/plugin/hq/#](http://<address_of_the_server>/plugin/hq/#). The web interface should show “wiser-vulnscanreports”, “wiser-events” and “wiser-alarms” indices and the number of documents in each of them. By requesting a list of documents from the DWH REST API, its connection with ElasticSearch can be verified.

A successful connection between all components is confirmed by sending a message about e.g. a monitoring event to the RabbitMQ and trying to retrieve the event by means of the Data Warehouse API.

The availability of the Data Warehouse can also be verified by the operation of other components that rely on it, such as the Risk Assessment Engine or WISER Dashboard.

## 4.5 Operation

### 4.5.1 User Manual

This section is intended to help users communicate with the Data Warehouse’s API, either manually or through development of other tools<sup>13</sup>.

#### 4.5.1.1 Data format

Request and response data is transferred in JSON format. Requests need to have the Content-Type HTTP header set to “application/json”.

References to other objects in the database are passed by the ID of the object; IDs are positive integers. In case of a many-to-many relationship, an array of referenced objects’ IDs are passed.

Possible data formats are:

- PK = private key (ID) of this object,
- FK = foreign key = private key (ID) of the referenced object,
- text (n) = text field (with maximum length of n characters if specified),
- Boolean = True/False (mind the first capital letter),

<sup>13</sup> This will be done by the entity providing the WISER Service, this setup process is not to be done by the client.

- datetime = combined date and time information is passed in ISO 8601 format:  
YYYY-MM-DDThh:mm[:ss[.uuuuuu]][+HH:MM|-HH:MM|Z],
- date = date format is YYYY-MM-DD,
- int = integer number,
- float = floating point number.

#### 4.5.1.2 API functions

The API endpoints correspond to the data types represented in the Data Warehouse. The data types backed by the relational database are presented on Figure 30. The endpoint addresses are comprised of two parts: a module name and a table name. For example, data about risk models can be retrieved using “/modelling/risk\_models/” endpoint.

The API supports the following functions (depending on the endpoint of the data type requested):

Read functions:

- List: fetch a list of objects. Request is made by issuing a GET request to the endpoint address. The response contains a JSON array of objects and is paginated: the first request contains the first 50 objects, and the next page can be obtained by issuing a request to the URL returned in the “next” field of the response.
- Retrieve: fetch a specific object by its ID number (private key). Request is made by issuing a GET request to the endpoint address followed by the object’s ID and a final slash.

Write functions:

- Create: persist a new object in the data warehouse. Request is made by issuing a POST request to the endpoint address with the request payload containing the data of the object to be created.
- Update: change an existing object in the data warehouse. Update can be either full or partial. A full update (replace) is issued with a PUT request to the endpoint address followed by the object’s ID and a final slash. The payload has to contain all the fields that the object has. A partial update is made by a PATCH request to the same address. A partial update can only contain the fields needed to be changed.
- Delete: delete an existing object from the data warehouse. Request is made by issuing a DELETE request to the object URL (endpoint address followed by the object’s ID and a final slash). It is not possible to delete an object, referenced by foreign key in another object. Such operation will return a HTTP 409 error with a description stating which tables contain objects referencing the object the user is trying to delete.

#### 4.5.1.3 Documents

The endpoints in this section are interfaces to the document-based store in the DWH backend. Using this API, it is possible to list the documents and to retrieve a single document instance based on its ID. IDs in the document-based store are strings, which is a distinction with respect to the relational stored data, where integers are used as IDs. Inserting, modifying or deleting objects in the document-based store is not possible.

##### Filtering

Filtering is possible by any of the data fields. Three types of filtering are supported:

1. Match filtering: the data value must exactly match the query parameter.

Match filtering is done by simply including a query parameter with the name of the data field and the value we want to match. Example:

```
/documents/events/?event_id=b01611e5-a3e6-0016-3e23-4dbef07ffc30
```

2. Range filtering: queries represent the following inequality operators:

- a. greater than (gt),
- b. greater than or equal to (gte),
- c. less than (lt),
- d. less than or equal to (lte).

Range filtering is done by including a query parameter with the name of the data field followed by two underscores (\_\_) and the range query operator. Example:

```
/documents/events/?fdate__gt=2016-06-23T00:00:00Z
```

3. Not equal filtering: the data value must not match the query parameter.

Not equal filtering is done by including a query parameter with the name of the data field followed by two underscores (\_\_) and "ne" (meaning not equal). Example:

```
/documents/events/?organization__ne=WISER
```

Multiple filtering parameters can be included in a single query.

### Document types

The supported document types are:

1. Event: an event generated by a monitoring sensor, sent by the WISER Agent (/documents/events/),
2. Alarm: an alarm generated by the WISER Monitoring Engine (/documents/alarms/),
3. Vulnerability scan report: a report generated by the vulnerability scanner (/documents/vuln\_scan\_reports/).

#### 4.5.1.4 Authentication mechanism

The Data Warehouse authentication mechanism is based on OAuth2 access and refresh tokens. This section provides guidelines on how the tokens are obtained and renewed. The access token has to be included in every request to the API in the "Authorization" HTTP header of type "Bearer".

### Obtaining an access token

An access token request is made with a POST request to /o/token/. The request should contain the following parameters (passed as URL query parameters):

- client\_id, client\_secret: credentials issued specifically to each application accessing the DWH API,
- grant\_type: "password",
- username, password: credentials of the end user logging in.

The response contains the access token, its expiration period in seconds, and the refresh token needed for renewal of the access token.

The application connecting to the API is expected to forget the end-user's credentials as soon as it obtains the initial access token.

### Renewing an access token

The access token can be renewed by issuing a request similar to the initial obtaining of the token. Instead of the end-user's credentials, the refresh token is required by this function. Application credentials (client\_id, client\_secret) passed should be the same as in the request to obtain a token. The grant type parameter should be set to »refresh\_token«.

### Example of usage

1. List all users

```
curl -XGET https://dwh-wiser.xlab.si/users/users/ -H "Authorization: Bearer
```



```
d9EkHbtMgm1MZw4JzyxSM53TQp6yGa"
{
  "count": 2,
  "next": null,
  "previous": null,
  "results": [
    {
      "email": "",
      "group": 2,
      "id": 3,
      "is_active": true,
      "organization": 3,
      "username": "test2"
    },
    {
      "email": "",
      "group": 1,
      "id": 4,
      "is_active": true,
      "organization": 3,
      "username": "test3"
    }
  ]
}
```

2. Retrieve event with ID »AVWV\_ulyBp-zTewyhO\_I«

```
curl -XGET "https://dwh-wiser.xlab.si/documents/events/AVWV_ulyBp-zTewyhO_I/" -H
"Authorization: Bearer QykK60YRpiSDqtzeeKJzyegCEhynMB"
{
  "date": "1467100349",
  "device": "37.230.121.177",
  "doc_type": "event",
  "dst_ip": "37.230.121.177",
  "event_id": "3d0511e6-88f7-0026-db00-523045133d86",
  "fdate": "2016-06-28T07:52:29",
  "id": "AVWV_ulyBp-zTewyhO_I",
  "interface": "eth0",
  "log": "Jun 28 09:52:29 labo ossec: Alert Level: 8; Rule: 5104 - Interface
entered in promiscuous(sniffing) mode.; Location: (local) 127.0.0.1-
>/var/ossec/logs/alerts/alerts.log|(local) 127.0.0.1-
>/var/ossec/logs/alerts/alerts.log|(local) 127.0.0.1-
>/var/ossec/logs/alerts/alerts.log|(local) 127.0.0.1-
>/var/ossec/logs/alerts/alerts.log|(local) 127.0.0.1->/var/ossec/logs/alerts/; Jun
28 08:25:22 labo kernel: device eth0 entered promiscuous mode ",
  "organization": "Enervalis",
  "plugin_id": "80001",
  "plugin_sid": "80000",
  "src_ip": "0.0.0.0",
  "type": "detector",
  "tzone": "2.0",
  "userdata1": "5104",
  "userdata2": "Interface entered in promiscuous(sniffing) mode.",
  "userdata3": "labo",
  "userdata4": "labo kernel: device eth0 entered promiscuous mode"
}
```

3. Create an indicator

```
curl -XPOST https://dwh-wiser.xlab.si/rae/indicators/ -H "Authorization: Bearer
```

```
vRKcSQeLXUNGUvpqoJJagKlp7T00AH" -H "Content-Type: application/json" -d  
"{\"data_type\": \"integer\", \"indicator_type\": 2, \"means\": \"aaa\",  
\"motivation\": \"bbb\", \"question\": \"????\"}"
```

```
{  
  "data_type": "integer",  
  "id": 5,  
  "indicator_type": 2,  
  "means": "aaa",  
  "motivation": "bbb",  
  "question": "????"  
}
```

4. Update the question in the indicator, created in the previous example (indicator ID=5)

```
curl -XPATCH https://dwh-wiser.xlab.si/rae/indicators/5/ -H "Authorization: Bearer  
vRKcSQeLXUNGUvpqoJJagKlp7T00AH" -H "Content-Type: application/json" -d  
"{\"question\": \"ccc?\"}"
```

```
{  
  "data_type": "integer",  
  "id": 5,  
  "indicator_type": 2,  
  "means": "aaa",  
  "motivation": "bbb",  
  "question": "ccc?"  
}
```

## 5 Decision Support System

### 5.1 Functional design

The Decision Support System works in a different manner depending on the CyberWISER service in question, this is, CyberWISER-Light, CyberWISER-Essential and CyberWISER-Plus.

In CyberWISER-Light, the Decision Support System only performs the feature of showing the results of the risk evaluation performed by the algorithms within the Risk Assessment Engine. As explained in Section 3.1.3.1, there are two kinds of report produced:

- 1) The report specifying the risk derived from the company profile (addressed in Section 3.1.3.1.1, and illustrated in Figure 16 and Figure 17).
- 2) The report specifying the risk due to vulnerabilities found in the target infrastructure (addressed in Section 3.1.3.1.2 and illustrated in Figure 19).

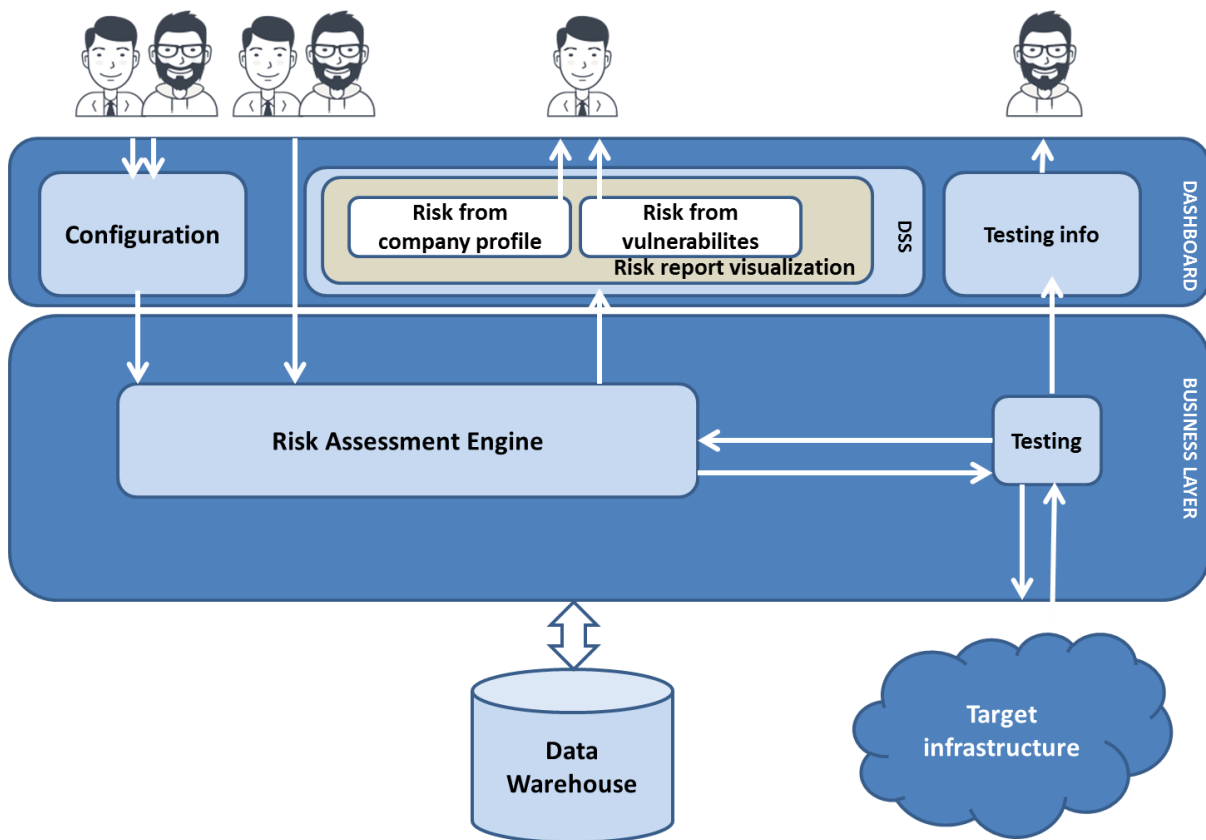


Figure 31. Decision Support System internal detail (CyberWISER-Light)

Figure 31 shows the internal composition of the Decision Support System in CyberWISER-Light. As it can be seen, the aforementioned two kinds of report are produced by specific modules, which collect the output of the Risk Assessment Engine and produce the reports accordingly. The logics performing this are part of the implementation of the CyberWISER-Light dashboard.

