

<b>Project Title</b>	Civil Cyber Range Platform for a novel approach to cybersecurity threats simulation and professional training
<b>Project Acronym</b>	CYBERWISER.EU
<b>Project Number</b>	786668
<b>Type of instrument</b>	Innovation Action
<b>Topic</b>	DS-07-2017 Cybersecurity PPP: Addressing Advanced Cyber Security Threats and Threat Actors
<b>Starting date of Project</b>	01/09/2018
<b>Duration of the project</b>	30
<b>Website</b>	<a href="http://www.cyberwiser.eu">www.cyberwiser.eu</a>

## D2.5 Platform Design, Final Version

<b>Work Package</b>	WP2 Requirements, Design and Building Blocks
<b>Lead author</b>	Antonio Álvarez (ATOS)
<b>Contributors</b>	Javier Ramírez (ATOS), Tobias Groothuyse (RHEA), Matteo Merialdo (RHEA), Manca Bizjak, Anže Žitnik (XLAB), Liliana Ribeiro (EDP), Niccolò Zazzeri (TRUST-IT), Alessandro Petrocelli (TRUST-IT), Massimo Berutto (TRUST-IT), Gencer Erdogan (SINTEF), Dario Varano (UNIFI)
<b>Peer reviewers</b>	Niccolò Zazzeri (TRUST-IT), Alessandro Petrocelli (TRUST-IT), Liliana Ribeiro (EDP)
<b>Version</b>	V1.0
<b>Due Date</b>	31/08/2019
<b>Submission Date</b>	31/08/2019

Dissemination Level:

X	PU: Public
	CO: Confidential, only for members of the consortium (including the Commission)
	EU-RES. Classified Information: RESTREINT UE (Commission Decision 2005/444/EC)
	EU-CON. Classified Information: CONFIDENTIEL UE (Commission Decision 2005/444/EC)
	EU-SEC. Classified Information: SECRET UE (Commission Decision 2005/444/EC)



The work described in this document has been conducted within the CYBERWISER project. This project has received funding by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 786668.

## Version History

Revision	Date	Editor	Comments
0.0	14/03/2019	Antonio Álvarez (ATOS)	Initial table of contents
0.1	15/05/2019	Antonio Álvarez (ATOS), Tobias Groothuyse (RHEA), Matteo Merialdo (RHEA)	Input to section 4.2, 6.1 and 7
0.2	22/05/2019	Antonio Álvarez (ATOS), Tobias Groothuyse (RHEA)	Refinement of section 2.3, 4.2. New input to section 6.1
0.3	07/06/2019	Manca Bizjak, Anže Žitnik (XLAB), Liliana Ribeiro (EDP)	Input to section 1.4 and 4.4. Refinement in section 7
0.4	07/06/2019	Antonio Álvarez (ATOS), Javier Ramírez (ATOS)	Document layout
0.5	13/06/2019	Niccolò Zazzeri (Trust-IT)	Input in section 6
0.6	18/06/2019	Antonio Álvarez (ATOS)	Refinements in section 6
0.7	18/06/2019	Antonio Álvarez (ATOS), Javier Ramírez (ATOS)	Inputs to sections 3.2.16 and 5.5. Document layout
0.8	19/06/2019	Antonio Álvarez (ATOS)	Input to sections 3.2.7, 3.2.8, 5.10
0.9	25/06/2019	Tobias Groothuyse (RHEA)	Input to sections 3.2.3, 3.2.4, 3.2.5, 3.2.6, 4.4.7, 5.3, 5.4, 5.6. Input to table of interfaces in section 3.2. Input to table of technologies in section 4.4. Refinements in section 6.1.
0.10	26/06/2019	Antonio Álvarez (ATOS)	Input to section 1.1, 1.2, 1.3
0.11	26/06/2019	Manca Bizjak (XLAB), Anže Žitnik (XLAB)	Input to sections 3.2.10-3.2.12. Updated table in 4.4. Refined sections 4.4.1-4.4.4. Updated glossary of acronyms. Input to section 5.3.1, new section 5.8, input to 5.9, 5.12, 6.1. Refinement of section 4.2
0.12	26/06/2019	Antonio Álvarez (ATOS)	Input to sections 1.4, 3.2.9, 3.2.14, 3.2.15. Minor modification to the title of section 3.2.1
0.13	27/06/2019	Niccolò Zazzeri (TRUST-IT), Alessandro Petrocelli (TRUST-IT)	Inputs to sections 4.4.12 and 4.4.13. Inputs and refinements in section 6
0.14	27/06/2019	Antonio Álvarez (ATOS), Javier Ramírez (ATOS)	Input to sections 5.7, 5.10. Input to table in section 4.4. Input to section 4.4.8.
0.15	27/06/2019	Antonio Álvarez (ATOS)	Input to sections 3.2.13, 4.4.10, 5.13
0.16	01/07/2019	Niccolò Zazzeri (TRUST-IT), Alessandro Petrocelli (TRUST-IT), Massimo Berutto (TRUST-IT)	Refinements to sections 4.4 4.4.12 and 4.4.13
0.17	09/07/2019	Antonio Álvarez (ATOS)	Refinements in sections 3.2.7, 3.2.15, 3.2.16, 4.2. Inputs to sections 4.1, 4.1.1, 4.1.2, 4.1.3, 4.4.9. Created subsections 4.3.1 and 4.3.2 for 4.3
0.18	09/07/2019	Manca Bizjak (XLAB), Anže Žitnik (XLAB)	Input and refinements to section 6.1, refinements to sections 4.4.1, 4.4.2, 4.4.3, 5.8, 5.9 and 5.12
0.19	09/07/2019	Antonio Álvarez (ATOS), Gencer Erdogan (SINTEF)	Refinement in sections 1.3, 4.4, 5.5. Input to section 4.4.11, 5.7.1
0.20	10/07/2019	Antonio Álvarez (ATOS), Tobias Groothuyse (RHEA)	Document layout. Input to section 4.3.1, 4.4, 4.4.14. Refinement of section 5.13
0.21	11/07/2019	Niccolò Zazzeri (TRUST-IT), Alessandro Petrocelli (TRUST-IT), Massimo Berutto (TRUST-IT)	Input to section 5.1 and 5.2
0.22	11/07/2019	Gencer Erdogan (SINTEF)	Updated section 5.7.1
0.23	12/07/2019	Antonio Álvarez (ATOS)	Input to section 6.1
0.24	12/07/2019	Antonio Álvarez (ATOS)	Input to section 3.2

Revision	Date	Editor	Comments
0.25	12/07/2019	Anže Žitnik (XLAB)	Input to section 3.2, data model diagrams updated
0.26	15/07/2019	Antonio Álvarez (ATOS)	Input to section 6.1
0.27	15/07/2019	Antonio Álvarez (ATOS)	Input to section 6.1
0.28	17/07/2019	Niccolò Zazzeri (TRUST-IT)	Input to section 6.2
0.29	17/07/2019	Antonio Álvarez (ATOS)	Document layout
0.30	18/07/2019	Anže Žitnik (XLAB)	Resolving comments, formatting in section 4.4
0.31	18/07/2019	Niccolò Zazzeri (TRUST-IT)	Input to section 6.2
0.32	19/07/2019	Antonio Álvarez (ATOS), Tobias Groothuyse (RHEA)	Input to section 6.1. Refinement of sections 2.3, 4.4.7. Document layout
0.33	19/07/2019	Antonio Álvarez (ATOS)	Document layout
0.34	22/07/2019	Tobias Groothuyse (RHEA)	Refinement of section 1.1, 1.3, 3.2. Input to section 2.1.1, 2.1.2, 2.2, 4.3, 6.1
0.35	23/07/2019	Niccolò Zazzeri (TRUST-IT), Alessandro Petrocelli (TRUST-IT), Massimo Berutto (TRUST-IT)	Refinement of section 6.1 and 6.2, input on section 1.3
0.36	24/07/2019	Anže Žitnik (XLAB)	Refinements in sections 5.9, 6.1 and 6.2
0.37	26/07/2019	Dario Varano (UNIPi)	Refinements in section 6.2
0.38	29/07/2019	Javier Ramírez (ATOS)	Contributions merging
0.39	29/07/2019	Gencer Erdogan (SINTEF)	Commented on and provided input to Section 4.
0.40	30/07/2019	Dario Varano (UNIPi)	Updated sequence diagrams and corrected some typos for section 6.2.
0.41	07/08/2019	Javier Ramírez (ATOS), Antonio Álvarez (ATOS)	General updates to the document
0.42	07/08/2019	Antonio Álvarez (ATOS)	Document layout
0.43	08/08/2019	Niccolò Zazzeri (TRUST-IT), Alessandro Petrocelli (TRUST-IT), Massimo Berutto (TRUST-IT)	Updated figure in section 4.3
0.44	09/08/2019	Anže Žitnik (XLAB)	Updated interfaces table in Section 3.2.16 and data model diagrams in Section 4.3.
0.45	09/08/2019	Antonio Álvarez (ATOS)	Adding keywords, executive summary, refinement and consolidation of section 1.3.
0.46	12/08/2019	Antonio Álvarez (ATOS)	Introduction in section 6. Layout in sections 7.1, 7.3, 7.4
0.47	12/08/2019	Antonio Álvarez (ATOS), Javier Ramírez (ATOS)	Refinement in section 3.2, 4.3.2, 5.11. Writing section 9.
0.48	13/08/2019	Antonio Álvarez (ATOS)	Refinement in section 6.1
0.49	13/08/2019	Antonio Álvarez (ATOS), Tobias Groothuyse (RHEA)	New input to section 6.2. Refinement of sections 2.3, 4.3..
0.50	13/08/2019	Antonio Álvarez (ATOS), Tobias Groothuyse (RHEA)	Refinement of section 8
0.51	13/08/2019	Dario Varano (UNIPi)	Update to diagrams of section 6.2 and general review of the section
0.52	14/08/2019	Antonio Álvarez (ATOS)	Refinement in sections 2.3, 6.1. Input and refinements in section 6.2
0.53	18/08/2019	Antonio Álvarez (ATOS)	Refinement in section 4.3. Some issues solved along the document
0.54	18/08/2019	Antonio Álvarez (ATOS)	Document layout
0.55	18/08/2019	Antonio Álvarez (ATOS)	Version ready for QA
0.56	22/08/2019	Antonio Álvarez (ATOS), Liliana Ribeiro	QA from EDP and answers