In CyberWISER-Essential, the Decision Support System is also part of the Dashboard. It offers two different features:

- 1) **Risk report visualization:** The risk report offered here has a more solid rationale than that of CyberWISER-Light, due to the better inputs available in quantity and quality. This feature is broken down into the following one:
  - a. As far as the current (the latest) report is concerned, it offers the chance to see the report on the dashboard and also to convert it into PDF. These reports show not only the risk assessment itself, but also the proposed mitigation measures with the goal to eventually apply them and therefore diminish the risk. This has already been addressed in Section 3.1.3.2.
  - b. The look-up module allows the user retrieving cyber risk models produced in the past and also to convert them into PDF.
  - c. Finally, it adds a graphic analytics function where the evolution of the cyber risk can be further analysed in order to find out trends that can be used to anticipate future scenarios and countermeasures.
- 2) **Societal impact analysis:** for each of the risks evaluated, the user answers a set of 19 questions in order to assess the societal impact the risk entails. In the dashboard, for the model being active, a combo box with the risks being evaluated within the model appears.

The user selects one of the risks and, for that risk, the questionnaire is filled out, moving to the following risk of the model then. The user only needs to complete the questionnaire once, as the answers are saved for further use. Of course the answers can be edited. If the risk model active changes because of user decision, then the risks appearing in the combo list will be those corresponding to the new model being active. If the previous model becomes active again, the user will see, for each of the risks of the model, the questionnaire filled out with the latest answer given to each question. All the logics concerning the evaluation of the answers to the questionnaire and issuing a result of the societal impact assessment belong to the business layer of the Decision Support System, as shown in Figure 32.

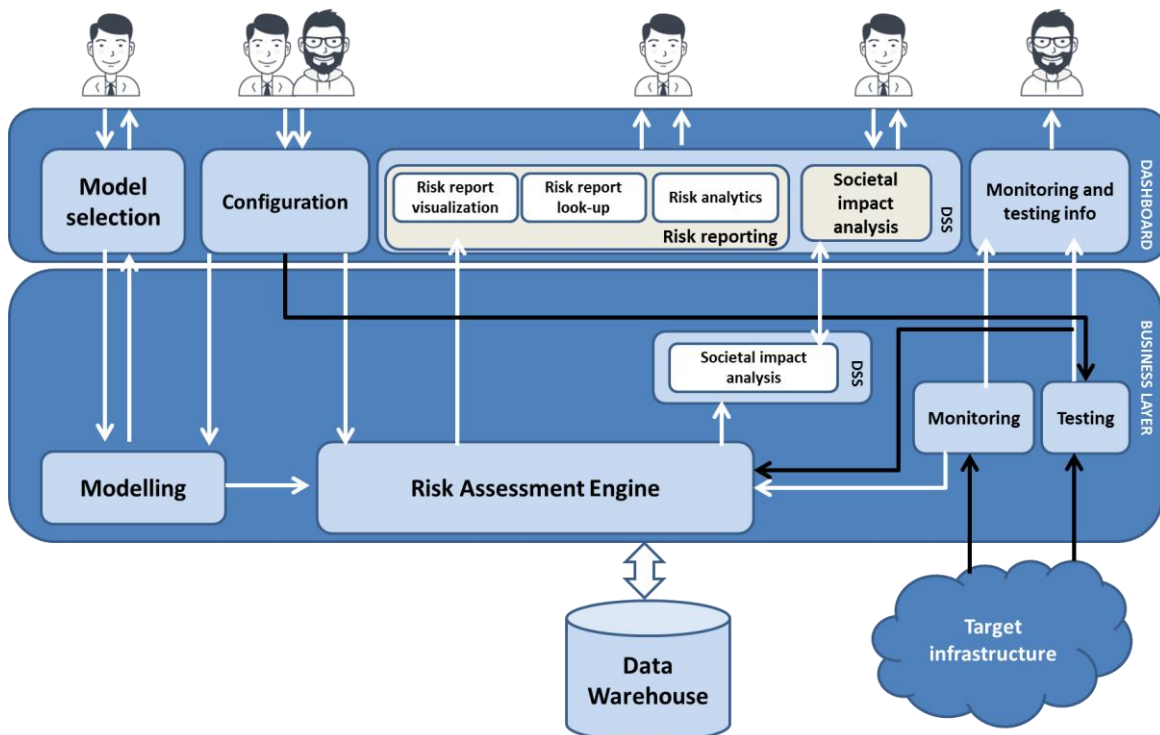


Figure 32. Decision Support System internal detail (CyberWISER-Essential)

CyberWISER-Plus adds to the features of CyberWISER-Essential a cost-benefit analysis of the mitigation measures proposed for the risks. For the risk report being active, a set of mitigation measures may have been proposed. When the user access to the cost-benefit analysis feature, a combo menu is made available and, when expanded, shows the different mitigation measures being active. The user selects the mitigation measure for which the analysis will be done and proceeds to complete the cost-benefit analysis template. CyberWISER-Plus presents a more advanced analytics functionality, as depicted in Figure 33.

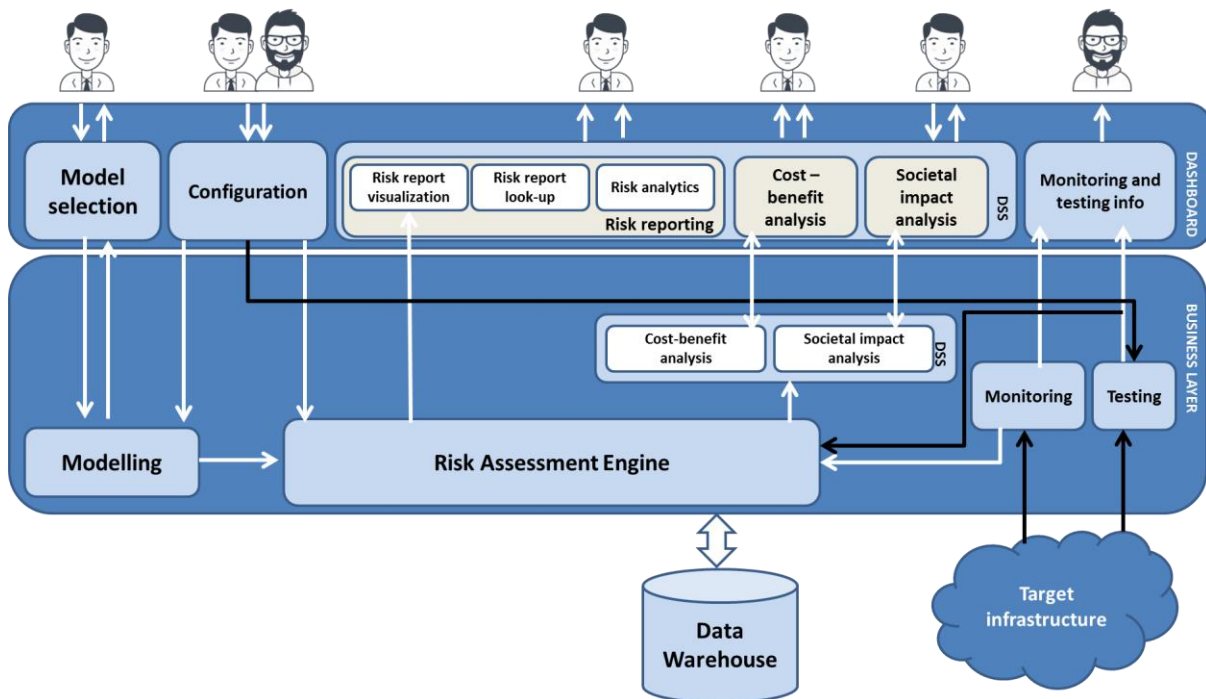


Figure 33. Decision Support System internal detail (CyberWISER-Plus)

### 5.1.1 Overview

The Decision Support System is the actor located at the end of the WISER cycle. It is a module in charge of:

- 1) Showing the user the results of the computations carried out by the Risk Assessment Engine. This includes showing the current risk report and a feature to look for older reports by following some kind of filtering criteria. The output to be shown follows the format explained in Section 3.1.3.1 for CyberWISER-Light and in Section 3.1.3.2 for CyberWISER-Essential and CyberWISER-Plus. The risk assessment includes the degree of risk (for qualitative reports), the expected losses (for quantitative reports), how the risk is faced by the different components of the target infrastructure and some supplementary information.
- 2) Suggesting mitigation measures in the reports shown to the user. These mitigation measures are envisioned to diminish the risk when applied. Once applied, the evaluation of risk is likely to show a better result because of the effect of those measures. Some supplementary information about the mitigation measures is provided in Section 5.1.2. This applies to CyberWISER-Light, CyberWISER-Essential and CyberWISER-Plus.
- 3) Offering a tool to learn about the suitability of implementing one mitigation measure from a cost-benefit perspective (cost-benefit analysis). There is a methodology underlying which is addressed in Section 5.1.3. This applies only to CyberWISER-Plus.
- 4) Showing information about the societal impact of the cyber risks assessed. Leveraging the Modelling module features, the user can learn about this societal impact. Section 5.1.4 provides some insights about the methodology, and gives some examples of quality criteria which can be used to evaluate the societal impact. This applies to CyberWISER-Essential and CyberWISER-Plus.
- 5) Offering analytical features to help the user to better understand the results produced by the Risk Assessment Engine (risk analytics). CyberWISER-Essential provides a reduced version

of this feature, whereas CyberWISER-Plus brings an extended one. This is further addressed in Section 5.1.5.

### 5.1.2 Mitigation measures

The Deliverable D5.1, Section 5.2 explains the WISER approach to the suggestion of mitigation measures. The risk assessment algorithm, executed within the Risk Assessment Engine will trigger the suggestion of certain mitigation measures. Whether or not to propose a mitigation option should be based on the following criteria:

- 1) Is the risk level for the risk in question sufficiently high to warrant proposals for mitigations?
- 2) Is the particular mitigation option expected to significantly reduce the risk?

These mitigation options, when applied, should have an effect on the cyber climate, change some indicator values, and consequently change the result of the cyber risk assessment for the better. The mitigation measures are presented in a simple way: just the name of the measure and a description of what the measure is about. The table below shows some examples of mitigation measures:

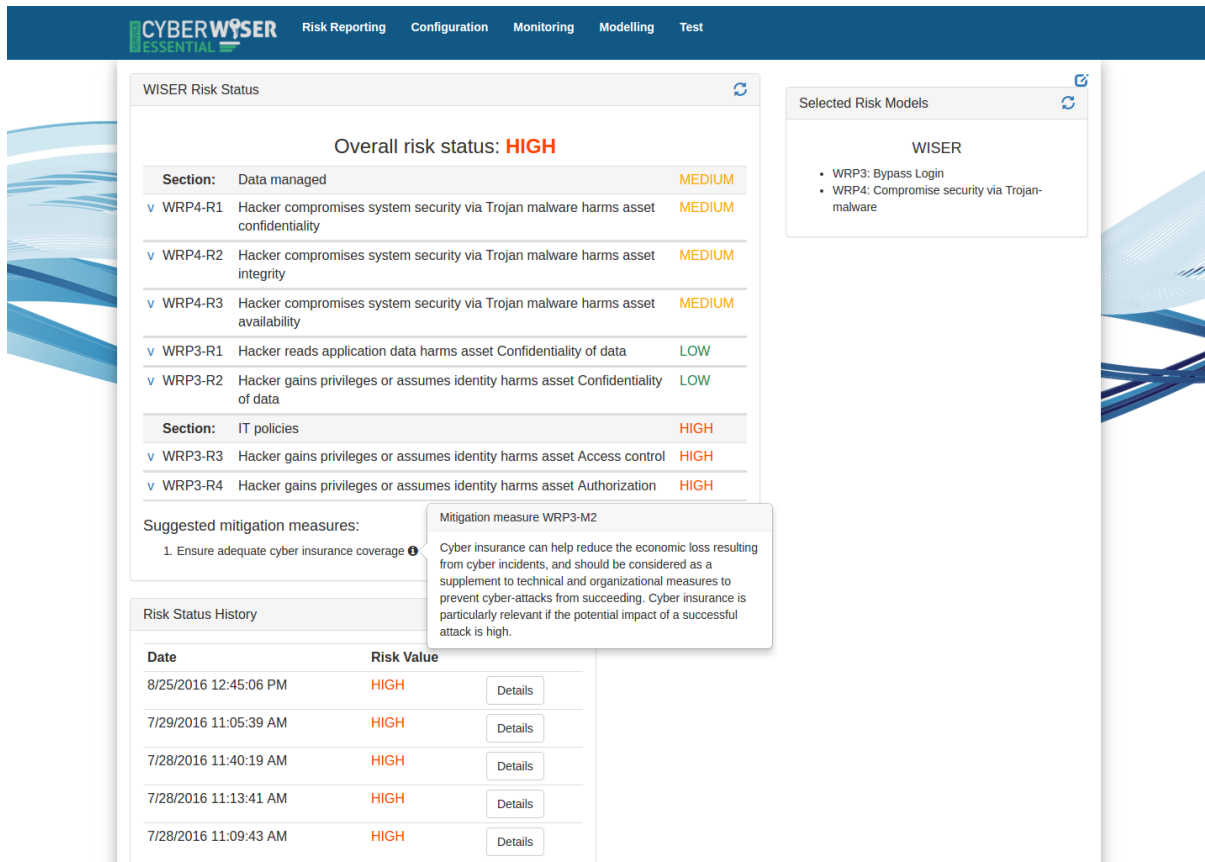
Mitigation measure name	Description
Strengthen encryption and password requirements	Inadequate encryption strength is included among CWE (Common Weakness Enumeration) as CWE-326. A weak encryption scheme can leave a system vulnerable to brute force attacks. It is therefore advisable to use a cryptographic algorithm that is currently considered strong by experts in the field <sup>14</sup> .  Weak password requirements are included as CWE-521. A policy for strong passwords should include 1) minimum and maximum length; 2) require mixed character sets (alpha, numeric, special, mixed case); 3) do not contain user name; 4) expiration; 5) no password reuse <sup>15</sup> .
Ensure adequate cyber insurance coverage	Cyber insurance can help reduce the economic loss resulting from cyber incidents, and should be considered as a supplement to technical and organizational measures to prevent cyber-attacks from succeeding. Cyber insurance is particularly relevant if the potential impact of a successful attack is high.
Install or upgrade malware detection and prevention	Anti-malware software can provide real-time protection by detecting malicious code and prevent it from being installed on a computer, or by scanning the computer and alerting the user and/or removing suspicious code that have been installed.

Table 5. Some examples of mitigation measures

The figure below depicts how the mitigation measures are shown on CyberWISER Dashboard. The short description of the measures appears initially, but if the mouse is placed over it a detailed description will be displayed.

<sup>14</sup> See <https://cwe.mitre.org/data/definitions/326.html> (accessed 14/9-2016).

<sup>15</sup> See <https://cwe.mitre.org/data/definitions/521.html> (accessed 14/9-2016).



**WISER Risk Status**

Overall risk status: **HIGH**

**Section: Data managed** MEDIUM

- WRP4-R1 Hacker compromises system security via Trojan malware harms asset confidentiality MEDIUM
- WRP4-R2 Hacker compromises system security via Trojan malware harms asset integrity MEDIUM
- WRP4-R3 Hacker compromises system security via Trojan malware harms asset availability MEDIUM
- WRP3-R1 Hacker reads application data harms asset Confidentiality of data LOW
- WRP3-R2 Hacker gains privileges or assumes identity harms asset Confidentiality of data LOW

**Section: IT policies** HIGH

- WRP3-R3 Hacker gains privileges or assumes identity harms asset Access control HIGH
- WRP3-R4 Hacker gains privileges or assumes identity harms asset Authorization HIGH

**Suggested mitigation measures:**

1. Ensure adequate cyber insurance coverage

**Mitigation measure WRP3-M2**  
 Cyber insurance can help reduce the economic loss resulting from cyber incidents, and should be considered as a supplement to technical and organizational measures to prevent cyber-attacks from succeeding. Cyber insurance is particularly relevant if the potential impact of a successful attack is high.

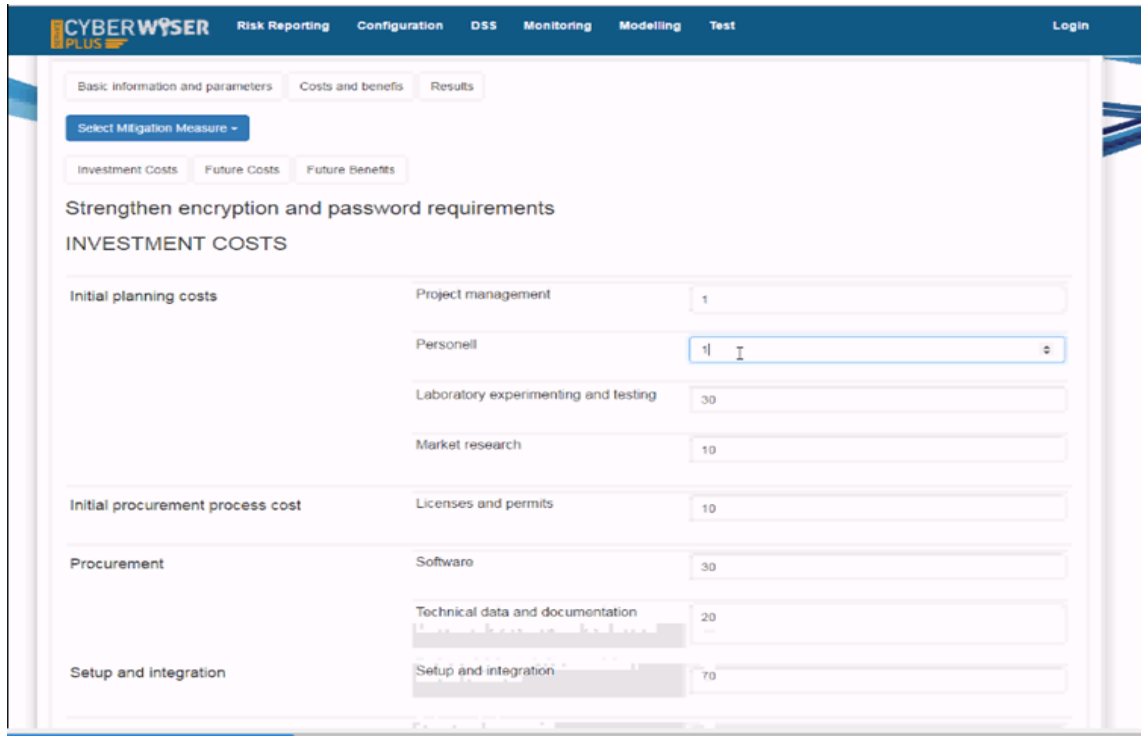
**Risk Status History**

Date	Risk Value	Details
8/25/2016 12:45:06 PM	HIGH	Details
7/29/2016 11:05:39 AM	HIGH	Details
7/28/2016 11:40:19 AM	HIGH	Details
7/28/2016 11:13:41 AM	HIGH	Details
7/28/2016 11:09:43 AM	HIGH	Details

Figure 34. Suggestion of mitigation measures. Short and detailed description of the measure

### 5.1.3 Cost-benefit analysis

WISER proposes an innovative and systematic method to compare, rank, and prioritize these mitigation measures, based on a cost-benefit analysis. The user has a template to fill (see Figure 35), where the initial and future costs, as well as the future benefits of applying a mitigation measure can be estimated. The costs and benefits are broken down into several parts by default, but the user is given the chance to customize the template by adding or removing fields. In addition, the user can choose the timeframe for which this analysis is performed, establish a discount rate, choose the currency or express some constraints. The cost-benefit analysis methodology, along with the detail of the template to fill in, are described in Deliverable D5.1, Section 5.3.



Category	Item	Value
Initial planning costs	Project management	1
	Personell	1
	Laboratory experimenting and testing	30
	Market research	10
Initial procurement process cost	Licenses and permits	10
Procurement	Software	30
	Technical data and documentation	20
Setup and integration	Setup and integration	70

Figure 35. Decision Support System. Cost-benefit analysis. Filling out the template

The result inspection allows comparing different measures by looking into different parameters, such as: the total amount of investment, the future costs and benefits; the payback period, which allows the calculation of the break-even point or the cost-benefit ratio; or the cashflow. These are some key indicator figures which can be represented in graphs for a better understanding

As an outcome, the user is offered some well-based criteria to make the final choice. The cost-benefit analysis enables the decision-maker to compare all, direct and indirect, positive and negative effects of the proposed decisions. This is possible by providing key people with an understanding of the economic costs and benefits of decisions, allowing arguments to be made for or against based on economic considerations, and brings issues into discussions and ensures better transparency of decisions (see Figure 36).

This is a specific feature of CyberWISER-Plus.



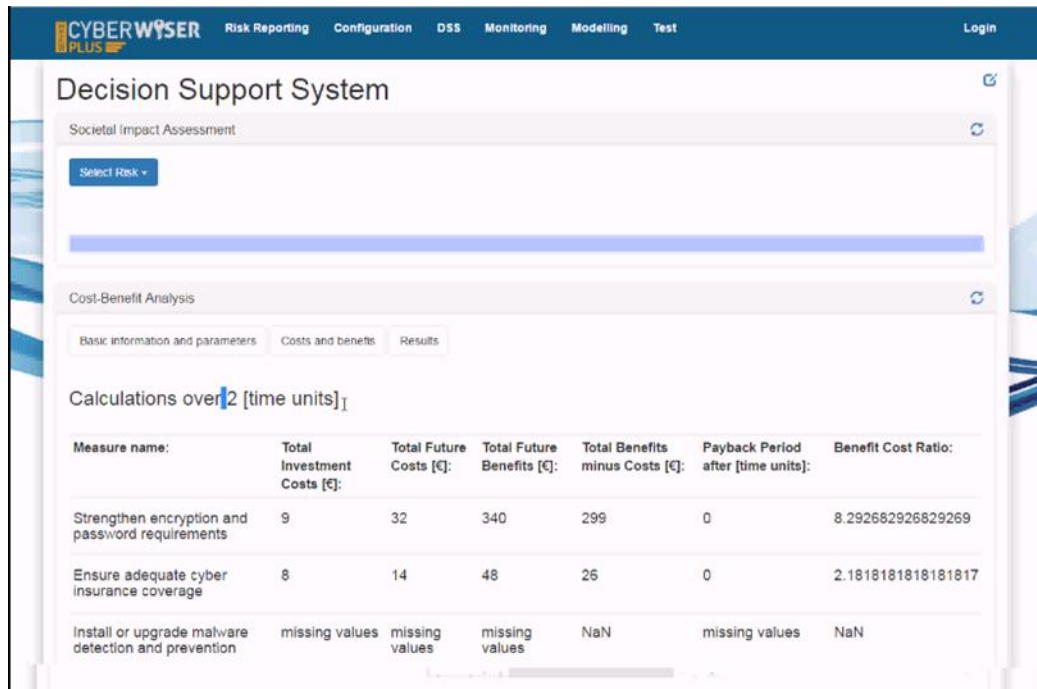


Figure 36. Decision Support System. Cost-benefit analysis. Result inspection

#### 5.1.4 Societal impact analysis

WISER offers the calculation of the societal impact of the risks being evaluated. This is done by answering a questionnaire where intangible (but very important) topics related to risk are addressed. The questions are grouped into the following categories:

- **Society:** This is the largest category, where the most important societal effects are recorded. WISER will consider criteria such as Social cohesion, Trust in fellow citizens, Emotions, Social alertness, Job quality and labour market, Education, Reputation, Interplay with media, Consumption, Market & trade relations, as drivers capable of affecting the quality of the societal environment and society in its entirety.
- **Individual:** In this category WISER will record the effect on the individual as part of the society, considering motivation and quality of life / comfort as the main drivers
- **Law:** Company accountability is considered as the main driver for this category, where the impact on the judicial system is recorded.
- **Rights and ethics:** This category encompasses values considered by the European Union as fundamental for European society: privacy, personal data and liberty, freedoms of thought, conscience, religion, expression, information, movement.
- **Politics:** In this category the potential effect on politics will be taken into account, considering the culture of control/authority and trust.
- **Environment:** In this category the potential effect on the environment, considered as a societal asset, will be considered. Potential environment effects and the reaction on the environmental organization will be evaluated.

There is a total of 19 questions which are posed to the user per risk being analysed. Each question has 5 possible answers: low, medium-low, medium, medium-high or high. The table below shows the

19 questions, the criterion to which each one corresponds and how the criteria are grouped into categories

Category	Criterion	Question
<b>Society</b>	Social cohesion	Does the risk entail social tensions?
	Trust in fellow citizens	Does the risk harm the trust in fellow citizens?
	Emotions	Does the risk provoke fear frustration, anger, etc?
	Social alertness	Does the risk produce social alertness?
	Job quality and labour market	Does this risk affect job quality? Does it increase the demand of jobs or the loss of jobs?
	Education	Do people know about this risk?
	Reputation	Does the risk influence the internal and external reputation of the victim?
	Interplay with media	How will the media react to this risk?
	Consumption	Does the risk influence consumption behavior?
	Market & trade relations	Does the risk influence the capacity of the company to compete in the market?
<b>Individual</b>	Motivation	How does the risk influence motivation to work, commit, etc?
	Quality of life / comfort	Does the risk affect quality of life or comfort?
<b>Law</b>	Accountability	How is the impact of the risk from the point of view of company's accountability?
<b>Rights and ethics</b>	Privacy, personal data and liberty	Does the risk impact in privacy, family life, personal data protection, liberty and security of individuals?
	Freedoms of thought, conscience, religion, expression, information, movement	Does risk compromise the freedoms of thought, conscience, religion, expression, information and movement?
<b>Politics</b>	Culture of control / authority	Could this risk entail promoting and perpetuating a 'culture of control', where authorities have

		an overview / power over people's movements, bodies and actions?
	Trust	Does the risk affect trust in politics?
<b>Environment</b>	Hidden effects	Does the risk involve any chance of hidden environmental effects?
	Organization	Should reactions of national and international environmental organizations be expected or considered?

Table 6. Societal impact questionnaire

The qualitative answers are translated into numbers by means of utility functions and then they are weighted in order to come with an overall evaluation of the societal impact of the risk. The table below shows an example of a possible configuration of utility functions and weighting.

Category	Criterion	Question	Utility function
<b>Society: 50%</b>	Social cohesion: 5%	Does the risk entail social tensions?	[L,M-L,M,M-H,H] [1,4,7,9,10]
	Trust in fellow citizens: 2%	Does the risk harm the trust in fellow citizens?	[L,M-L,M,M-H,H] [1,4,7,9,10]
	Emotions: 20%	Does the risk provoke fear frustration, anger, etc?	[L,M-L,M,M-H,H] [1,3,6,8,10]
	Social alertness: 2%	Does the risk produce social alertness?	[L,M-L,M,M-H,H] [1,3,5,7,9]
	Job quality and labour market: 15%	Does this risk affect job quality? Does it increase the demand of jobs or the loss of jobs?	[L,M-L,M,M-H,H] [3,4,5,6,7]
	Education: 3%	Do people know about this risk?	[L,M-L,M,M-H,H] [0,2,6,8,10]
	Reputation: 20%	Does the risk influence the internal and external reputation of the victim?	[L,M-L,M,M-H,H] [0,2,6,8,10]
	Interplay with media: 10%	How will the media react to this risk?	[L,M-L,M,M-H,H] [0,2,6,8,10]
	Consumption: 3%	Does the risk influence consumption behavior?	[L,M-L,M,M-H,H] [2,4,5,6,8]
	Market & trade relations: 20%	Does the risk influence the capacity of the company to compete in	[L,M-L,M,M-H,H] [3,5,7,9,10]

		the market?	
<b>Individual: 25%</b>	Motivation: 60%	How does the risk influence motivation to work, commit, etc?	[L,M-L,M,M-H,H] [3,5,7,9,10]
	Quality of life / comfort: 40%	Does the risk affect quality of life or comfort?	[L,M-L,M,M-H,H] [3,5,6,8,10]
<b>Law: 3%</b>	Accountability: 100%	How is the impact of the risk from the point of view of company's accountability?	[L,M-L,M,M-H,H] [3,5,7,9,10]
<b>Rights and ethics: 10%</b>	Privacy, personal data and liberty: 60%	Does the risk impact in privacy, family life, personal data protection, liberty and security of individuals?	[L,M-L,M,M-H,H] [3,5,7,9,10]
	Freedoms of thought, conscience, religion, expression, information, movement: 40%	Does risk compromise the freedoms of thought, conscience, religion, expression, information and movement?	[L,M-L,M,M-H,H] [3,5,7,9,10]
<b>Politics: 2%</b>	Culture of control / authority: 90%	Could this risk entail promoting and perpetuating a 'culture of control', where authorities have an overview / power over people's movements, bodies and actions?	[L,M-L,M,M-H,H] [3,5,7,9,10]
	Trust: 10%	Does the risk affect trust in politics?	[L,M-L,M,M-H,H] [2,4,6,7,10]
<b>Environment: 10%</b>	Hidden effects: 75%	Does the risk involve any chance of hidden environmental effects?	[L,M-L,M,M-H,H] [3,5,7,9,10]
	Organization: 25%	Should reactions of national and international environmental organizations be expected or considered?	[L,M-L,M,M-H,H] [2,4,6,7,10]

Table 7. Societal impact evaluation. Example of configuration of the utility functions and weights to process the answers to the questionnaire.