Revision	Date	Editor	Comments
0.57	23/08/2019	Javier Ramírez (ATOS), Manca Bizjak (XLAB), Tobias Groothuyse (RHEA), Liliana Ribeiro (EDP), Alessandro Petrocelli (TRUST-IT), Massimo Berutto (TRUST-IT), Niccolò Zazzeri (TRUST-IT)	QA issues and answers
0.58	26/08/2019	Javier Ramírez (ATOS), Alessandro Petrocelli (TRUST-IT), Massimo Berutto (TRUST-IT), Niccolò Zazzeri (TRUST-IT)	QA issues and answers
0.59	30/08/2019	María Teresa García (ATOS)	Final Quality Check
1.0	30/08/2019	Antonio Álvarez (ATOS), María Teresa García (ATOS)	Quality Assessment comments addressed. Document ready for submission

## List of Contributors

The list of contributors to this deliverable are presented in the following table:

Section	Author(s)
Executive Summary	Antonio Álvarez (ATOS)
1. Introduction	Antonio Álvarez (ATOS), Tobias Groothuyse (RHEA)
2. Methodology	Antonio Álvarez (ATOS), Tobias Groothuyse (RHEA)
3. CYBERWISER.eu top-down final design	Antonio Álvarez (ATOS), Tobias Groothuyse (RHEA), Manca Bizjak (XLAB), Anže Žitnik (XLAB)
4. Other design perspectives	Antonio Álvarez (ATOS), Javier Ramírez (ATOS), Tobias Groothuyse (RHEA), Manca Bizjak, Anže Žitnik (XLAB), Niccolò Zazzeri (TRUST-IT), Alessandro Petrocelli (TRUST-IT), Massimo Berutto (TRUST-IT), Gencer Erdogan (SINTEF)
5. Detail of the building components	Antonio Álvarez (ATOS), Javier Ramírez (ATOS), Tobias Groothuyse (RHEA), Manca Bizjak (XLAB), Anže Žitnik (XLAB), Gencer Erdogan (SINTEF), Niccolò Zazzeri (TRUST-IT), Alessandro Petrocelli (TRUST-IT), Massimo Berutto (TRUST-IT)
6. Using the platform	Antonio Álvarez (ATOS), Tobias Groothuyse (RHEA), Niccolò Zazzeri (TRUST-IT), Alessandro Petrocelli (TRUST-IT), Manca Bizjak (XLAB), Anže Žitnik (XLAB), Dario Varano (UNIP)
7. Impact of the business requirements	Matteo Merialdo (RHEA), Tobias Groothuyse (RHEA), Antonio Álvarez (ATOS), Liliana Ribeiro (EDP)
8. Final requirements traceability	Matteo Merialdo (RHEA), Tobias Groothuyse (RHEA)
9. Conclusions	Antonio Álvarez (ATOS)

## Keywords

Design, requirement traceability, component, architecture, tier, data model, data flow, technology stack, use cases, deployment, platform, scenario.

## Disclaimer

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

## Table of Contents

1. INTRODUCTION .....	14
1.1 Purpose and scope of the document.....	14
1.2 Structure of the document .....	14
1.3 Relation to other work in the project .....	15
1.4 Glossary of Acronyms.....	17
2. METHODOLOGY .....	19
2.1 System architecture / design engineering process.....	19
2.1.1 Methodology for defining the system architecture.....	19
2.1.2 Conclusions .....	19
2.2 Modelling languages.....	20
2.3 User flows and use cases.....	20
2.4 Traceability between requirements and design.....	21
3. CYBERWISER.EU TOP-DOWN FINAL DESIGN .....	22
3.1 Level 1 component diagram .....	22
3.2 Level 2 component diagram .....	23
3.2.1 Message Broker (MB).....	24
3.2.2 Centralized Logging Component (CLC) .....	24
3.2.3 Digital Library (DL).....	24
3.2.4 Training Manager (TM).....	24
3.2.5 Simulated Infrastructure Manager (SIM) .....	25
3.2.6 Infrastructure as a Service (IaaS).....	25
3.2.7 Anomaly Detection Reasoner (ADR).....	25
3.2.8 Anomaly Detection Reasoner Agent (ADR Agent).....	25
3.2.9 Monitoring Sensors.....	25
3.2.10 Event-based Service Orchestrator (EBSO) .....	26
3.2.11 Vulnerability Assessment Tools (VAT) .....	26
3.2.12 Attack Simulator (AS) .....	26
3.2.13 Countermeasures Simulator (CS) .....	26
3.2.14 Economic Risk Models (ERM).....	26
3.2.15 Economic Risk Evaluator (ERE).....	26
3.2.16 Performance Evaluator (PE).....	26
4. OTHER DESIGN PERSPECTIVES.....	29
4.1 Software components structure perspective .....	29
4.1.1 Presentation tier.....	30
4.1.2 Business tier .....	30
4.1.3 Back tier.....	30
4.2 Software deployment perspective .....	31
4.3 Data view .....	32
4.3.1 Data model .....	32
4.3.2 Data flows .....	40
4.4 Technology view .....	42
4.4.1 Event-based Service Orchestrator (xRuntime).....	43
4.4.2 Attack Simulator.....	43
4.4.3 Vulnerability Assessment Tools.....	44

4.4.4 Pluggable deployment manager (xOpera) .....	45
4.4.5 Message Broker.....	46
4.4.6 Centralized Logging Component .....	46
4.4.7 Digital Library, Training Manager, Simulated Infrastructure Manager.....	46
4.4.8 Anomaly Detection Reasoner .....	46
4.4.9 Monitoring Sensors .....	47
4.4.10 Economic Risk Evaluator .....	47
4.4.11 Performance Evaluator .....	48
4.4.12 Cross-Learning Facilities .....	48
4.4.13 Web Portal .....	49
4.4.14 Countermeasures Simulator .....	49
5. DETAIL OF THE BUILDING COMPONENTS .....	50
5.1 Web Portal .....	50
5.2 Cross-Learning Facilities .....	50
5.3 Simulated Infrastructure Manager .....	50
5.3.1 Pluggable Deployment Manager (xOpera) .....	52
5.4 Training Manager.....	53
5.4.1 Scenario Designer .....	53
5.5 Performance Evaluator .....	55
5.6 Digital Library.....	58
5.7 Economic Risk Evaluator.....	59
5.7.1 Models .....	62
5.8 Event-Based Service Orchestrator (EBSO) .....	63
5.9 Vulnerability Assessment Tools.....	65
5.10 Monitoring Sensors.....	68
5.11 Anomaly Detection Reasoner .....	70
5.12 Attack Simulator.....	73
5.13 Countermeasures Simulator .....	75
6. USING THE PLATFORM.....	81
6.1 User flows through the user interface.....	81
6.2 Use cases .....	124
6.2.1 Website.....	124
6.2.2 Login page .....	125
6.2.3 Private area .....	126
6.2.4 Workspace area.....	126
6.2.5 Moodle Dashboard – Trainee interaction .....	128
6.2.6 Moodle dashboard – Trainer interaction.....	129
6.2.7 Creating and using new Assets .....	130
6.2.8 Designing a new Scenario .....	131
6.2.9 Using a designed Scenario.....	133
6.2.10 Vulnerability Assessment Tools – Login Page .....	135
6.2.11 Vulnerability Assessment Tools – Main Page .....	136
6.2.12 Vulnerability Assessment Tools – Scan Details Page.....	137
6.2.13 Vulnerability Assessment Tools – Vulnerability Scan Creation .....	138
6.2.14 Vulnerability Assessment Tools – Vulnerability Scan Modification .....	140
6.2.15 Attack Simulator – Login Page .....	141
6.2.16 Attack Simulator – Main Page .....	143



6.2.17 Attack Simulator – Attack Details Page .....	144
6.2.18 Attack Simulator – Attack Creation .....	145
6.2.19 Attack Simulator – Attack Modification .....	147
6.2.20 Centralized Logging Component Dashboard .....	148
6.2.21 Economic Risk Evaluator – getting risk information .....	149
6.2.22 Economic Risk Evaluator – consulting target configuration .....	150
6.2.23 Economic Risk Evaluator – consulting models selected .....	151
6.2.24 Anomaly Detection Reasoner – monitoring the simulated infrastructure .....	152
6.2.25 Countermeasures Simulator – applying mitigation .....	153
7 IMPACT OF THE BUSINESS REQUIREMENTS ON THE DESIGN .....	155
7.1 CYBERWISER.eu versions and components .....	155
7.2 Primer Version Deployment Diagram .....	158
7.3 Basic Version deployment diagram .....	159
7.4 Intermediate Version Deployment Diagram.....	161
8 FINAL REQUIREMENTS TRACEABILITY .....	163
9 CONCLUSIONS .....	183

## List of figures

Figure 1. Level 1 component diagram .....	22
Figure 2. Level 2 component diagram .....	23
Figure 3. 3-tier-software structure perspective .....	29
Figure 4. Deployment diagram .....	31
Figure 5. Package level ERD .....	33
Figure 6. Entity level ERD – Course package .....	34
Figure 7. Entity level ERD – Scenario package .....	35
Figure 8. Entity level ERD – Actions package .....	35
Figure 9. Entity-level ERD – Monitoring package.....	36
Figure 10. Attribute ERD – Course package .....	37
Figure 11. Attribute ERD – Scenario package .....	38
Figure 12. Attribute ERD –Actions package .....	39
Figure 13. Attribute ERD – Monitoring package.....	40
Figure 14. Data Flow Diagram.....	41
Figure 15. Simulated Infrastructure Manager components .....	51
Figure 16. Pluggable Deployment Manager component diagram .....	52
Figure 17. Conceptual design of the TOSCA Translator.....	53
Figure 18. Training Manager components.....	54
Figure 19. Internal composition of the Performance Evaluator and interfaces from/to the outside .....	55
Figure 20. Internal functional design of the Performance Evaluator .....	57
Figure 21. Digital Library Component Diagram .....	58
Figure 22. Economic risk evaluator internal functional design .....	60
Figure 23. Economic Risk Evaluator layer design .....	61
Figure 24. Outline of overall method for cyber risk modelling and its relationship to the Economic Risk Evaluator .....	63
Figure 25. Event-Based Service Orchestrator component diagram.....	64
Figure 26. Logical view of vulnerability assessment tools .....	66
Figure 27. Vulnerability Assessment Tools component diagram .....	67
Figure 28. Monitoring sensors and Cyber Agent .....	69
Figure 29. Anomaly Detection Reasoner diagram .....	71
Figure 30. Anomaly Detection Reasoner, layer design .....	72



Figure 31. Logical view of the Attack Simulator .....	74
Figure 32. Attack simulator component diagram .....	75
Figure 33. Concept of Countermeasures Simulator .....	76
Figure 34. Inter-relation of concepts for mitigation identification .....	77
Figure 35. Design to apply the Countermeasures Simulator concept .....	79
Figure 36. Internal functional design of the Countermeasures Simulator .....	80
Figure 37. Relations among the user interfaces .....	123
Figure 38. Website sequence diagram .....	124
Figure 39. Login page sequence diagram .....	125
Figure 40. Private area sequence diagram .....	126
Figure 41. Workspace area sequence diagram .....	127
Figure 42. Moodle dashboard – trainee interaction sequence diagram .....	128
Figure 43. Moodle dashboard – Trainer interaction sequence diagram .....	129
Figure 44. Creating and assigning an Asset .....	130
Figure 45. Designing a new scenario .....	132
Figure 46. Using a designed scenario .....	134
Figure 47. VAT – Login Page Sequence Diagram .....	135
Figure 48. VAT – Main Page .....	136
Figure 49. VAT – Scan Details Sequence Diagram .....	137
Figure 50. VAT – Vulnerability Scan Creation Sequence Diagram .....	139
Figure 51. VAT – Vulnerability Scan Modification Sequence Diagram .....	140
Figure 52. AS – Login Page Sequence Diagram .....	142
Figure 53. AS – Main Page Sequence Diagram .....	143
Figure 54. AS – Attack Details Page .....	144
Figure 55. AS – Attack Creation Sequence Diagram .....	146
Figure 56. AS – Attack Modification Sequence Diagram .....	147
Figure 57. CLC Dashboard Sequence Diagram .....	148
Figure 58. Economic Risk Evaluator – getting risk information sequence diagram .....	149
Figure 59. Economic Risk Evaluator – consulting target configuration sequence diagram .....	150
Figure 60. Economic Risk Evaluator – consulting models selected sequence diagram .....	151
Figure 61. Anomaly Detection Reasoner – simulating the monitored infrastructure sequence diagram .....	152
Figure 62. Countermeasures Simulator – applying mitigation sequence diagram .....	154
Figure 63. CYBERWISER.eu Primer version deployment diagram .....	158
Figure 64. CYBERWISER.eu Basic version deployment diagram .....	159
Figure 65. CYBERWISER.eu Intermediate Version Deployment Diagram .....	161

## List of tables

Table 1: Differences between D2.3 and D2.5 .....	17
Table 2. Table of acronyms .....	18
Table 3. Template for user interface description .....	20
Table 4. Template for use case definition .....	21
Table 5. Required / provided interfaces .....	28
Table 6. A summary of main technologies used in CYBERWISER.eu components .....	43
Table 7. UI.01.000 – CYBERWISER.eu website .....	81
Table 8. UI.01.001 – CYBERWISER.eu Login page .....	82
Table 9. UI.01.002 – CYBERWISER.eu private area .....	83
Table 10. UI.02.001 – CYBERWISER.eu workspace area .....	85
Table 11. UI.02.002 – CYBERWISER.eu Moodle dashboard – trainee perspective .....	87
Table 12. UI.02.003 - CYBERWISER.eu Moodle Dashboard – Trainer perspective .....	89
Table 13. UI.03.000 – Simulated Infrastructure Manager .....	89
Table 14. UI.04.001 – CYBERWISER.eu Scenario listing page .....	90
Table 15. UI.04.002 – CYBERWISER.eu Scenario versions page .....	91
Table 16. UI.04.003 - CYBERWISER.eu Scenario Editor Page .....	92
Table 17. UI.04.004 - CYBERWISER.eu – Network Policies .....	93

Table 18. UI.04.005 – CYBERWISER.eu – VM network details .....	94
Table 19. UI.04.006 – CYBERWISER.eu – Scenario user permissions .....	96
Table 20. UI.04.007 – Add/edit asset .....	97
Table 21. UI.04.008 – CYBERWISER.eu – VM access and visibility .....	98
Table 22. UI.04.005 – CYBERWISER.eu – VNC in modal .....	99
Table 23. UI.05.000 – Performance Evaluator .....	100
Table 24. UI.06.001 – CYBERWISER.eu Digital Library .....	100
Table 25. UI.06.002 – CYBERWISER.eu Document Library .....	101
Table 26. UI.07.001 – Economic Risk Evaluator – Quantitative risk report .....	102
Table 27. UI.07.002 – Economic Risk Evaluator – Mitigation measures .....	103
Table 28. UI.07.003 – Economic Risk Evaluator – Risk reports info .....	104
Table 29. UI.07.004 – Economic Risk Evaluator – Targets configuration .....	105
Table 30. UI.07.005 – Economic Risk Evaluator – Target detail .....	106
Table 31. UI.07.006 – Economic Risk Evaluator – Models selected .....	107
Table 32. UI.08.001 – CYBERWISER.eu – Vulnerability Assessment Tools, Login page .....	107
Table 33. UI.08.002 – CYBERWISER.eu – Vulnerability Assessment Tools, Main page .....	108
Table 34. UI.08.003 – CYBERWISER.eu – Vulnerability Assessment Tools, Vulnerability Scan details page .....	109
Table 35. UI.08.004 – CYBERWISER.eu – Vulnerability Assessment Tools, creating a new vulnerability scan .....	110
Table 36. UI.08.005 – CYBERWISER.eu – Vulnerability Assessment Tools, Modifying a vulnerability scan .....	110
Table 37. UI.09.000 – Monitoring Sensors .....	111
Table 38. UI.10.001 – Anomaly Detection Reasoner landing page .....	112
Table 39. UI.10.002 – Event overview .....	113
Table 40. UI.10.003 – Event detail .....	114
Table 41. UI.10.004 – Alarms overview .....	115
Table 42. UI.10.005 – Alarm detail .....	116
Table 43. UI.11.001 – CYBERWISER.eu – Attack Simulator, Login page .....	116
Table 44. UI.11.002 – Attack Simulator, Main page .....	117
Table 45. UI.11.003 – Attack Simulator, Attack details .....	118
Table 46. UI.11.004 – Attack Simulator, Creating a new attack .....	118
Table 47. UI.11.005 – Attack Simulator, Modifying an attack .....	119
Table 48. UI.12.001 – Countermeasures Simulator – Proposed countermeasures .....	119
Table 49. UI.12.002 – Countermeasures Simulator – Mitigations details and execution .....	120
Table 50. UI.12.003 – Countermeasures Simulator – Mitigations history and remaining budget .....	121
Table 51. UI.13.000 – Economic Risk Models .....	121
Table 52. UI.14.000 – Message Broker .....	121
Table 53. UI.15.001 – Centralized Logging Component Dashboard .....	122
Table 54. UI.16.000 – Infrastructure as a Service .....	122
Table 55. UI.17.000 – Event-Based Service Orchestrator .....	123
Table 56. UC.01.000 – Website .....	124
Table 57. UC.01.001 – Login page user interaction .....	125
Table 58. UC.01.002 – CYBERWISER.eu private area user interaction .....	126
Table 59. UC.02.000 – CYBERWISER.eu workspace area user interaction .....	127
Table 60. UC.02.001 CYBERWISER.eu Moodle dashboard – trainee interaction .....	128
Table 61. UC.02.002 – CYBERWISER.eu Moodle dashboard – Trainer interaction .....	129
Table 62. UC.06.001 – Creating and assigning a new asset .....	130
Table 63. UC.04.001 – Designing a new scenario .....	131
Table 64. UC.04.002 – Using a designed scenario .....	133
Table 65. UC.08.001 – Vulnerability Assessment Tools – Login Page .....	135
Table 66. UC.08.002 – Vulnerability Assessment Tools – Main Page .....	136
Table 67. UC.08.003 – Vulnerability Assessment Tools – Scan Details Page .....	137
Table 68. UC.08.004 – Vulnerability Assessment Tools – Vulnerability Scan Creation .....	138
Table 69. UC.08.005 – Vulnerability Assessment Tool – Vulnerability Scan Modification .....	140
Table 70. UC.11.001 – Attack Simulator – Login Page .....	141
Table 71. UC.11.002 – Attack Simulator – Main Page .....	143
Table 72. UC.11.003 – Attack Simulator – Attack Details Page .....	144
Table 73. UC.11.004 – Attack Simulator – Attack Creation .....	145