CyberWISER-Essential will preconfigure the utility functions and the weights. On the contrary, CyberWISER-Plus will allow the users to configure the utility functions and the weights according to their own needs, with the support of consultants.

Therefore, WISER proposes an innovative approach to consider in the overall evaluation of risk not only the tangible effects, which can be clearly perceived, but also those that, despite not being tangible, should not be neglected at all, as their influence is much more notable than thought.

This feature is offered by both CyberWISER-Essential and CyberWISER-Plus.

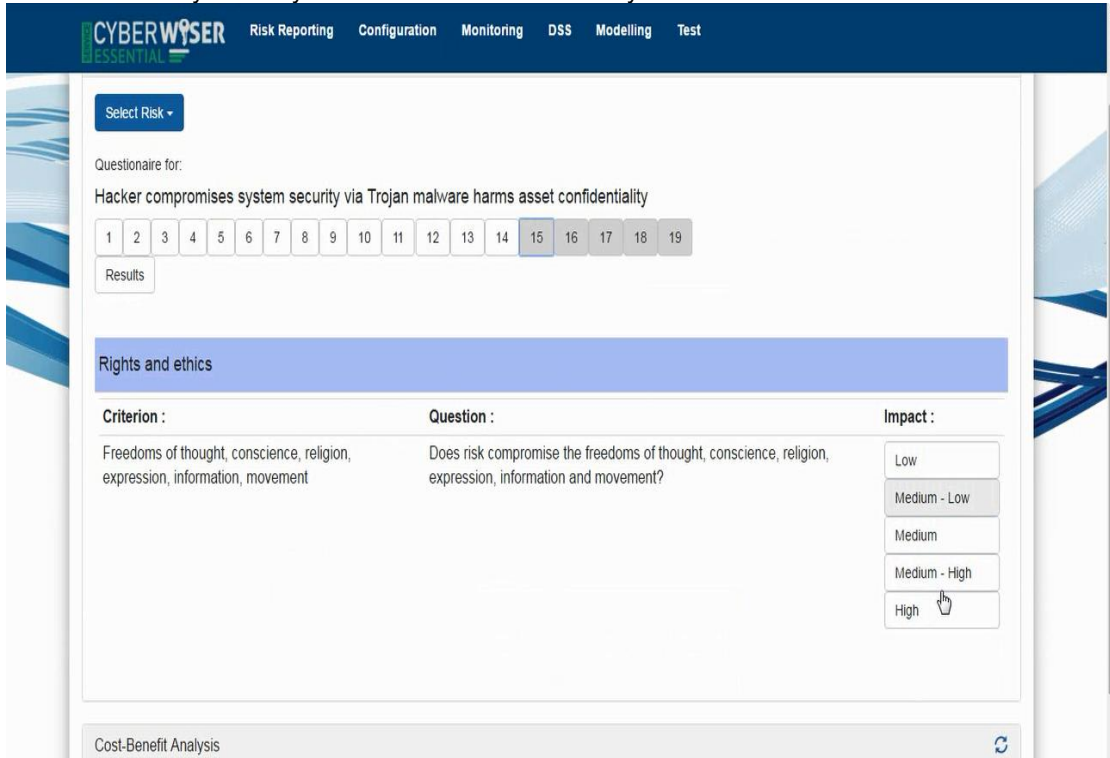


Figure 37. Decision Support System. Societal impact analysis. Questionnaire

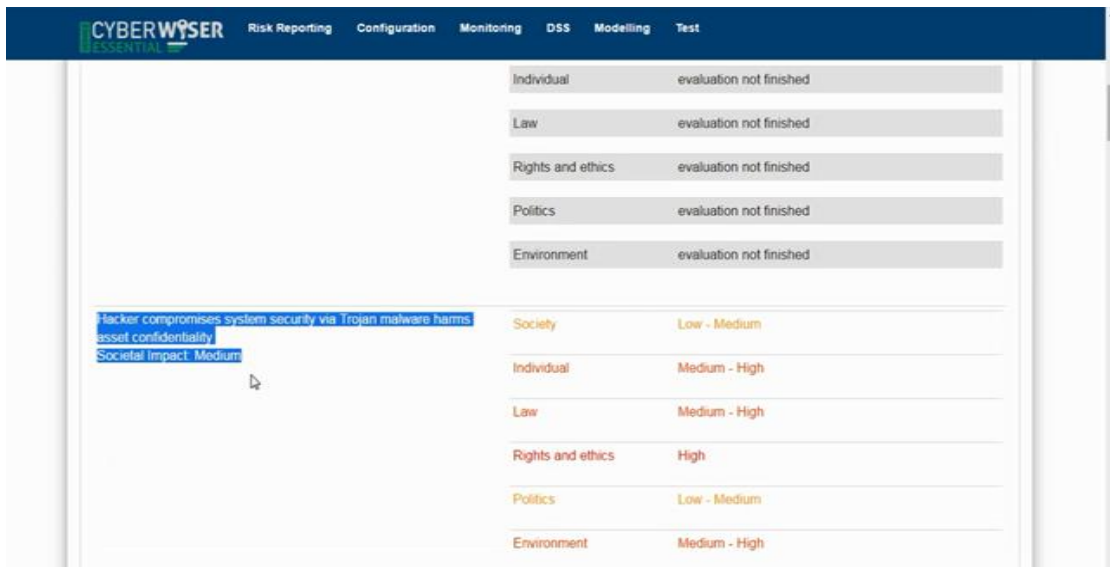


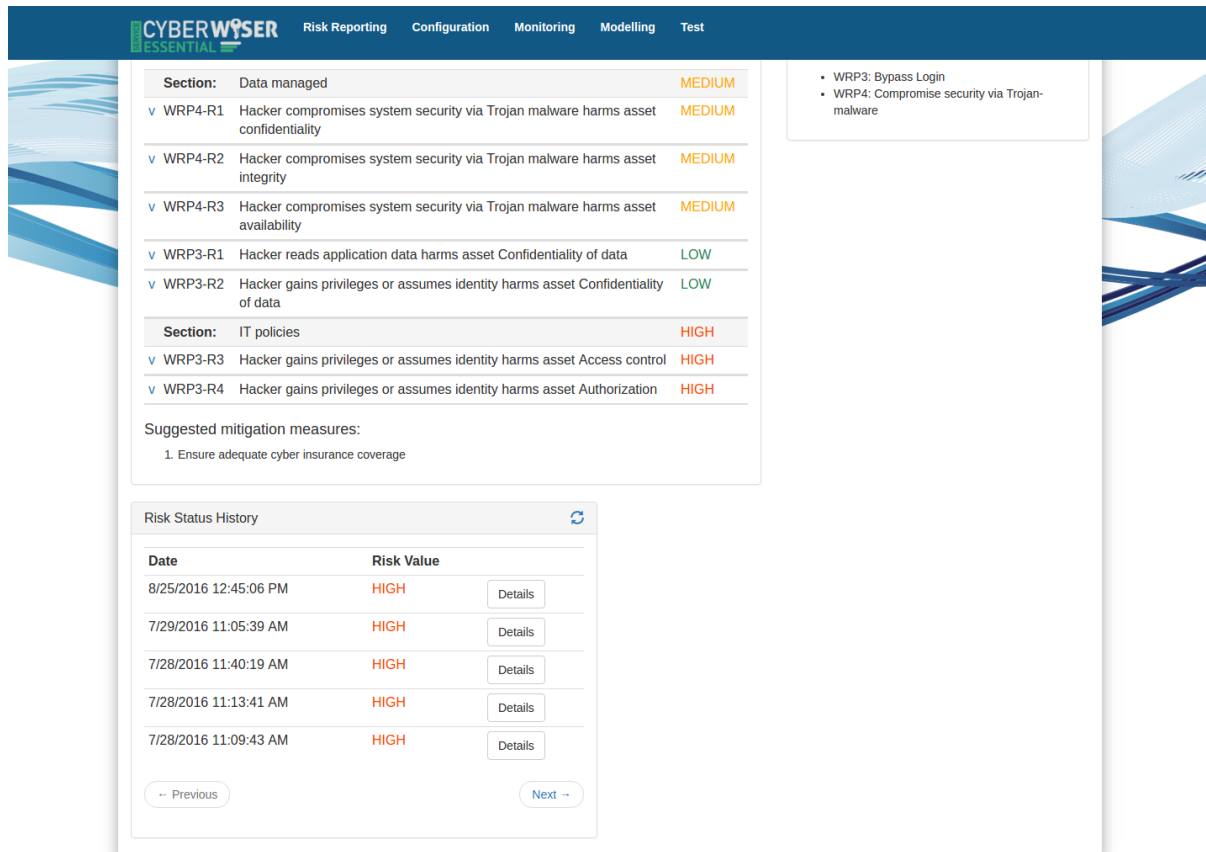
Figure 38. Decision Support System. Societal impact analysis. Results

### 5.1.5 Support to dashboard visualization

As described in Figure 31, Figure 32, and Figure 33, the Decision Support System provides the needed logics to show information about the risk analysis and mitigation measures.

First, it provides the visualization of the risk reports. The reports are described in Section 3.1.3. The information can be shown both in the dashboard itself and be converted into PDF in order to be downloaded.

Second, the user is offered the chance to look for older risk reports, previously stored in the Data Warehouse. To ease the look-up process, the user will establish filtering criteria and those reports meeting the criteria will be shown. Each row will correspond to a report. The information shown in this list is very basic and it is limited to a report ID, a report issuing timestamp, and the risk level reported (specified in a qualitative scale with a quantitative estimation).



The screenshot shows the CYBERWISER dashboard interface. At the top, there is a navigation bar with the following tabs: Risk Reporting, Configuration, Monitoring, Modelling, and Test. The main content area is divided into several sections:

- Section: Data managed (MEDIUM)**
  - WRP4-R1: Hacker compromises system security via Trojan malware harms asset confidentiality (MEDIUM)
  - WRP4-R2: Hacker compromises system security via Trojan malware harms asset integrity (MEDIUM)
  - WRP4-R3: Hacker compromises system security via Trojan malware harms asset availability (MEDIUM)
  - WRP3-R1: Hacker reads application data harms asset Confidentiality of data (LOW)
  - WRP3-R2: Hacker gains privileges or assumes identity harms asset Confidentiality of data (LOW)
- Section: IT policies (HIGH)**
  - WRP3-R3: Hacker gains privileges or assumes identity harms asset Access control (HIGH)
  - WRP3-R4: Hacker gains privileges or assumes identity harms asset Authorization (HIGH)
- Suggested mitigation measures:**
  - Ensure adequate cyber insurance coverage
- Risk Status History** (Table):
 

Date	Risk Value	Details
8/25/2016 12:45:06 PM	HIGH	Details
7/29/2016 11:05:39 AM	HIGH	Details
7/28/2016 11:40:19 AM	HIGH	Details
7/28/2016 11:13:41 AM	HIGH	Details
7/28/2016 11:09:43 AM	HIGH	Details

Figure 39. Older risk report visualization feature

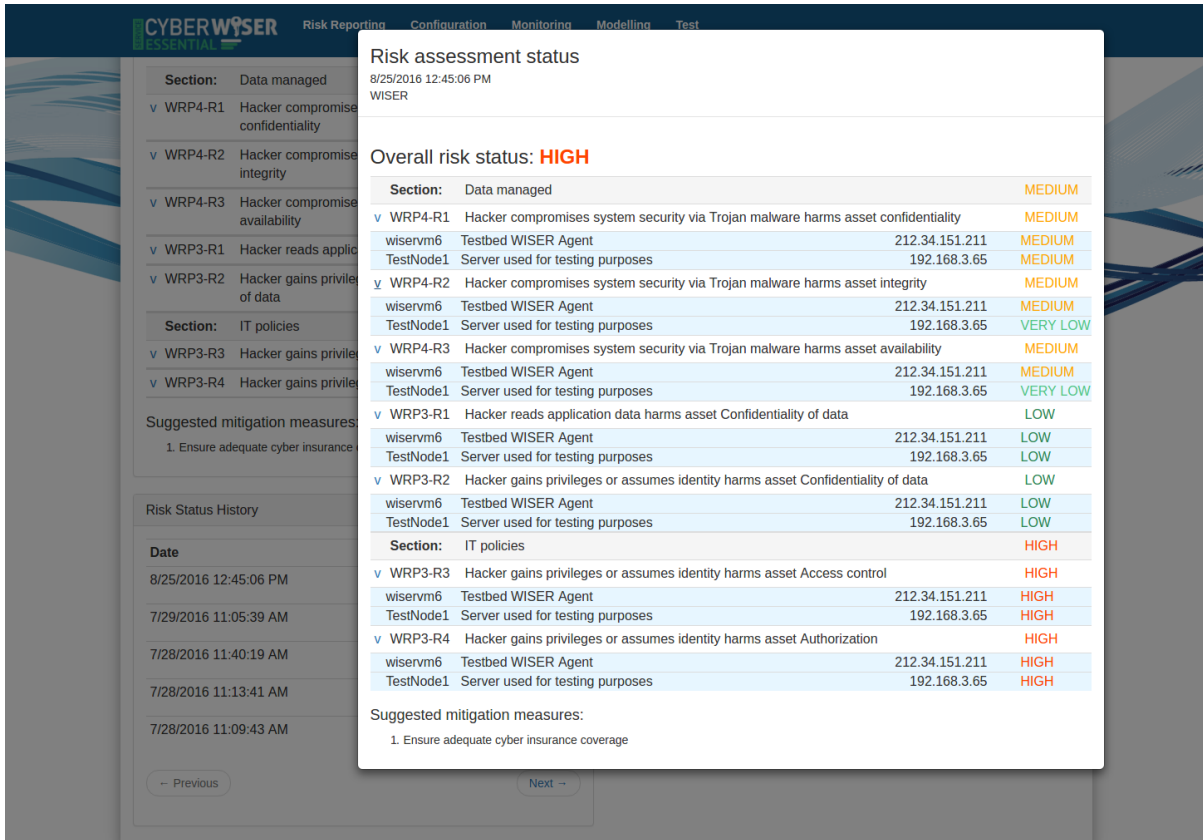


Figure 40. Older report visualization. Report detail

Third, the dashboard offers a risk analytics feature where the user can consult in a graphical way information such as the evolution of the risk assessments performed by means of the different models or the current risk status of the different elements of the target infrastructure in a nice and intuitive way, thanks to evolution curves or heat maps, for instance. Some examples of the kind of graphics that may be found: overall risk evolution over time, bars representing one many risks are in each of the possible status (very low, low, medium, high, very high), current risk status vs average for a certain timeframe, bars representing the risk per target, bar representing risks per pattern, or target map indicating the risk level by means of colours, for instance. This feature is specific of CyberWISER-Essential (a reduced version) and CyberWISER-Plus (an expanded version).