Table 74. UC.11.005 – Attack Simulator – Attack Modification.....	147
Table 75. UC.15.001 – Centralized Logging Component Dashboard.....	148
Table 76. UC.07.001 – Economic Risk Evaluator – getting risk information.....	149
Table 77. UC.07.002 – Economic Risk Evaluator – consulting target configuration.....	150
Table 78. UC.07.003 – Economic Risk Evaluator – consulting models selected.....	151
Table 79. UC.10.001 – Anomaly Detection Reasoner – monitoring the simulated infrastructure.....	152
Table 80. UC.12.001 – Countermeasures Simulator – applying mitigation .....	153
Table 81. CYBERWISER.eu versions and components .....	157
Table 82. Functional requirements .....	175
Table 83. Platform requirements .....	178
Table 84. Legal requirements.....	179
Table 85. Security requirements.....	180
Table 86. Usability requirements .....	181
Table 87. Performance requirements .....	182

## Executive Summary

This deliverable is the final output of Task T2.2, entitled “Overall Vision and Design”. It reports the work done during the whole task timeline (M3-M12, November 2018 – August 2019). Although it builds on top of deliverable D2.3, it is a self-contained document and the reader does not need to refer to D2.3 at any point. The design of the CYBERWISER.eu Platform consists of an architecture based on a set of components working together to provide a cyber range platform. The interfaces between the different components are described, remarking which components provide or consume them.

The CYBERWISER.eu Consortium strongly believes that a complete and useful design must describe the product to be implemented from different perspectives. The design task used as input the results of the Task T2.1 on requirements, which provided the specifications both at system and subsystem level, along with the architectural vision and system overview of the CYBERWISER.eu Platform. The Consortium firstly prepared a top-down design considering two different levels of detail (sections 3.1 and 3.2), making fully clear the list of components, their role and how they would be inter-related.

As stated above, it is necessary to provide more views of the design to consider it complete and useful enough. The Consortium provided a perspective based on the software components structure (section 4.1), another considering how the software is deployed (section 4.2), the data view (section 4.3) and also the technology stack (section 4.4) with the full picture of technologies used to develop the different components.

The following step was to consider one by one the components and document their internal design and make clear how they would operate to provide the needed functionalities meeting the requirements. This is covered in section 5.

With the general picture and those of the individual components already in place, the Consortium identified the relevant user interfaces and use cases involving both those interfaces and backend components. This is documented in section 6.

Section 7 makes the link to the business requirements and the section 8 revises the traceability to the requirements.

The design task has been carried out in a continuous, smooth way always keeping the coordination with those project activities that are closely related, such as T2.3 (model development), T2.4 (tools development), T3.1 (integration and assembling of components), T5.1 (requirements from the pilots) and the market needs researched in the different activities of WP6. As a result, the CYBERWISER.eu Platform will be depicted at various levels of detail by combining different views of the system.

The main takeaways of the document are the following:

- Two levels of detail of the overall architecture diagram to provide the high-level perspective.
- Description of the platform components, namely: Web Portal, Cross-Learning Facilities, Message Broker, Centralized Logging Component, Digital Library, Training Manager, Simulated Infrastructure Manager, IaaS infrastructure, Anomaly Detection Reasoner, ADR-Agent, Monitoring Sensors, Event-based Service Orchestrator, Vulnerability Assessment Tool, Attack Simulator, Countermeasures Simulator, Economic Risk Models, Economic Risk Evaluator, Pluggable Deployment Manager and Performance Evaluator. This description includes the main feature, role in the platform, internal architecture and internal operation.
- Summary of interfaces among components and communication protocols used.
- A 3-tier-diagram representing the software components structure perspective.
- Software deployment perspective.
- The data view, considering the data model and the data flows that take place in relation to the platform operation.
- The technology stack, showing the richness of the platform from the perspective of the tools and means used to develop and integrate the components.

- Main user interfaces definition.
- Use cases definition.
- Requirements traceability.

# 1. Introduction

## 1.1 Purpose and scope of the document

Deliverable D2.5, entitled “Platform Design, Final Version” is the final delivery produced by Task T2.2 entitled “Overall vision and design” within Work Package WP2 entitled “Requirements, Design and Building Blocks”. This task started in month M3 (November 2018) and finalized in month M12 (August 2019). D2.5 builds on top of D2.3, which is the intermediate delivery of the task and was submitted in M6 (February 2019). WP2 starts in M1 (September 2018) and finishes in M20 (April 2020).

This deliverable continues and completes the approach to the design of the platform. The components of the platform are presented and their role is explained. Additionally, the interfaces establishing the relations among the components are described.

We consider that the complexity of the CYBERWISER.eu Platform demands the design to be addressed from different perspectives, as one single perspective will never provide all the needed information for the subsequent implementation activities. In D2.3 we followed a top-down initial approach, creating two diagrams of different level of detail reflecting the general picture. This provided an overall perspective as starting point. Then, two more perspectives were added: one based on the software components structure (section 4.1) and another based on the software deployment perspective (section 4.2). Then, for each component and internal diagram was presented to better understand how they work. This was done in section 5 where we addressed the internal detail of the building components.

In D2.5 all the contents generated in D2.3 have been refined and updated, and more relevant information has been added. Such is the case of the data view (section 4.3) where the data needed to run an exercise is presented in a structured way and a general vision of what data is exchanged among the components, highlighting the producers and consumers, is provided as well. The technology view (section 4.4) summarizes the technology stack behind the development of CYBERWISER.eu indicating what is used to develop each component. In section 6.1 the user interfaces that can be found within the platform and the navigation through them, as well as the integration with the project portal, are documented, being the basis for the definition of use cases in section 6.2.

All in all, D2.5 broadens and completes the scope of D2.3, aligning it with the objectives of the task and the WP, being an important milestone and relevant entry point for ongoing and future tasks along the project.

## 1.2 Structure of the document

The document is structured in the following way

- Executive summary;
- Section 1: Introduction to the deliverable, explaining the purpose and scope, the structure of the document itself and its relation to other activities carried out within the project. Also, for convenience, a glossary of acronyms is added in this section;
- Section 2: The methodology to produce the document is explained. This section addresses the following aspects: the engineering process of the system design and architecture, the modelling languages, the user flows and use cases and how to establish the traceability between the requirements and the design;
- Section 3: It is the first chapter addressing the design itself. Following a top-down approach, the component diagram is presented with two levels of detail. Then, the different components are briefly presented, explaining their role in the platform. All the details about each component are provided in section 5;
- Section 4: It provides additional design perspectives for the sake of completeness: the software components perspective, which is in essence a tier diagram, the software deployment perspective, the data view and the technology view;



- Section 5: It zooms into each of the building components, providing further information about them and explaining their internal operation with accompanying diagrams;
- Section 6: Presents the different user interfaces that can be found within the platform, the navigation through them and a list of different use cases of the platform;
- Section 7: It addresses the impact of the business requirements on the design;
- Section 8: It is the final traceability between the design and the requirements;
- Section 9: Conclusions and closing remarks.

### 1.3 Relation to other work in the project

Due to the fact that CYBERWISER.eu is a very complex platform, during the definition of the project it was decided that the task would have two stages:

- In the initial stage, covering from M3 (November 2018) to M6 (February 2019), a first approach was made to the challenge of designing the platform. In this stage it was very relevant to coordinate with the requirements elicitation task (T2.1 “Requirements” [M1-M6]) which was the main information input to the task. It was important to ensure that the design covers what is demanded in the requirements. Also, it was relevant to note that the delivery of the final requirements (D2.2, M6 <sup>1</sup>) was synchronised with the delivery of the initial design (D2.3, M6).
- During the second stage, running from M7 (March 2019) to M12 (August 2019) the design has been refined and the aspects that were left at a high level in D2.3 were more and more detailed as T2.2 has been running in parallel with the development activities T2.3 “Models adaptation and development” (M4-M20) and T2.4 “Tools adaptation and development” (M4-M20), and also with the integration activities taking place in T3.1 “Infrastructure and platform development” (M6-M24). As a result, the design has been consolidated and delivered in this document that closes task T2.2. In addition, there is mutual influence between WP2 and WP4, that provides training goals and objectives, and WP6 that provides business requirements.

This document continues and finishes the work documented in D2.3. We are adopting the approach of treating this deliverable as a standalone document, so it can be read without needing to check D2.3 for any kind of content. In order to make clear what has changed from the previous version and what has not, this table reports what is new in D2.5 and what contents are inherited from D2.3.