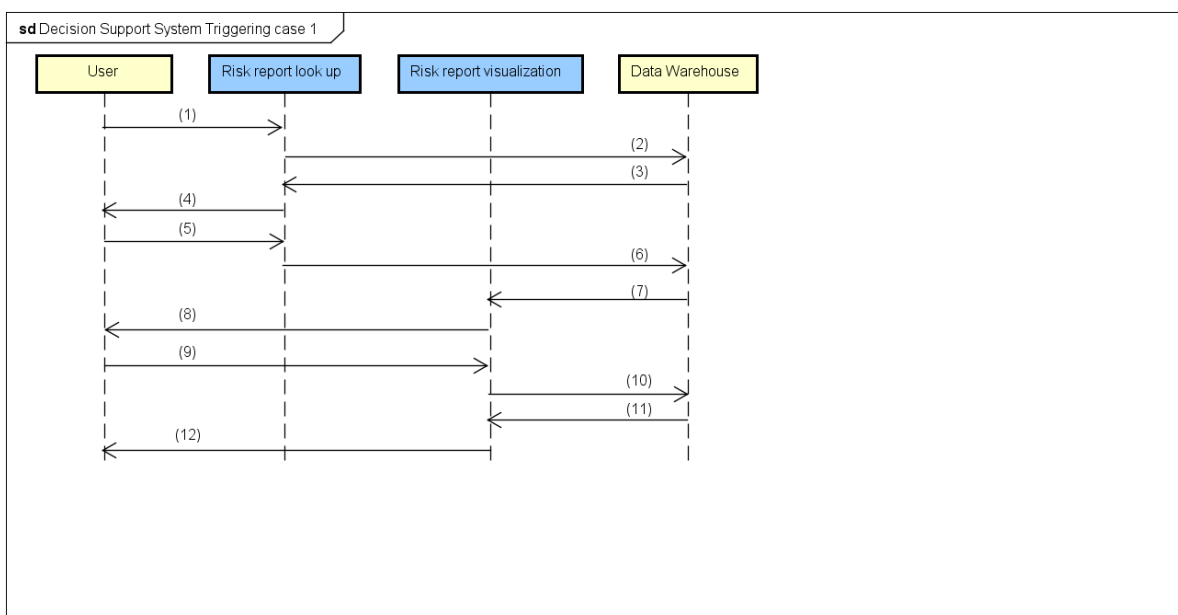
Fourth, an interface is offered to the user in order to answer the questionnaire linked to the societal impact of each of the risks being evaluated. In addition, once the questions are answered, the dashboard shows the result of the evaluation to the user. This feature is specific of CyberWISER-Essential and CyberWISER-Plus.

Finally, the user can go to the cost-benefit tool to evaluate which mitigation measures are more convenient to apply from the monetary aspect. The Decision Support System contains the logic of the tool, which guides the user through a set of steps which have to be accomplished to make possible this cost-benefit analysis and to extract valuable conclusions from it. The user interacts with this tool by means of the dashboard. The steps followed are explained in D5.1, Section 5.3 and a short outline about this feature is provided in Section 5.1.3. It is paramount the fact that two or more mitigation actions can be compared from different points of view, and this comparison is reported by means of different kinds of charts. It is important to highlight that the cost-benefit analysis is linked to a particular report produced by the Risk Assessment Engine. This means that deleting a report produced by the engine would imply deleting the whole cost-benefit analysis of the mitigation measures.

### 5.1.6 Triggering cases

In the following the different cases when the Decision Support System is triggered are presented.

The first one is presented in Figure 41. It is specific of CyberWISER-Essential and CyberWISER-Plus. The user interacts with the Dashboard to obtain the look up of the risk assessment reports available (1). In particular, he interacts with the module in charge of the risk report look up. To obtain this information, this module queries the Data Warehouse and retrieves the data from there. This information is shown to the user (4). Then, by interacting with the Dashboard, the user chooses one of the reports, the one he is interested in (5) and the details are obtained from the Data Warehouse (6)(7) and shown to the user by means of the component for visualization of reports (8). Then the user, by placing the mouse over the text briefly describing a mitigation measure (9), can obtain an extended description of the measure (10)(11) that is displayed on the Dashboard (12).

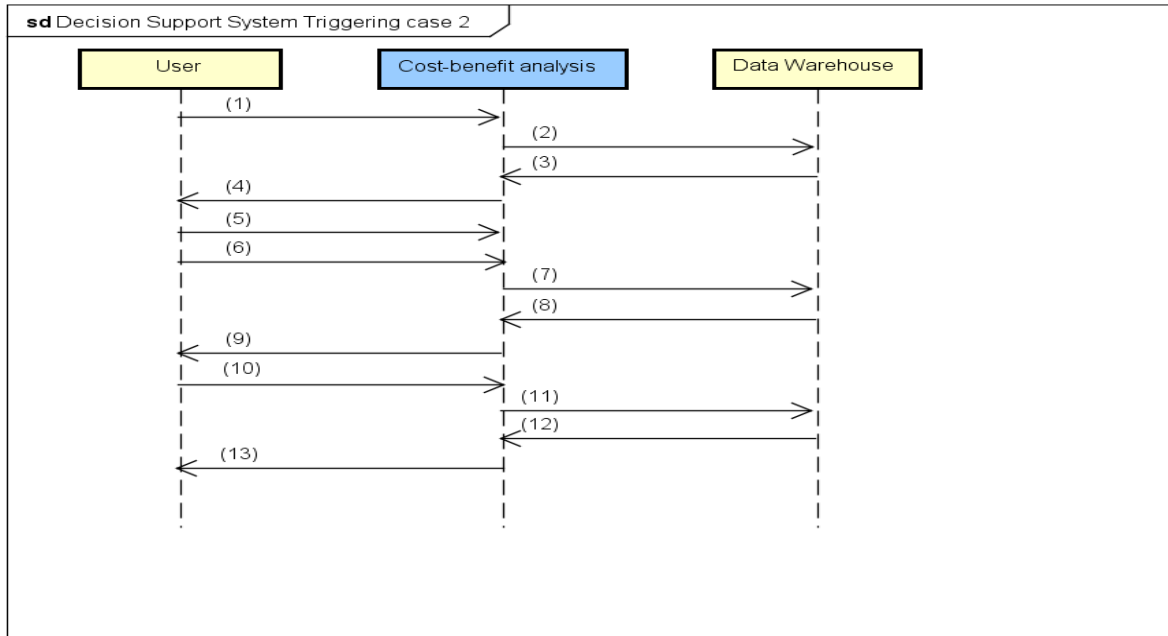


powered by Astah

Figure 41. Decision Support System Triggering Case 1: the user requests information about mitigation measures

The second case (see Figure 42) is when a user performs a cost-benefit analysis of mitigation actions. This is specific of CyberWISER-Plus. The user chooses the module for cost-benefit analysis (1). Once activated, this module retrieves the mitigation actions being active (this is, those which are recommended in the latest risk assessment report produced by the Risk Assessment Engine) (2)(3) and includes this information in the drop down menu from which the user will choose the mitigation measure/s to analyse (4). The user chooses the mitigation measure from the combo menu (5) and shapes the cost-benefit analysis structure according to his needs and introduces the figures for the cost-benefit analysis process (6)(7). The user is confirmed that the information was introduced (8)(9). Then, the user calls the program to produce the results (10). The program retrieves the information from the Data Warehouse (11)(12) and prepares the results presentation by means of summarized scores and graphs (13).

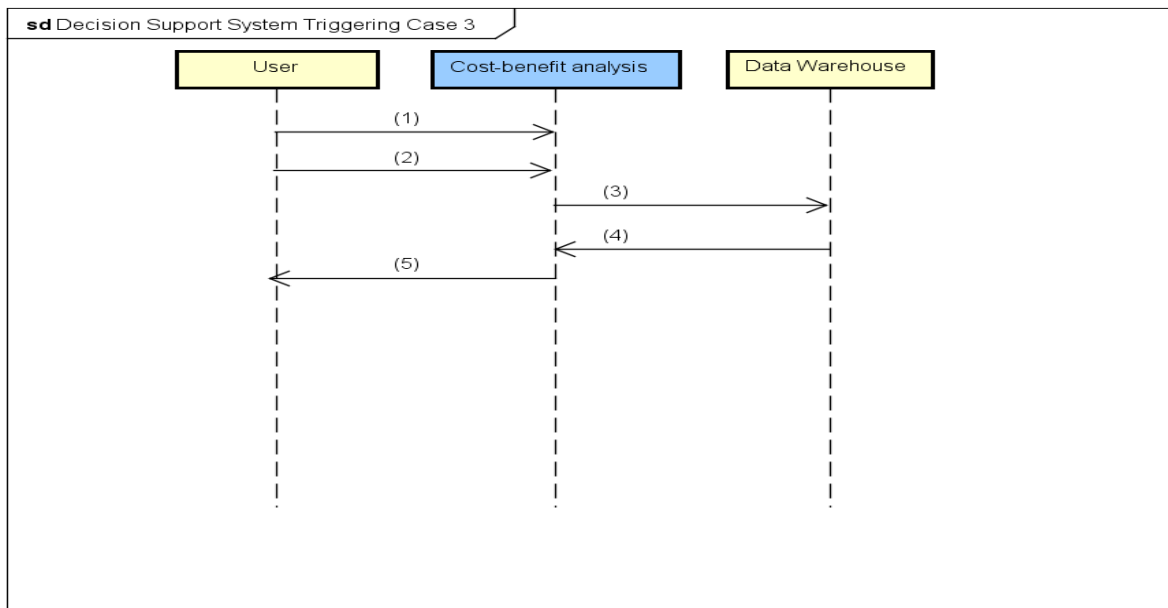




powered by Astah

Figure 42. Decision Support System Triggering Case 2: cost-benefit analysis of mitigation measures

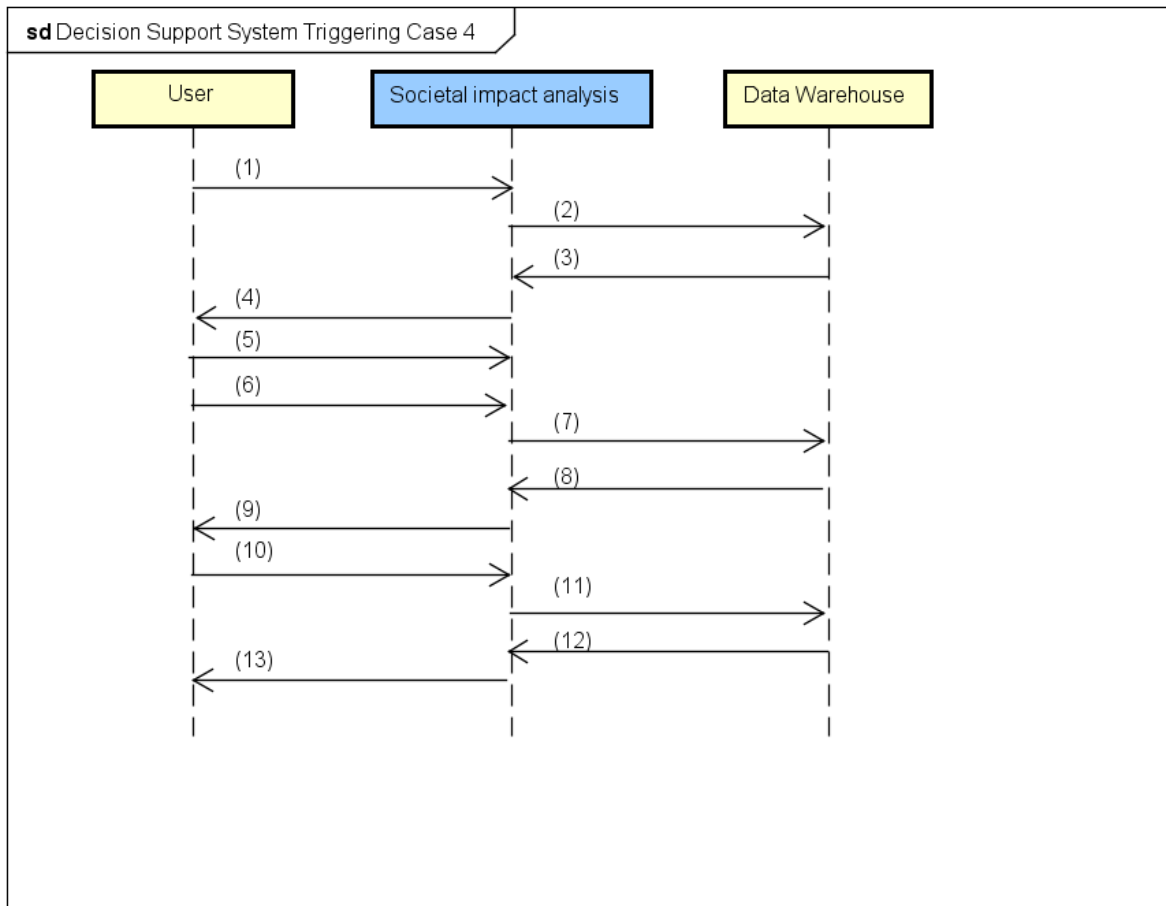
The third case (see Figure 43) is related to the comparison of mitigation measures. It is specific of CyberWISER-Plus. It takes as a starting point the assumption that two or more mitigation measures were analysed in cost-benefit terms. The user chooses the measure to compare (1) and launches the comparison (2). The cost-benefit analysis logic interacts with the Data Warehouse to retrieve the needed data (3)(4) and presents the user the results of the comparison (5).



powered by Astah

Figure 43. Decision Support System Triggering Case 3. Comparison of mitigation measures in cost-benefit terms.