D2.5 section	D2.3 section	What is new: differences
1. Introduction	Section 1	Several updates across the section
1.1 Purpose and scope of the document	Section 1.1	The section has been slightly modified for updates
1.2 Structure of the document	Section 1.2	The section has been slightly modified for updates
1.3 Relation to other work in the project	Section 1.3	The section has been slightly modified for updates
1.4 Glossary of acronyms	Section 1.4	The glossary has been updated with new entries
2. Methodology	Section 2	Introduction added
2.1 System architecture / design engineering process	Section 2.1	Updated to reflect the additional detail of the D2.5
2.2 Modelling languages	Section 2.2	No changes
2.3 User flows and use cases		New section in D2.5

<sup>1</sup> The initial requirements version is compiled in D2.1 (M3, November 2018)



D2.5 section	D2.3 section	What is new: differences
2.4 Traceability between requirements and design	Section 2.3	Minor modifications.
3. CYBERWISER.eu top-down final design	3. CYBERWISER.eu top-down initial design	This section has several modifications for updates
3.1 Level 1 component diagram	Section 3.1	This section has been slightly modified for updates
3.2 Level 2 component diagram	Section 3.2	This section has several modifications for updates. Sections 3.2.1, 3.2.10 slightly changed its title
4. Other design perspectives	Section 4	The section has been added new contents and has refined and updated the ones existing in D2.3
4.1 Software components structure	Section 4.1	There have been some minor modifications for updates.
4.2 Software deployment perspective	Section 4.2	This section has been slightly modified
4.3 Data view		New section in D2.5
4.4 Technology view		New section in D2.5
5. Detail of the building components	Section 5	Several modifications along the section
5.1 Web Portal	Section 5.1	Minor additions
5.2 Cross-Learning Facilities	Section 5.2	Minor additions
5.3 Simulated Infrastructure Manager	Section 5.3	Minor modifications. Section 5.3.1 slightly changed its title
5.4 Training Manager	Section 5.4	Minor modifications
5.5 Performance Evaluator	Section 5.5	The description of the component has been notably extended, including two diagrams
5.6 Digital Library	Section 5.6	Minor modifications
5.7 Economic Risk Evaluator	Section 5.7	Minor modifications
5.8 Event-Based Service Orchestrator (EBSO)		New section in D2.5
5.9 Vulnerability Assessment Tools	Section 5.8	Section updated and expanded
5.10 Monitoring Sensors	Section 5.9	No changes
5.11 Anomaly Detection Reasoner	Section 5.10	No changes
5.12 Attack Simulator	Section 5.11	Section updated and expanded
5.13 Countermeasures Simulator	Section 5.12	The description of the component has been notably extended, including two diagrams
6. Using the platform		New section in D2.5
6.1 User flows through the user interfaces		New section in D2.5
6.2 Use cases		New section in D2.5
7. Impact of the business requirements	Section 6	This section has been slightly modified
7.1 CYBERWISER.eu versions and components	Section 6	This section has been slightly modified
7.2 Primer Version Deployment Diagram	Section 6.1	This section has been slightly modified
7.3 Basic Version Deployment Diagram	Section 6.2	This section has been slightly modified
7.4 Intermediate Version Deployment Diagram	Section 6.3	This section has been slightly modified

D2.5 section	D2.3 section	What is new: differences
8. Final requirements traceability	Section 7	This section has been updated to synchronize with the final version of requirements in D2.2.
9. Conclusions	Section 8	The section has been updated

Table 1: Differences between D2.3 and D2.5

With the delivery of this document, the task T2.2 concludes. The results obtained are relevant especially for the development activities in T2.3 and T2.4 and the integration activities running in task T3.1 that will use this document for guidance and as reference point to make clear the road ahead. The ultimate goal is to deliver the integrated platform, that fulfills the goals of the project and satisfies the expectations of the different stakeholders involved in CYBERWISER.eu.

## 1.4 Glossary of Acronyms

Acronym	Description
ADR	Anomaly Detection Reasoner
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
AS	Attack Simulator
AWS	Amazon Web Services
CA	Certificate Authority
COTS	Commercial-Off-The-Shelf
CSAR	Cloud Service Archive
CLC	Centralized Logging Component
CPU	Central Processing Unit
CS	Countermeasures Simulator
DL	Digital Library
EBSO	Event-Based Service Orchestrator
ERD	Entity Relationship Diagram
ERE	Economic Risk Evaluator
ERM	Economic Risk Models
GUI	Graphical User Interface
HTTPS	Hyper Text Transfer Protocol Secure
IaaS	Infrastructure as a Service
ICT	Information and Communications Technology
IP	Internet Protocol
JSON	Java Script Object Notation
KVM	Kernel-based Virtual Machine
LCMS	Learning Content Management System
NFS	Network File System
MB	Message Broker
OWASP	Open Web Application Security Project
PaaS	Platform as a Service
PE	Performance Evaluator
RAE	Risk Assessment Engine

Acronym	Description
RBAC	Role-Based Access Control
REST	Representational State Transfer
SIEM	Security Information Event Management
SIM	Simulated Infrastructure Manager
SSH	Secure Shell
SSO	Single Sign-On
TOSCA	Topology and Orchestration Specification for Cloud Applications
TM	Training Manager
UI	User Interface
UML-CD	Unified Modelling Language – Component Diagram
URL	Uniform Resource Locator
VAT	Vulnerability Assessment Tool
VM	Virtual Machine
VNC	Virtual Network Computing
YAML	Yet Another Markup Language
WP	WorkPackage
ZAP	Zed Attack Proxy

Table 2. Table of acronyms

## 2. Methodology

For any large engineering effort, it is important to define the methodology that will be followed to produce the relevant design artefacts. This is also the case for the CYBERWISER.eu project. The following sections will provide the methodology used for this project.

### 2.1 System architecture / design engineering process

#### 2.1.1 Methodology for defining the system architecture

The goal of a system architecture process is to translate the established system and subsystem requirements into system design elements, usually collected in different views and diagrams, showing the system from different perspectives.

Within the CYBERWISER.eu project, this process began with a review of the system-level requirements, subsystem requirements, and preliminary architectural vision from the CYBERWISER.eu proposal. The combination of these elements with the architectural constraints provided by the existing system's assets lead to a preliminary version of the CYBERWISER.eu system architecture, proposed in D2.1 as a system overview. This preliminary architecture has been refined for the D2.3 and it is further detailed in this D2.5, taking into consideration the final version of the requirements from D2.2.

The design team conducted a functional decomposition of the requirements, defining and assigning specific functions required to address each requirement to its specific subsystem(s). A primary focus has been given to non-functional requirements, since they are the main drivers of a system architecture. From this, an initial separation of concerns has been developed for the system and each subsystem. Workflows and high-level use cases or user stories defined by the requirements have been reviewed and translated into system components and sub-components, interfaces and data flows.

Key design drivers have been identified from this analysis effort. These are elements expected to have the greatest impact on key performance indicators or to present specific constraints on the system architecture or a higher level of risk to the development and deployment of the system. Examples may include specific system performance needs, limitations of external interfaces, development or acquisition costs, operating and maintenance costs, flexibility required by specific project elements or identified project risks, feature prioritization, project schedule limitations, security or existing assets with reduced adaptability.

With these key design drivers, the team developed a list of design elements and specific strategies to address the system and subsystem requirements and key design drivers. From this activity, an overall system architecture, including high-level design elements, data flows, data entity relationship diagrams for each system and subsystem components and their functions and primary system interfaces (external and internal), have been produced. CYBERWISER.eu design has been hence developed in iterations, from the high-level system elements to the low-level sub-systems, with the aim of considering all elicited requirements.

After the first iterations, a preliminary system overview could be produced (depicted in deliverable D2.1). With further iterations, it was possible to detail the initial design from the overview in D2.1 to a detailed level (D2.3 and this document, D2.5).

Since parts of the CYBERWISER.eu Platform will be composed by commercial-of-the-shelf (COTS) software components (Hypervisor, Platform as a Service [PaaS] and Infrastructure as a Service [IaaS]), architecture and design will consider these elements as black boxes, while the interfaces between CYBERWISER.eu components and COTS software components, used for the integration, have been detailed.

#### 2.1.2 Conclusions

The CYBERWISER.eu Platform architecture/design has been developed using various iterations of the described design methodologies. The resulting architecture will depict the CYBERWISER.eu Platform at various levels of details by combining different views of the system.

## 2.2 Modelling languages

The chosen language for designing the architecture and the detailed design is Unified Modelling Language (UML). While in D2.3 mainly Component and Deployment diagrams were used, this D2.5 document additionally encompasses a number of Sequence Diagrams in order to better depict the interactions between components.

Within D2.5, a Design Object Model will be then created, identifying in detail the needed interfaces and the main data types. The data model will be modelled with UML Class Diagrams. As CYBERWISER.eu is a super system composed by multiple components, interfaces are a critical element. The detailed data model will hence cover in particular the data exchanged between the components' interfaces.

The CYBERWISER.eu Platform Architecture/Design will be developed with the aim of creating a consistent Design Project covering all components, interfaces and interactions between components and between components and COTS software at various levels of details. Every requirement will be traced to the design, in order to be able to manage changes and updates (and validate these changes) both in the design or in the requirements.

## 2.3 User flows and use cases

To describe the user flows, we will describe each interface the user may find when navigating through the platform. To describe interfaces, we will use the following template:

User Interface name	<Write the name of the user interface>
UI Id	<UI.x.y> X is a number identifying a component, as follows: <ol style="list-style-type: none"> <li>1. Web Portal</li> <li>2. Cross-Learning Facilities</li> <li>3. Simulated Infrastructure Manager</li> <li>4. Training Manager</li> <li>5. Performance Evaluator</li> <li>6. Digital Library</li> <li>7. Economic Risk Evaluator</li> <li>8. Vulnerability Assessment Tool</li> <li>9. Monitoring Sensors</li> <li>10. Anomaly Detection Reasoner</li> <li>11. Attack Simulator</li> <li>12. Countermeasures Simulator</li> <li>13. Economic Risk Models</li> <li>14. Message Broker</li> <li>15. Centralized Logging Component</li> <li>16. Infrastructure as a Service</li> <li>17. Event-Based Service Orchestrator</li> </ol> Y is a number running sequentially.
Description	<This is a free style text describing what the user sees in the interface. Some examples: information display areas, indicating how the information is provided (text, graphics... etc); buttons to perform certain actions or to move to another interface; menus and/or submenus to move to another interface, pop-ups; logout buttons...>
Purpose	<Describe the purpose of such interface>
Navigation and user interaction	<Description of the actions the user may perform of the interface and where such action will take him, basically to another interface. If something relevant happens in the backend it can be explained here even though the use case sections is envisioned to provide those details>
Other comments	<For any other information to be given, if any>

Table 3. Template for user interface description

Once all the interfaces have been described, they will be represented as boxes in a diagram in which relationships among them will be described by means of arrows.

To describe use cases, we will use a template and a sequence diagram per case.

The template to be used is the following:

Use Case Name	<Meaningful and descriptive name of the use case>
UC id	<UC.x.y> X corresponds with the component number for the asset in section 6.1. Y is a number running sequentially
Description	<Explanation in free-style text of what the functionality does in this use case>
Actors	<Stakeholders involved, these are the platform itself on one side and the different human roles that might interact with the platform>
Objects	<Entities being part of the use case and will appear in the sequence diagrams. They can be any of the user interfaces previously presented or backend components. The sequence diagram will explain the interaction among them>
Basic flow	<Basic flow explanation, in essence it will be the explanation of the sequence diagram that will appear along with the use case template>
Preconditions	<Constraints that must be fulfilled by the system at the moment to start the flow of the use case, for it to work properly>
Postconditions	<Constraints to be met by the system when the execution of the use case ends>
Dependencies	<Dependencies indicate when one use case depends on the execution of another or includes its execution as part of its own behaviour>

Table 4. Template for use case definition

## 2.4 Traceability between requirements and design

As depicted in Section 2.1.1, Requirements are the foundation of the CYBERWISER.eu architecture/design. All requirements (functional and non-functional) have been modelled in Requirements Diagrams and connections between them are designed and can be accessed via Requirements Matrixes in D2.1 and D2.2. Following that path, if a requirement needs to change, it is possible to easily find if any other Requirement has a connection or a dependency with it: changes can then be evaluated and tested with respect of the other requirements.

Since the global architecture is built over the Requirements, it is important to trace the produced design with the related requirements. This allows verification of the level of *satisfaction* (or *realization*) of the design over the Requirements as well as change management. If the design changes, it can be validated over the traced requirements, and vice versa. Requirements Realization Matrixes have been created, reported in Section 8 and they will potentially be updated during the lifetime of the project. The aim is that the design elements realize ALL functional and non-functional requirements.

Non-functional requirements usually are not easily traced to single design elements, since they regard architectural decisions involving multiple aspects of the design. When needed, a rationale (for example, a system or software design pattern) is added to the non-functional Requirements Realization Matrixes.

### 3. CYBERWISER.eu top-down final design

The design of the CYBERWISER.eu Platform consists of an architecture based on a set of components working together to provide a cyber-range training platform. To show the relationships between the different components in our platform, we have modelled them using the component diagram from the UML (UML-CD) and, emphasizing the required/provided interfaces approach by remarking which components provide or consume them.

In this case, we have covered the design of the CYBERWISER.eu Platform by defining two different levels of abstraction by means of two different component diagrams. The level 1 component diagram has a higher level of abstraction than the level 2 component diagram. Level 1 component diagram provides a general overview of the platform which shows the platform as observed by the end user more than by the developer. The level 2 component diagram is focused on the Cyber-Range Service component available in the level 1 component diagram and has detailed relations among its components.

#### 3.1 Level 1 component diagram

This component diagram is depicted in Figure 1 and represents the user accessing the platform's single point of origin by using its web browser to access the CYBERWISER.eu Web Portal. Every user interaction should start here including accessing the external services being provided by the Cyber-Range Service.

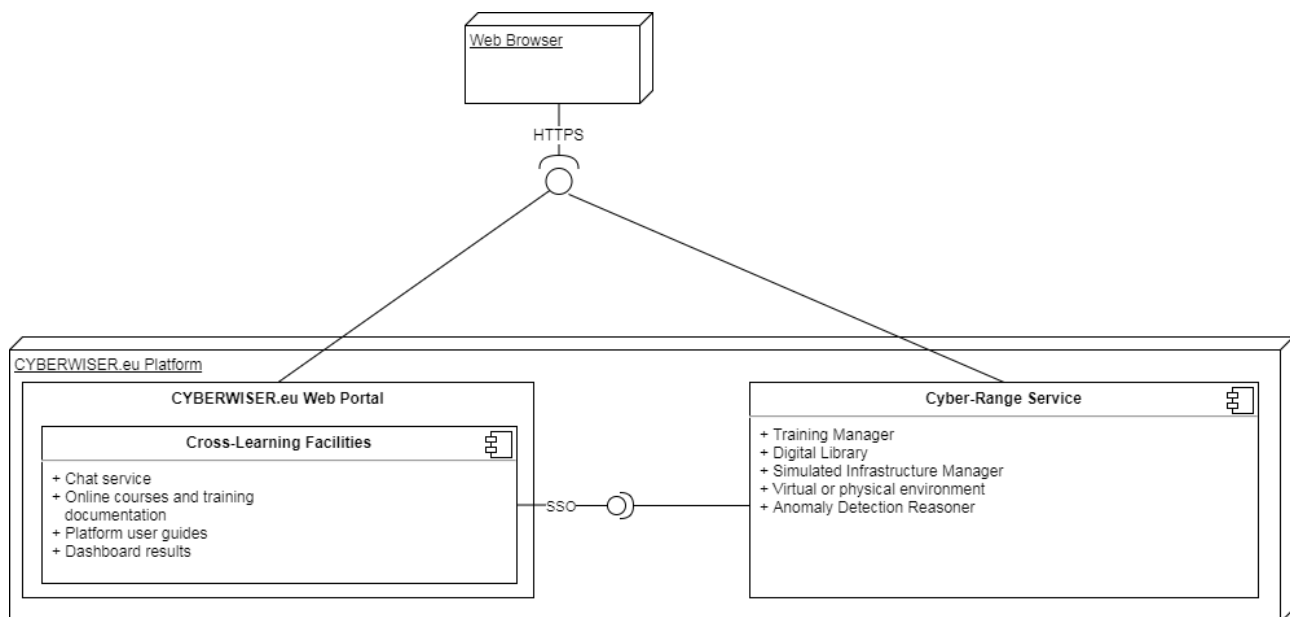


Figure 1. Level 1 component diagram

The users can access the CYBERWISER.eu Platform through a web browser using HyperText Transfer Protocol Secure (HTTPS). The web browser can consume the interface provided by the CYBERWISER.eu Web Portal or even has direct access to the Cyber-Range Service using the interface provided by it.

The CYBERWISER.eu Web Portal is the first container for the all platform and its services and will provide the launching Graphical User Interface (GUI). From the point of view of the end-users (operators or trainees), it is the entry point to access the system. The CYBERWISER.eu Web Portal enables the users to access different systems in the platform with a single identification instance. This authentication process is provided by the Cross-Learning Facilities component through the Single Sign-On (SSO) service, which allows users to transparently access multiple applications inside the platform using a single login. The Cross-Learning Facilities component will be accessible from the CYBERWISER.eu Web Portal and will also provide, among others, learning materials and documentation, training courses, communication tools such a chat service,



dashboards for the users and access to the Cyber-Range Service. The user can access the platform and all its services using the SSO service through the CYBERWISER.eu Web Portal, however, if there is no learning service required, the user can access directly to the Cyber-Range Service by login in it.

### 3.2 Level 2 component diagram

The level 2 component diagram is focused on the Cyber-range service and zooms into this component as depicted in Figure 1. The details of the Cyber-range service are shown in Figure 2 and encompass all aspects related to the cyber-range environment and the elements of the training scenarios.