The fourth case (see Figure 44) considers the Societal Impact Analysis. This is specific of CyberWISER-Essential and CyberWISER-Plus. The user goes to the societal impact analysis feature within the Dashboard (1). This activates the retrieving of the risks being evaluated in the model/s being active (2)(3). These risks are fed into a drop down menu, that is shown to the user (4). The user selects the risk for which the societal impact is to be analysed (5). Then, he answers the 19 questions posed by means of a questionnaire (6). The answers are stored in the Data Warehouse (7) and the user is confirmed this storage (8)(9). Then, the user initiates the execution the algorithm to evaluate the societal impact (10). The Data Warehouse is queried to obtain the answers the user previously provided to the questionnaire (11)(12), the calculations to evaluate the answers are carried out and the results displayed to the user (13).



powered by Astah

Figure 44. Decision Support System Triggering Case 4: Societal Impact Analysis of risks

## 5.2 Implementation

### 5.2.1 Presentation layer

The Decision Support System widgets are available on the dashboard made available with the use of Angular Dashboard Framework<sup>16</sup> (ADF). The ADF itself consists of 3 parts: the framework, structure and widgets. The Decision Support System is a set of widgets on the ADF dashboard: decision-support, risk-status, risk-status-history, risk-status-quantitative-history and risk-status-quantitative. Decision-support widget provides presentation of the cost-benefit analysis (the input) widget. Risk-

<sup>16</sup> <https://github.com/angular-dashboard-framework/angular-dashboard-framework>

status and risk-status-history widgets provide an overview of the overall risk status as shown in Figure 34. Moreover, it provides mitigation measures based on the risks found in the report. Risk-status-quantitative and risk-status-quantitative-history are similar to the former two: they provide overview of the risk status, but from quantitative aspects (reports as evaluated by RAE and DSS components on the Business Layer).

### 5.2.2 Business layer

The DSS is implemented as a standalone server exposing a REST API that provide means to assess risks in terms of their societal impacts and perform a cost benefit analysis of available security measures. The DSS has two dedicated widgets which serve the purpose of the user interface as well as data visualisation, although the user could implement his own visualisations and interfaces as the server is mainly used for computational purposes.

#### 5.2.2.1 Societal impact

The societal impact analysis is implemented as an interactive widget in the dashboard, consisting of two main views.

##### 5.2.2.1.1 Societal impact analysis

The module is represented by the first view of the widget implemented in the dashboard communicating with the DWH and the DSS server. The DWH is queried for the lists of risks that are of concern for the user. Once obtained, the DSS server “attaches” the category-criteria structure defined in Table 6 of this document.

The DSS server also provides a preconfigured set of utility functions and weights related to each criterion and category. It is not advisable to modify this configuration as the user should not be aware of the mapping and potential results.

The same set of societal impact criteria are assessed per risk.

The user interface facilitates the analysis by highlighting the parts of the analysis that were not finalised yet. This is also indicated in the results section described in the next section.

The user interface does not hold the mapping of utility functions, what means that the user cannot directly change the parameters of the algorithm.

##### 5.2.2.1.2 Societal impact analysis results

This module is implemented as the second view of the societal impact analysis widget. It provides a simple visualisation of the results mapping the numerical values accordingly to a utility function, providing a qualitative assessment per category, but also general assessment of the risk impact.

The results visualisation provides a result for a category if the whole category was assessed by the user. This is also true for the general risk assessment, which requires all categories to be completed.

All the computations and mappings are performed by the DSS server.

#### 5.2.2.2 Cost-benefit Analysis

The cost-benefit analysis is implemented as an interactive widget, consisting of three main views

##### 5.2.2.2.1 Basic info and calculation parameters

This view provides five input fields:

**Goal of analysis** – an optional field where the user may describe the goal of the analysis for informative purposes

**Viewpoint of analysis** – an optional field where the user may provide the description of the stakeholder committing the analysis.

**Time Range [time unit]** – the number of time units that the analysis corresponds to – it maybe be treated as days, weeks or years, depending on the user preference.

**Budget [units]** – the number of units that are a limit value for the analysis used in the computations – the units are a symbolic value, usually referred to particular currencies.

**Comments** – additional, optional field where the user may provide his comments.

The input fields have a validation check assuring the user inputs only valid values for each field.

#### 5.2.2.2.2 *Cost-benefit breakdown structure*

This view provides a list of security actions and three main cost-benefit categories per each:

**Investment costs** – allows entering one value per sub-category

**Future costs** – allows entering one value, that will be assigned to each time unit in the time frame, or multiple values, each corresponding to a specific time unit in the analysis (the time frame is the number of time units specified in the basic info view).

**Future benefits** – same as the Future costs, allows entering one value, that will be assigned to each time unit in the time frame, or multiple values, each corresponding to a specific time unit in the analysis (the time frame is the number of time units specified in the basic info view).

The sub-categories for each of the main categories are described in D5.1, Section 5.3. Nevertheless, the user is not restricted to the list and may delete or add new categories from each list.

#### 5.2.2.2.3 *Cost-benefit results*

The results view provides a set of numerical values, calculated by the DSS server, per security action. The values are the following:

**Total Investment Costs** – the overall investment value for the action.

**Total Future Costs** – the overall cost for the whole duration of the time frame.

**Total Future Benefits** – the overall benefits value for the time frame duration.

**Payback Period** – the time unit after which the benefits exceeded the costs.

**Benefit Cost Ratio** – overall benefit-cost ratio for the timeframe (the higher the better).

The results are sorted accordingly to the Benefit Cost Ratio to provide a ranking of the most beneficial security action.

## 5.3 Deployment

### 5.3.1 *Pre-requisites*

#### 5.3.1.1 Presentation layer

Main requirements for deploying the presentation layer of DSS (the dashboard with the set of aforementioned widgets) is a host machine with Docker<sup>17</sup> engine preinstalled and presentation layer code.

#### 5.3.1.2 Business layer

- Python 2.7 or later. The Python module 'tornado'<sup>18</sup> is required:
- From the server where the Decision Support System is installed, it is necessary to have access to the Data Warehouse endpoint.

<sup>17</sup> <https://www.docker.com/>

<sup>18</sup> <http://www.tornadoweb.org/en/stable/>

- The Decision Support System has been tested on Windows 7.

### 5.3.2 Installation Procedure

#### 5.3.2.1 Business layer

The DSS is provided as a light-weight http python web server exposing a REST API. Once extracted, it can be started by running the following command in the root folder of the DSS:

```
# python main.py
```

By default the server is started on the 8001 port, but this can be modified in the config.json file modifying the “port” attribute.

#### 5.3.2.2 Presentation layer

Presentation layer of DSS is part of the complete ADF dashboard. It is installed and deployed using Docker commands. Below we provide a list of commands necessary to run the ADF dashboard with DSS widgets:

```
$ sudo docker run --rm -v "${PWD}":/usr/share/nginx/html:ro -p 8084:80 nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
8ad8b3f87b37: Pull complete
c6b290308f88: Pull complete
f8f1e94eb9a9: Pull complete
Digest: sha256:aa5ac743d65e434c06fff5ceaab6f35cc8519d80a5b6767ed3bdb330f47e4c31
Status: Downloaded newer image for nginx:latest
```

### 5.3.3 How to verify the installation

#### 5.3.3.1 Presentation layer

This part can simply be verified via visiting a localhost:8084 on a machine, where the deployment is done.

#### 5.3.3.2 Business layer

Once the starting command is executed, the server does a self-test before going into the operational mode. When no problems are encountered, the server outputs in the console:

```
DSS server started on port: 8001
```

## 5.4 Operation

### 5.4.1 User Manual

#### 5.4.1.1 Presentational layer

The use of the widgets is very simple and straightforward. A user can simply navigate through the provided user interface (see Figure 45). The interface consists of the main menu on the top part of the user interface and set of widgets on the rest of the interface. The main menu consists of several sections: Risk Reporting, Configuration, Monitoring, Modelling and Test. DSS components are provided under Risk Reporting section, where the set of DSS widgets described in this section are located. Each section is accompanied with “enable edit mode” option on the top-right section of the UI.

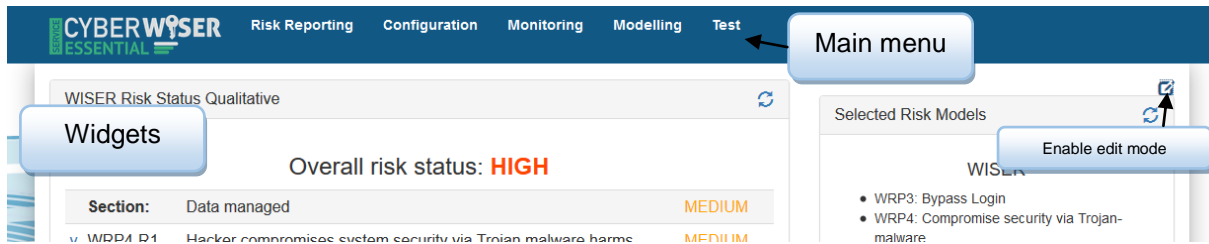


Figure 45: DSS presentation layer.

When the widgets are opened in edit mode (see Figure 46), there are several additional options user can use to configure the widget itself: reload the content, change the widget's location, edit the widget's configuration, and remove the widget from the view.

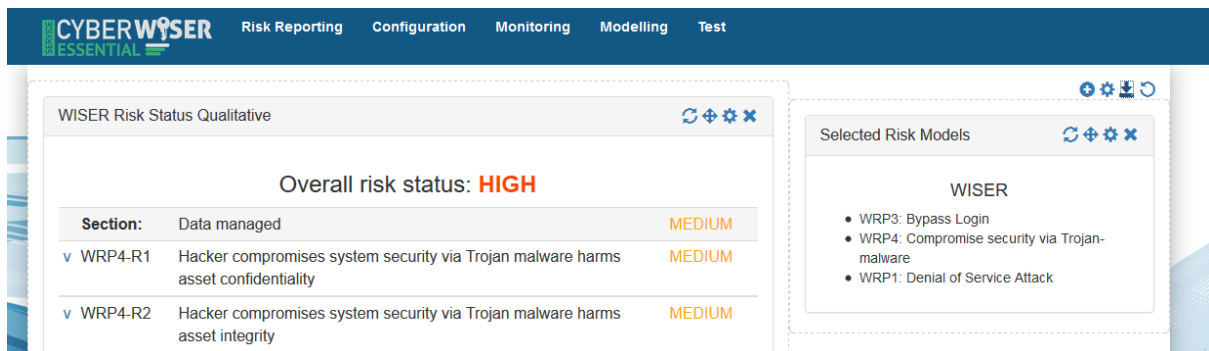


Figure 46: Widgets in edit mode.

#### 5.4.1.2 Business layer

The DSS Server is connected closely to the dashboard widgets which serve as a user interface for the analysis. The communication between the dashboard and the DSS is transparent to the user as each component is preconfigured.

The cost-benefit categories structure is predefined, but can easily be modified by the user, being directly editable from the dashboard.

Since the societal impact is more of a questionnaire, where the user is obliged to answer all questions, it is not advisable to modify neither its structure nor the scoring mapping, although we leave this ability to the system operator. The structure of the societal impact categories can be modified in the DSS server config file, modifying the societal\_impact\_categories attribute. The interface automatically renders the structure, and uses the mapping for generating the results.

An example of a category structure:

```
{
  "name": "Individual",
  "weight": 0.25,
  "criteria": [
    {
      "name": "Motivation",
      "weight": 0.6,
      "question": "How does the risk influence motivation to work, commit, etc?",
      "utility_function": [3,5,7,9,10],
      "value": -1
    },{
      "name": "Quality of life / comfort",
      "weight": 0.6,

```

```
"question": "Does the risk affect quality of life or comfort?",  
"utility_function": [3,5,6,8,10],  
"value": -1  
  }  
}
```

## 6 Summary and Conclusions

The goal of this document is to report the implementation, deployment, and operation of the Real-Time Assessment Infrastructure. It is composed of three modules, namely: 1) The Risk Assessment Engine, 2) The Data Warehouse, and 3) The Decision Support System.