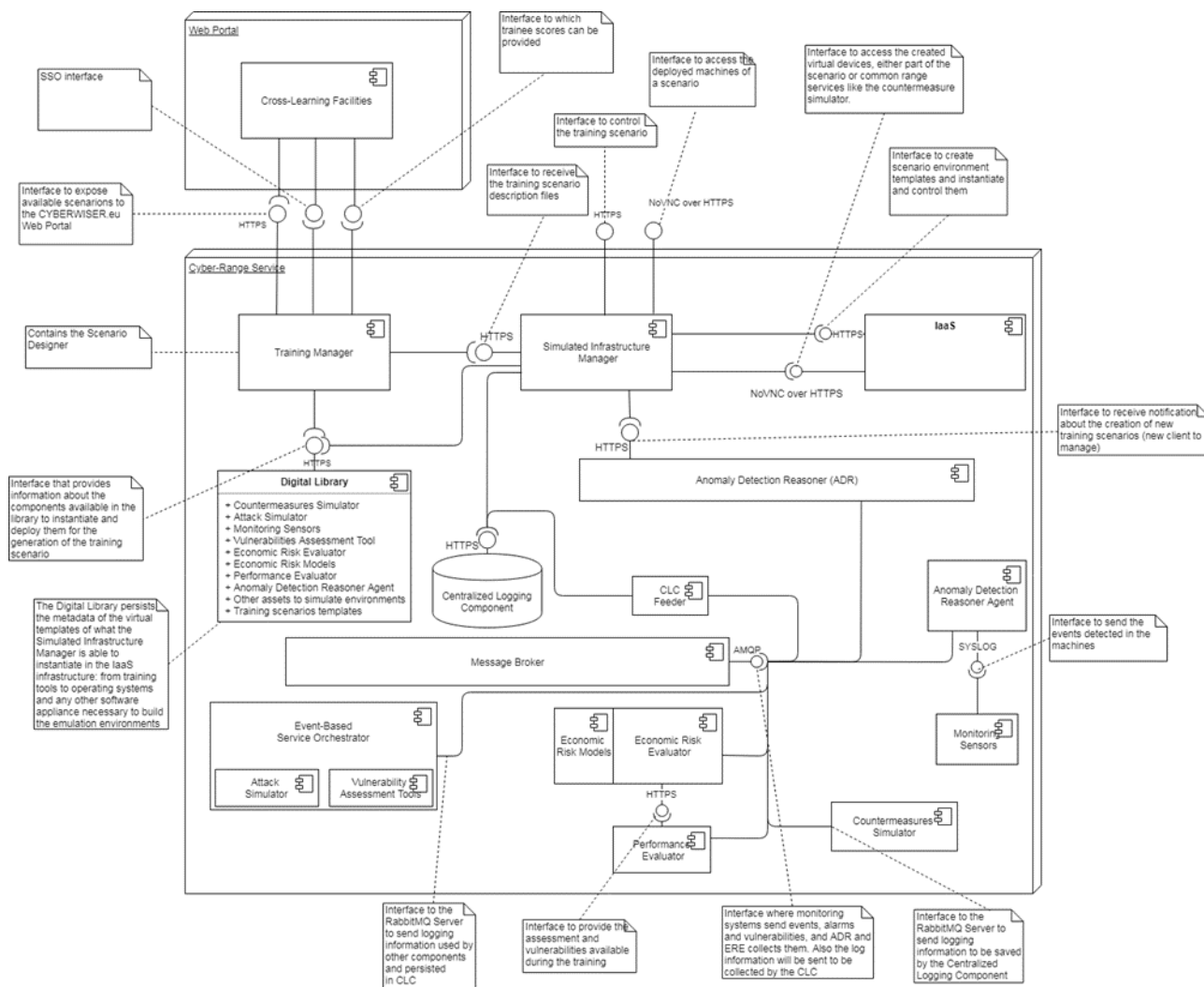


Figure 2. Level 2 component diagram

The Cyber-Range Service is composed by a set of components that interoperate and work together to provide the users with the proper environment and tools for the management and use of cyber-range training exercises and their monitoring.

The Cyber-Range Service supports virtual or physical environments since a training scenario may consist of virtual and physical elements. It has some components which are always available in the system, regardless of whether there are training exercises in progress. The rest of the components in the component diagram depends on the definition of the training scenarios and are deployed on demand. To explain the entire component diagram, the components and their relationship represented in Figure 2 are defined below.

### 3.2.1 Message Broker (MB)

The Message Broker (MB) enables communication between various components of the CYBERWISER.eu Platform in the scope of a single training exercise. It is implemented as an instance of the RabbitMQ service, exposing an Advanced Message Queuing Protocol (AMQP) interface to exchange messages from multiple sources. Depending on the types of messages and configuration, MB directs the messages to all the appropriate components. Messages sent through the MB include events, alarms, vulnerability reports, trainees' scoring data, component configuration and logging data to be persisted in the Centralized Logging Component (CLC).

#### *CLC Feeder*

CLC Feeder is a microservice, acting as a bridge between the MB and the CLC. It listens to logging messages coming from all the components inside an exercise through the MB and forwards them to the HTTPS interface of the CLC, where they are persisted.

### 3.2.2 Centralized Logging Component (CLC)

The Centralized Logging Component (CLC) is a database storing all the data that needs to be persisted from the training exercises. CLC is deployed outside of the scenario environment and receives messages from all the components inside the exercise environments via the CLC Feeder. Data saved in the CLC includes trainees' scoring information (making it available to users from the Cross-Learning Facilities) as well as all events detected in an exercise (to support potentially required detailed inspection of the current and past exercises by the trainers) and detailed logging information (for debugging all parts of the exercise environment and its components by the developers or the platform operators).

### 3.2.3 Digital Library (DL)

The Digital Library (DL) component is the repository for the predefined set of virtual/physical elements and training scenarios, so it will contain the metadata and templates of the elements, such as operating systems, preconfigured virtual machines, or any other software to deploy on any virtual machine. Within the list of virtual machine templates, the DL also contains the rest of components to monitor and control the training scenario:

- Countermeasures Simulator (CS).
- Attack Simulator (AS).
- Monitoring Sensors.
- Vulnerabilities Assessment Tool (VAT).
- Economic Risk Evaluator (ERE).
- Economic Risk Models (ERM).
- Performance Evaluator (PE).
- Anomaly Detection Reasoner (ADR) Agent.

The DL will also contain a list of metadata of the available attacks that can be automatically executed during the course of the scenario. The DL maintains the list of available attacks and the available configuration parameters for each script.

### 3.2.4 Training Manager (TM)

The TM is the component responsible for the management (creation, editing and deletion) of the activities and the training scenarios within each activity. It contains the Scenario Designer, which can generate, update or remove a training scenario design. This component provides a web interface to manage it from the browser. It requires the interface provided by the DL to manage the catalogue of pre-defined training scenarios and virtual/physical elements when defining a scenario. The TM also requires an interface to send the training scenario requests previously designed to the SIM. In addition, this interface provides the capability to retrieve the achieved scores from the CLC through the SIM.

### 3.2.5 Simulated Infrastructure Manager (SIM)

The SIM component is a key component in the infrastructure since it is responsible for instantiating and deploying the training scenarios. It also allows the access to the machines and components available in the scenario environment. This component provides an HTTP(S) interface that is used by the TM to send the training scenario requests. The SIM retrieves the required metadata using the DL interface to translate the training scenario requests into virtual environment templates. In order to instantiate the virtual environment templates, the SIM uses two HTTP(S) interfaces from the IaaS, which will also be used to control them. In addition, the SIM uses the NoVNC<sup>2</sup> over HTTPS interface to access the deployed machines. These machines can either be the specific machines related to the specific scenario or the other components of the platform itself. The SIM component provides the HTTPS and the NoVNC over HTTPS interfaces to control the training scenarios and the deployed machines, to be used from a browser by the end-users. Finally, this component can implement connections with:

- The ADR to notify about the instantiation of new training scenarios and to access the information provided by this component.
- The CLC to retrieve the information about the performance evaluation reports.

### 3.2.6 Infrastructure as a Service (IaaS)

The IaaS is the component that offers services regarding virtualized computing resources that are both on-demand and scalable. It allows the generation, instantiation and control of the scenario environments through the HTTPS interface; and provides a NoVNC over HTTPS interface to access the created virtual machines, either part of the scenarios or common range services represented as cyber-range components in our diagram.

### 3.2.7 Anomaly Detection Reasoner (ADR)

This component receives the events generated by the ADR Agent. The sensors deployed in the monitored simulated infrastructure send the alerts to the ADR Agent. The ADR Agent generates the corresponding events to be sent to the Message Broker and will be collected by the Anomaly Detection Reasoner. Then, the ADR analyses the events and raises the corresponding alarms according to the correlation rules. For this purpose, the ADR connects to the AMQP interface of the MB. The ADR will expose HTTPS interfaces to receive notifications about new training scenario instantiations and, to provide access to its control panel to check events and alarms triggered in the monitored training scenarios.

### 3.2.8 Anomaly Detection Reasoner Agent (ADR Agent)

The ADR agent components collect the alerts coming from the monitoring sensors deployed into machines along the training scenarios. They provide a Syslog interface to receive the alerts and use the AMQP interface of the RabbitMQ server to send the corresponding events when it is applicable in order to reach the ADR application.

### 3.2.9 Monitoring Sensors

These components are different software elements distributed along the training scenario machines to be monitored. These sensors use the Syslog interface provided by the ADR Agent to send the alerts detected. There are two kinds of sensors based on the monitoring target:

- Host activity. This kind of sensor should be installed on each host to be monitored. It will provide information about potential threats in the local host.
- Network activity. This sensor should be installed for each network in the infrastructure. It will monitor and analyse network traffic to detect and prevent intrusion in the network.

<sup>2</sup> NoVNC is a browser-based VNC client implemented using HTML5 Canvas and Web Sockets

The monitoring sensors are addressed on detail in section 5.10.

### 3.2.10 Event-based Service Orchestrator (EBSO)

Event-Based Service Orchestrator (EBSO) is a component that prepares lightweight environments for containerized services, scheduling and orchestration of such services. It acts as a shared backend supporting VAT and AS components. It also deduces exercise-related events based on the results of the executions of managed services (attacks, scans). EBSO is controlled by the user interfaces of VAT and AS. It propagates output of the managed services via the MB to other parts of the platform (vulnerability reports to the ERE, logging and exercise events to the CLC).

### 3.2.11 Vulnerability Assessment Tools (VAT)

The VAT component analyses targets by running vulnerability scans. Scans may be triggered automatically or by end-users of the platform. To the end-users, VAT offers a Web interface for configuration of scanners. The VAT uses EBSO as a backend to manage the scans and propagate the results. It communicates with EBSO via AMQP.