The WISER Real-Time Assessment Infrastructure allows shifting away from a purely technical assessment of cyber risk, only considering the information obtained by testing and monitoring the target infrastructure, to an assessment which considers the business aspect linked to the threatened elements of such infrastructure. Always in the light of the three different WISER Services, the role played by each of the components is briefly explained below:

- 1) **Risk Assessment Engine** implements the logic needed in order to obtain a report on the assessment of the risk faced. It puts the cyber incidents in the necessary context to evaluate their impact in the company business process. It uses modelling techniques with an associated algorithm which executes machine-readable model rules. As part of the report, it suggests mitigation measures according to the risks evaluated. The algorithm is executed in near real-time.
- 2) **Decision Support System** includes a web-based dashboard where the results of the computations carried out by the Risk Assessment Engine are shown. This includes both the qualitative and quantitative assessment of the risk. It performs the societal impact assessment of cyber risks based on quality criteria. It performs the cost-benefit analysis of implementing a mitigation measure, enabling the decision maker to compare all direct and indirect positive and negative effects of the proposed decisions. As for the mitigation measures, it uses the suggestions received from the Risk Assessment Engine based on the risk level, and then prioritizes the mitigation measures by using the result of the cost-benefit analysis. It helps the operator in the decision process by providing an interface to compare the mitigation measures in terms of cost-benefit.
- 3) **Data Warehouse** is the central data-interchange and storage facility, being the access point to the information managed across the WISER Framework. Most relationships and data exchanges among the WISER components are based on read and write operations within this Data Warehouse, as described in deliverables D2.3 and D5.1. It has two parts: a document-based storage and a relational-database storage.

The Risk Assessment Engine is the brain of the WISER Framework. This module executes a risk model-based algorithm in order to evaluate the cyber risk of the company. This evaluation is done in near real-time. It is a key component capable of putting the cyber incidents detected in the client infrastructure in the necessary context to evaluate their direct impact in the company business processes. Along with the evaluation of the cyber risk, the Engine suggests mitigation measures, if any. This information is sent to the Decision Support System, where it is represented in a user-friendly way. There are major differences between how it works for CyberWISER-Light, on one side, and CyberWISER-Essential and CyberWISER-Plus, on the other side, with growing complexity. It is implemented as a Python application that, in the most complex case (CyberWISER-Light and CyberWISER-Plus) is broken down into the following modules described below:

- 1) **Indicator Value Generator:** it is in charge of receiving the different inputs required for the generation of the indicators values used in the Risk Assessment Engine and storing those indicator values in the Data Warehouse for its usage by the rest of modules.

- 2) **Triggering Detector:** detects changes in the inputs for the models and, according to certain rules, launches a new execution of the model rules. In such case, a new risk report will be generated.
- 3) **DEXi Model Rules Executor:** it is the Risk Assessment Engine module in charge of the execution the qualitative risk assessment algorithm through the evaluation of the DEXi model file defined for a specific risk model and a specific target in an organization infrastructure.
- 4) **DEXi Model instantiator:** it offers a method *instantiate\_DEXi\_model(data)* which is invoked by the DEXi Model Rules Executor to do the instantiation of a risk model selected for a specific target.
- 5) **R Model Rules Executor:** it is in charge of the execution of the quantitative risk assessment through the use of R scripts.
- 6) **R Model Instantiator:** this module is the counterpart of the DEXi Model Instantiator, and provides a similar functionality, but adapted for the R scripts. The R Model Instantiator is invoked by the R Model Rules Executor to gather the indicator values and the risk model selected for a specific target from the DWH.
- 7) **Aggregator:** aggregates the results at the different levels foreseen in the risk report and sends the output to the Decision Support System.

The Data Warehouse is the central data storage component of the WISER Framework. It stores and provides information crucial to the operation of the Framework including:

- 1) Client users and organizations
- 2) Users' configuration parameters
- 3) Risk models
- 4) Catalogues of risks, mitigation measures and indicators
- 5) Risk reports (results of finished risk assessment procedures)
- 6) Data about active deployed sensors
- 7) Events reported by sensors
- 8) Alarms reported by the Monitoring Engine
- 9) Reports of vulnerabilities found by the vulnerability scanners

The Data Warehouse consists of two database backend services (relational and document-based), their respective definitions of data models, and a common API, providing a single data retrieval point to other components of the WISER Framework.

The relational-database component of the Data Warehouse is intended for storage of data with a rigid schema and a low rate of input. The document-based storage component provides storage for less-structured documents with no relations to other documents and with a high rate of input. The API used for accessing Data Warehouse from other WISER components is based on HTTP REST technology and written using Django REST Framework in Python.

The Data Warehouse REST API ensures security by using HTTPS for all requests and the OAuth2 authentication mechanism. There are three user authorization levels corresponding to three pre-defined user groups:

1. "Wiser Coordinator": A representative of the WISER consortium, who has the highest level of access: he/she can read and write most objects and create new users of both lower-permission roles.
2. "Client Organization Administrator": A manager of the client organization, who can read and write objects that are associated with their organization.
3. "Client Organization Staff": A user belonging to the client organization with the most basic permission levels: he/she can only read objects that are associated with their organization.



---

The Decision Support System is the actor located at the end of the WISER cycle. It is a module in charge of:

- 1) Showing the user the results of the computations carried out by the Risk Assessment Engine. This includes showing the current risk report and a feature to look for older reports by following some kind of filtering criteria. The risk assessment includes the degree of risk (for qualitative reports), the expected losses (for quantitative reports), how the risk is faced by the different components of the target infrastructure and some supplementary information.
- 2) Suggesting mitigation measures envisioned diminishing the risk when applied. WISER proposes an innovative and systematic method to compare, rank, and prioritize them, based on a cost-benefit analysis.
- 3) Offering a tool to learn about the suitability of implementing one mitigation measure from a cost-benefit perspective (cost-benefit analysis). This approach enables the decision-maker to compare all, direct and indirect, positive and negative effects of the proposed decisions. This applies only to CyberWISER-Plus.
- 4) Showing information about the societal impact of the cyber risks assessed, done by collecting inputs via a questionnaire where intangible (but very important) topics related to risk are addressed. This applies to CyberWISER-Essential and CyberWISER-Plus.
- 5) Offering analytical features to help the user to better understand the results produced by the Risk Assessment Engine (risk analytics). CyberWISER-Essential provides a reduced version of this feature, whereas CyberWISER-Plus brings an extended one.

As for the implementation of the Decision Support System, the separation between its presentation layer and its business layer has to be considered. Regarding the presentation layer, the user interacts with the dashboard by means of widgets which are made available thanks to the use of Angular Dashboard Framework (ADF). As for the business layer, the DSS is implemented as a standalone server exposing a REST API that provide means to assess risks in terms of their societal impacts and perform a cost benefit analysis of available security measures.

A first stable and complete version of the Real-Time Assessment Infrastructure is released contextually with the final editing of this deliverable. The remaining months before the end of WP5 activities will be devoted to strengthen and upgrade the infrastructure and to address, in close coordination with WP2 activities, its integration with the rest of the framework. Deliverable D2.4, to be released in M18 (November 2016), will report about this.

The virtues of the technical solution provided will be tested in the context of the three Full-Scale Pilots (WP6) and the validation activities (WP7). It will pave the way towards the rollout to other verticals (also in WP7) and ensure the marketability and impact on society of the project results (WP8), especially after project end.

## **Appendix I      API Endpoints**

---

The following tables contain descriptions of all the API endpoints. Each endpoint corresponds to a data type stored in the Data Warehouse.

Each table shows the data fields with their types and properties, fields which can be used to filter the list results by, access rights for each of the user groups and the API functions supported by the endpoint.

Abbreviations used for the properties of data fields:

- RO – read only
- RE – required
- WO – write only (response does not contain this field)
- OP – optional
- RE, RO – required when creating, but cannot be updated
- UQ – unique

PK – private key  
FK – foreign key

The following endpoints are part of the relational-backed store in the Data Warehouse, also shown on Figure 30 in section 4.

### User management (/users/)

Name	<b>User profile</b>		
Description	Includes properties of a user.		
API endpoint	/users/users/		
Data fields	id	PK	RO, UQ
	username	text (30)	RE, RO, UQ
	password	text	RE, WO
	email	text (254)	RE
	is_active	boolean	OP
	organization	FK	RE, RO
	group	FK	RE
Filter fields	username, organization, email		
Access rights	WISER Coordinator	Read all, write all (except create coordinator)	
	Client Organization Administrator	Read all from org., write staff users to org.	
	Client Organization Staff	Read own profile only	
Allowed functions	List, retrieve, create, update		

Name	<b>Group</b>		
Description	Defines access privileges to users; predefined entries are »WISER Coordinator«, »Client Organization Administrator«, »Client Organization Staff«.		
API endpoint	/users/groups/		
Data fields	id	PK	RO, UQ
	name	text (80)	RO
Access rights	WISER Coordinator	Read all	
	Client Organization Administrator	Read all	
	Client Organization Staff	Read all	
Allowed functions	List, retrieve		

Name	<b>Organization</b>		
Description	Represents a client organization.		
API endpoint	/users/organizations/		
Data fields	id	PK	RO, UQ
	name	text (50)	RE, UQ
Access rights	WISER Coordinator	Read all, write all	
	Client Organization Administrator	Read and write own organization	
	Client Organization Staff	Read own organization	
Allowed functions	List, retrieve, create, update		

### Modelling (/modelling/)

Name	<b>Risk model</b>		
Description	Represents a risk model.		
API endpoint	/modelling/risk_models/		

Data fields	id	PK	RO, UQ
	name	text (50)	RE
	description	text	OP
	dexi_model	text	RE
	r_model	text	RE
	coras_model	text	RE
	relevance_criteria	text	RE
	indicators	array of FKs	
	risks	array of FKs	
Access rights	WISER Coordinator	Read and write all	
	Client Organization Administrator	Read risk models associated with risk patterns, read and write risk models that belong to specific risk models of the user's organization	
	Client Organization Staff	Read risk models associated with risk patterns and risk models that belong to specific risk models of the user's organization	
Allowed functions	List, retrieve, update (creation and deletion only through specific risk model or risk pattern)		

Name	<b>Risk pattern</b>		
Description	Represents a risk pattern (associated to a generic risk model that is available to users of all organizations).		
API endpoint	/modelling/risk_patterns/		
Data fields	id	PK	RO, UQ
	risk_model	Risk model <sup>19</sup>	RE
Access rights	WISER Coordinator	Read and write all	
	Client Organization Administrator	Read all	
	Client Organization Staff	Read all	
Allowed functions	List, retrieve, create, delete <sup>20</sup>		

Name	<b>Specific risk model</b>		
Description	Represents a specific risk model (associated to a custom risk model that is only available to users of the same organization).		
API endpoint	/modelling/specific_risk_models/		
Data fields	id	PK	RO, UQ
	risk_model	Risk model <sup>19</sup>	RE
	organization	FK	RE
Filter fields	organization		
Access rights	WISER Coordinator	Read and write all	
	Client Organization Administrator	Read and write own organization	
	Client Organization Staff	Read own organization	
Allowed functions	List, retrieve, create, delete <sup>20</sup>		

### Risk Assessment Engine (/rae/)

Name	<b>Target</b>
Description	Represents a machine on the client's infrastructure used as a target for risk assessment and vulnerability scanning.

<sup>19</sup> On creation of a risk pattern or a specific risk model, the risk model data is provided as a nested JSON object. Format of the object is the same as described in the risk model specification.

<sup>20</sup> On deletion of a risk pattern or a specific risk model, the associated risk model is deleted as well.

API endpoint	/rae/targets/		
Data fields	id	PK	RO, UQ
	ip_address	text (39)	RE
	port	integer	OP
	short_description	text	RE
	complete_description	text	RE
	deleted <sup>21</sup>	boolean	OP
	organization	FK	RE
	availability	int (range 0-10)	RE
	confidentiality	int (range 0-10)	RE
integrity	int (range 0-10)	RE	
Access rights	WISER Coordinator	Read and write all	
	Client Organization Administrator	Read and write own organization	
	Client Organization Staff	Read own organization	
Allowed functions	List, retrieve, create, update, delete <sup>21</sup>		

Name	<b>Indicator</b>		
Description	Represents an indicator that can be used for risk assessment.		
API endpoint	/rae/indicators/		
Data fields	id	PK	RO, UQ
	question	text	RE
	data_type	text (30)	RE
	motivation	text	RE
	indicator_type	FK	RE
	means	text	RE
	targets	array of FKs	OP
rule	text	OP	
Access rights	WISER Coordinator	Read and write	
	Client Organization Administrator	Read	
	Client Organization Staff	Read	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Indicator type</b>		
Description	Represents an indicator type.		
API endpoint	/rae/indicator_types/		
Data fields	id	PK	RO, UQ
	name	text	RE
Access rights	WISER Coordinator	Read and write	
	Client Organization Administrator	Read	
	Client Organization Staff	Read	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Indicator value</b>		
Description	Represents a concrete value of an indicator.		
API endpoint	/rae/indicator_values/		
Data fields	id	PK	RO, UQ
	target <sup>22</sup>	FK	RE, RO

<sup>21</sup> An HTTP DELETE query to a target marks it as deleted. Deleted targets are not returned with normal queries, but can be listed by a WISER coordinator with a query containing a ?deleted=True parameter. The »deleted« field is not returned by the API.

	indicator <sup>22</sup>	FK	RE, RO
	value	text (30)	RE
	timestamp <sup>23</sup>	datetime	RO
Filter fields	target, indicator, value		
Access rights	WISER Coordinator	Read and write all	
	Client Organization Administrator	Read and write indicator values for targets of own organization	
	Client Organization Staff	Read and write indicator values for targets of own organization	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Risk report</b>		
Description	Represents a result of the risk assessment procedure.		
API endpoint	/rae/risk_reports/		
Data fields	id	PK	RO, UQ
	qualitative_assessment	text	OP
	quantitative_assessment	text	OP
	organization	FK	RE
	mitigation_measures	array of FKs	OP
	timestamp <sup>23</sup>	datetime	RO
Filter fields	organization, risk_report_risks__isnull <sup>24</sup>		
Access rights	WISER Coordinator	Read and write all	
	Client Organization Administrator	Read and write own organization	
	Client Organization Staff	Read and write own organization	
Allowed functions	List, retrieve, create, delete		

Name	<b>Section</b>		
Description	Represents a section of a risk report.		
API endpoint	/rae/sections/		
Data fields	id	PK	RO, UQ
	name	text (50)	RE
Access rights	WISER Coordinator	Read and write	
	Client Organization Administrator	Read	
	Client Organization Staff	Read	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Risk</b>		
Description	Represents a risk whose value can be assessed.		
API endpoint	/rae/risks/		
Data fields	id	PK	RO, UQ
	short_description	text	RE
	detailed_description	text	RE
	sections	array of FKs	RE
	mitigation_measures	array of FKs	OP
Filter fields	sections, mitigation_measures		

<sup>22</sup> Target and indicator pair must be unique (a target can only have one indicator value for a specific indicator).

<sup>23</sup> This field is automatically set to the current time – this cannot be overridden.

<sup>24</sup> The response will return only risk reports without any associated risk\_report objects, or only risk reports with at least one associated risk\_report object, respectively for parameter values »True« or »False«.

Access rights	WISER Coordinator	Read and write
	Client Organization Administrator	Read
	Client Organization Staff	Read
Allowed functions	List, retrieve, create, update, delete	

Name	<b>Mitigation measure</b>		
Description	Represents a mitigation measure that can be included in a risk report.		
API endpoint	/rae/mitigation_measures/		
Data fields	id	PK	RO, UQ
	short_description	text	RE
	detailed_description	text	RE
	extended_description	text	OP
Filter fields	risks, risks_risk_per_target <sup>25</sup> , risks_risk_per_target_report <sup>26</sup> , risks_risk_report_risks_report <sup>27</sup> , risk_report <sup>28</sup>		
Access rights	WISER Coordinator	Read and write	
	Client Organization Administrator	Read	
	Client Organization Staff	Read	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Risk report section</b>		
Description	Represents a result of a risk assessment for a report section.		
API endpoint	/rae/risk_reports_section/		
Data fields	id	PK	RO, UQ
	report	FK	RE
	section	FK	RE
	qualitative_assessment	text	RE
	quantitative_assessment	text	RE
Filter fields	report, section		
Access rights	WISER Coordinator	Read and write all	
	Client Organization Administrator	Read and write for reports of own organization	
	Client Organization Staff	Read and write for reports of own organization	
Allowed functions	List, retrieve, create, delete		

Name	<b>Risk report risk</b>		
Description	Represents an assessment of a specific risk in a section of a specific report.		
API endpoint	/rae/risk_reports_risk/		
Data fields	id	PK	RO, UQ
	report	FK	RE
	section	FK	RE
	risk	FK	RE
	qualitative_assessment	text	RE
	quantitative_assessment	text	RE
Filter fields	report, section, risk		
Access rights	WISER Coordinator	Read and write all	
	Client Organization	Read and write for reports of own organization	

<sup>25</sup> Filter by risk in the specified »risk per target« object.

<sup>26</sup> Filter by risks included in the specified risk report through »risk per target« objects.

<sup>27</sup> Filter by risks included in the specified risk report through »risk report risk« objects.

<sup>28</sup> Filter by risks included in the specified risk report through either »risk per target« or »risk report risk« objects.

	Administrator	
	Client Organization Staff	Read and write for reports of own organization
Allowed functions	List, retrieve, create, delete	

Name	<b>Risk per target</b>		
Description	Represents an assessment of a specific risk in a specific target in the client's infrastructure in a specific risk report.		
API endpoint	/rae/risk_per_target/		
Data fields	id	PK	RO, UQ
	report	FK	RE
	risk_model	FK	RE
	risk	FK	RE
	target <sup>29</sup>	FK	RE
	qualitative_assessment	text	RE
	quantitative_assessment	text	RE
Filter fields	Report, risk_model, risk, target		
Access rights	WISER Coordinator	Read and write all	
	Client Organization Administrator	Read and write for reports of own organization	
	Client Organization Staff	Read and write for reports of own organization	
Allowed functions	List, retrieve, create, delete		

### Decision Support System (/dss/)

Name	<b>CBA default structure</b>		
Description	Represents a default cost-benefit structure.		
API endpoint	/dss/cba_default_structures/		
Data fields	id	PK	RO, UQ
	cba_default_structure	text	RE
Access rights	WISER Coordinator	Read and write	
	Client Organization Administrator	Read	
	Client Organization Staff	Read	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Quality criteria</b>		
Description	Represents quality criteria which can be considered to evaluate the societal impact of risk.		
API endpoint	/dss/quality_criteria/		
Data fields	id	PK	RO, UQ
	category	text	RE
	short_name	text (50)	RE
	description	text	RE
Access rights	WISER Coordinator	Read and write	
	Client Organization Administrator	Read	
	Client Organization Staff	Read	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Quality criteria analysis</b>		
Description	Represents a quality criteria analysis performed on a specific risk report for a specific risk.		

<sup>29</sup> Target has to belong to the same organization as the risk report.

API endpoint	/dss/quality_criteria_analyses/		
Data fields	id	PK	RO, UQ
	report	FK	RE
	risk	FK	RE
	quality_criteria	FK	RE
	value	text (30)	RE
Filter fields	report, risk, quality_criteria		
Access rights	WISER Coordinator	Read and write all	
	Client Organization Administrator	Read and write for reports of own organization	
	Client Organization Staff	Read and write for reports of own organization	
Allowed functions	List, retrieve, create, delete		

Name	<b>Cost-benefit analysis</b>		
Description	Represents a cost-benefit analysis made for each of the considered mitigation measures associated to a risk assessment report.		
API endpoint	/dss/cost_benefit_analyses/		
Data fields	id	PK	RO, UQ
	report	FK	RE
	mitigation_measure	FK	RE
	cba_structure	text	RE
	cba_values	text	RE
Filter fields	report, mitigation_measure		
Access rights	WISER Coordinator	Read and write all	
	Client Organization Administrator	Read and write for reports of own organization	
	Client Organization Staff	Read and write for reports of own organization	
Allowed functions	List, retrieve, create, delete		

### Configuration (/config/)

Name	<b>Management of alerting</b>		
Description	User alert configuration to be used to show them in the dashboard.		
API endpoint	/config/alerts/		
Data fields	id	PK	RO, UQ
	user	FK	RO, UQ
	risk_threshold	int	RE
	dst_ip (range of ips)	text (100)	OP
	src_ip (range of ips)	text (100)	OP
	from_date	date	RE
	to_date	date	RE
	alert_source (range of plugin_id)	text (100)	OP
	alert_type (range of plugin_sid)	text (100)	OP
Access rights	WISER Coordinator	Read all, write all	
	Client Organization Administrator	Read and write own alert configuration	
	Client Organization Staff	Read and write own alert configuration	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>WISER agents</b>		
Description	Represents the WISER Agents deployed in the organization.		
API endpoint	/config/wiser_agents/		
Data fields	id	PK	RO, UQ
	organization	FK	RE, RO



	ip	text (39)	RE
	name	text (254)	RE
	is_active	boolean	RE
	description	text (200)	OP
Access rights	WISER Coordinator	Read all, write all	
	Client Organization Administrator	Read and write for own organization	
	Client Organization Staff	Read for own organization	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Monitoring engine network</b>		
Description	Represents the networks of the organization monitored by the Monitoring Engine.		
API endpoint	/config/monitoring_networks/		
Data fields	id	PK	RO, UQ
	organization	FK	RE, RO
	cidrs	text (100)	RE
	name	text (254)	RE
	description	text (200)	OP
Access rights	WISER Coordinator	Read all, write all	
	Client Organization Administrator	Read and write for own organization	
	Client Organization Staff	Read for own organization	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Monitoring engine policy</b>		
Description	Represents the policies of the organization with respect to allowed traffic (e.g. eDonkey P2P, BitTorrent, Skype...) to be taken into account with respect to correlation rules to be applied in the Monitoring Engine.		
API endpoint	/config/monitoring_policies/		
Data fields	id	PK	RO, UQ
	organization	FK	RE, RO
	policy	text (50)	RE
	is_active	boolean	RE
Access rights	WISER Coordinator	Read all, write all	
	Client Organization Administrator	Read and write for own organization	
	Client Organization Staff	Read for own organization	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Monitoring engine application</b>		
Description	Represents available services/applications in the organization (e.g. MS SQL Server, Cisco routers, Apache Server...) to be taken into account with respect to correlation rules to be applied in the Monitoring Engine.		
API endpoint	/config/monitoring_applications/		
Data fields	id	PK	RO, UQ
	organization	FK	RE, RO
	application	text	RE
	is_active	boolean	RE
Access rights	WISER Coordinator	Read all, write all	
	Client Organization Administrator	Read and write for own organization	
	Client Organization Staff	Read for own organization	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Company profile</b>		
Description	Represents answers to the company profile questionnaire.		
API endpoint	/config/company_profiles/		
Data fields	id	PK	RO, UQ
	organization	FK	RE, RO, UQ
	data	text	RE
Access rights	WISER Coordinator	Read all, write all	
	Client Organization Administrator	Read and write for own organization	
	Client Organization Staff	Read for own organization	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Selected risk model</b>		
Description	Indicates the selected risk models used for risk assessment for a specific organization.		
API endpoint	/config/selected_risk_models/		
Data fields	id	PK	RO, UQ
	organization	FK	RE, RO, UQ
	risk_models	array of FKs	
Filter fields	organization, risk_models		
Access rights	WISER Coordinator	Read all, write all	
	Client Organization Administrator	Read and write for own organization	
	Client Organization Staff	Read for own organization	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Question</b>		
Description	Represents a question in the company profile questionnaire.		
API endpoint	/config/question/		
Data fields	id	PK	RO, UQ
	data	text	RE
	question_number	int	RE, UQ
Access rights	WISER Coordinator	Read all, write all	
	Client Organization Administrator	Read all	
	Client Organization Staff	Read all	
Allowed functions	List, retrieve, create, update, delete		

### Monitoring (/monitoring/)

Name	<b>Sensor type</b>		
Description	A catalogue of sensor types.		
API endpoint	/monitoring/sensor_types/		
Data fields	id	PK	RO, UQ
	name	text (30)	RE, UQ
	short_description	text (50)	RE
Access rights	WISER Coordinator	Read all, write all	
	Client Organization Administrator	Read all	
	Client Organization Staff	Read all	
Allowed functions	List, retrieve, create, update, delete		

Name	<b>Sensor</b>
------	---------------

Description	Represents an instance of a sensor, deployed at the organization.		
API endpoint	/monitoring/sensors/		
Data fields	id	PK	RO, UQ
	sensor_type <sup>30</sup>	FK	RE
	organization <sup>30</sup>	FK	RE
	ip_address <sup>30</sup>	text (39)	RE
	sensor_timestamp	datetime	RE
	sensor_status	text (60)	RE
	last_seen	datetime	RO
	first_seen	datetime	RO
	extras	text	RE
Access rights	WISER Coordinator	Read all, write all	
	Client Organization Administrator	Read for own organization	
	Client Organization Staff	Read for own organization	
Allowed functions	List, retrieve, create <sup>30</sup> , update <sup>30</sup>		

### Documents (/documents/)

Below we present the API endpoints for data types, backed by the document-based store of the Data Warehouse.

Name	<b>Event</b>		
Description	Indicates an event generated by a monitoring sensor, sent by the WISER Agent.		
API endpoint	/documents/events/		
Data fields	id	PK	
	date	datetime	
	device	ip	
	dst_ip	ip	
	dst_port	int	
	event_id	string	
	fdate	datetime	
	interface	string	
	log	string	
	organization	string	
	plugin_id	int	
	plugin_sid	int	
	protocol	string	
	src_ip	ip	
	src_port	int	
	type	string	
	tzone	float	
	username	string	
	userdata1	string	
	userdata2	string	
	userdata3	string	
	userdata4	string	
Access rights	WISER Coordinator	Read all	
	Client Organization	Read documents of their own organization	

<sup>30</sup> Fields »sensor\_type«, »organization«, and »ip\_address« form a unique set among sensor objects. If another object with the same values in all these fields is POSTed, an existing sensor is updated and returned with status code 200. If a new object was created, the status code is 201.

	Administrator	
	Client Organization Staff	Read documents of their own organization
Allowed functions	List, retrieve	

Name	<b>Alarm</b>	
Description	Indicates an alarm generated by the WISER Monitoring Engine.	
API endpoint	/documents/alarms/	
Data fields	id	PK
	BACKLOG_ID	string
	DATE	datetime
	DST_IP	ip
	DST_PORT	int
	EVENT_ID	string
	PLUGIN_ID	int
	PLUGIN_NAME	string
	PLUGIN_sID	int
	PRIORITY	int
	PROTOCOL	int
	RELATED_EVENTS	string
	RELIABILITY	int
	RISK	int
	SID_NAME	string
	SRC_IP	ip
	SRC_PORT	int
	USERDATA2	string
	USERNAME	string
Access rights	WISER Coordinator	Read all
	Client Organization Administrator	Read documents of their own organization
	Client Organization Staff	Read documents of their own organization
Allowed functions	List, retrieve	

Name	<b>Vulnerability scan report</b>	
Description	Represents a report generated by the vulnerability scanner.	
API endpoint	/documents/vuln_scan_reports/	
Data fields	id	PK
	date	datetime
	device	ip
	dst_ip	ip
	event_id	string
	fdate	datetime
	interface	string
	log <sup>31</sup>	string
	organization	string
	plugin_id	int
	plugin_sid	int
	src_ip	ip
	type	string
	report <sup>31</sup>	string
	scanner_log <sup>31</sup>	string
	target	string

<sup>31</sup> Values in these fields can be very long and are not returned when listing vulnerability scan reports. All the fields are returned when retrieving a single report instance.

	target_id	int
	task_status	string
Access rights	WISER Coordinator	Read all
	Client Organization Administrator	Read documents of their own organization
	Client Organization Staff	Read documents of their own organization
Allowed functions	List, retrieve	