### 3.2.12 Attack Simulator (AS)

The AS launches attacks against the targets. It either automates the attacks in the absence of a human red team or allows end-users of the platform to trigger them via a Web interface. The attacks are based on pre-defined scripts and tools capable of executing them. The AS uses EBSO as a backend for running the attacks and propagation of results. It communicates with EBSO via AMQP.

### 3.2.13 Countermeasures Simulator (CS)

It provides the users with a series of predefined countermeasures or applies them automatically when there is no human blue team. The details of the component are presented in Section 5.13.

### 3.2.14 Economic Risk Models (ERM)

The ERM will provide an estimation and an impact of the risk level of a training scenario in terms of likelihood and monetary loss. The ERM are part of the specific training scenario configuration and are provided in the form of script files that will provide a quantitative assessment of the training scenarios. The scripts will be deployed together with the ERE component which will execute them.

### 3.2.15 Economic Risk Evaluator (ERE)

The ERE component is responsible for providing economic assessment reports based on the training scenario configuration, the security vulnerabilities detected, and the events and alarms triggered. The vulnerabilities will be collected from the MB interface coming from the VAT application, while the source of the events and the alarms will be the ADR that also sends them to the Message Broker. This component provides an HTTPS interface to request the resulting assessment reports and related information.

### 3.2.16 Performance Evaluator (PE)

The PE application will be in charge of evaluating the performance of the trainees. For this purpose, the PE must use:

- the economic assessment reports from the training exercise which are received through the HTTPS interface provided by the ERE component,
- the monitoring information captured by the sensors deployed within the simulated infrastructure, obtained through the MB,
- the vulnerability reports obtained also from the exercise by means of the MB,

- the specific events generated from the exercise sent using logs though the RabbitMQ, flags generated automatically for instance,
- flags submitted by the user himself by means of a questionnaire responded during the exercise within the scenario itself.

An overview of the connections among the components in the component diagram is shown in Table 5

Required interfaces	Provided interfaces	Protocol
Web Portal	Cross-Learning Facilities	HTTPS REST
Training Manager	Cross-Learning Facilities	HTTPS REST
Cross-Learning Facilities	Training Manager	HTTPS REST
Training Manager	Simulated Infrastructure Manager	HTTPS REST
Simulated Infrastructure Manager (optionally)	Pluggable Deployment Manager	HTTP REST
Training Manager	Digital Library	HTTPS REST
Simulated Infrastructure Manager	Digital Library	HTTPS REST
Performance Evaluator	Economic Risk Evaluator	HTTPS
Monitoring Sensors	Anomaly Detection Reasoner - Agent	Syslog
Simulated Infrastructure Manager	Anomaly Detection Reasoner	HTTPS
Economic Risk Evaluator	Economic Risk Models	Script files
CLC Feeder	Message Broker	AMQP
Event-Based Service Orchestrator	Message Broker	AMQP
Anomaly Detection Reasoner	Message Broker	AMQP
Anomaly Detection Reasoner - Agent	Message Broker	AMQP
Economic Risk Evaluator	Message Broker	AMQP
Performance Evaluator	Message Broker	AMQP
Countermeasures Simulator	Message Broker	AMQP
Simulated Infrastructure Manager	Centralized Logging Component	HTTPS REST
CLC Feeder	Centralized Logging Component	HTTPS REST
Simulated Infrastructure Manager	Infrastructure as a Service	XML RPC
Simulated Infrastructure Manager	Infrastructure as a Service	NoVNC over HTTPS
Pluggable Deployment Manager	Infrastructure as a Service	XML RPC
Vulnerability Assessment Tools	Event-Based Service Orchestrator	AMQP

Required interfaces	Provided interfaces	Protocol
Attack Simulator	Event-Based Service Orchestrator	AMQP

Table 5. Required / provided interfaces

## 4. Other design perspectives

### 4.1 Software components structure perspective

The CYBERWISER.eu Platform is presented as a web application organized as a multi-tier architecture with three different layers. These tiers conform a group of logical components which remarks the functional independence of all software elements.

The three-tier architecture is really useful for integrating third-party software into an existing platform. It is also helpful for separating, modularizing and scaling front-end, back-end and resources development.

This three-tier software structure perspective is depicted in Figure 3 and shows the logical separation:

- Presentation tier encompasses the components running on the client side and presented to the end-users.
- Business tier is the logical layer and includes the provided services within the CYBERWISER.eu platform.
- Back tier is the resources and data layer.

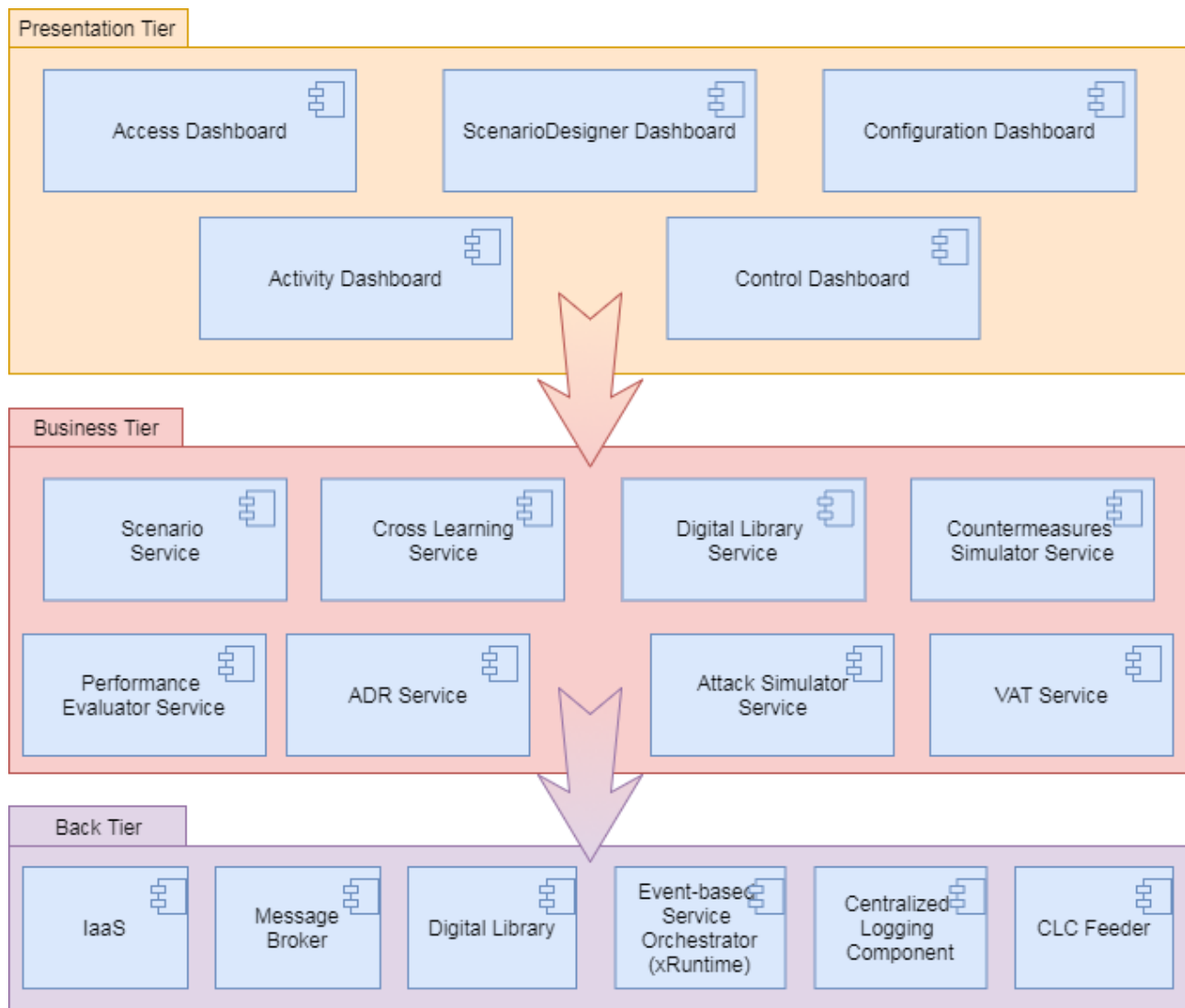


Figure 3. 3-tier-software structure perspective



#### 4.1.1 Presentation tier

The presentation tier provides an interface to the end-users and presents the results to the users through a web browser (client application) by communicating with the business tier. The Dashboard Components are the top level of the applications and will provide access to the user interface in a friendly and intuitive way. It is the first layer which the users can access directly to interact with the CYBERWISER.eu Platform in order to manage the activities and exercises, configure the training scenarios or to access and orchestrate the different applications or machines.

#### 4.1.2 Business tier

The business tier is also known as mid-tier or application tier since it is the logical layer which provides full control over the applications' functionality. Business tier is very close to the component diagrams presented in Section 1 since it consists of a set of services provided by the components presented there. The different services included in Figure 3 can be mapped to the components detailed in those designs: The Cross-Learning Service, as its name suggests, will be provided by the Cross-Learning Facilities component; the Scenario Service will be included in the TM and the SIM components; the DL Service will be able to contribute with the catalogue information through the DL component; the CS Service will be covered by the component of the same name; the PE Service encompass the ERM/ERE and the PE components; the AS and the VAT services are covered by the components with the same name; while the ADR Service will be supported by the ADR, the ADR Agent and the Monitoring Sensors components. We can notice that each service in the business tier is responsible for a set of logical functions within the CYBERWISER.eu Platform.

#### 4.1.3 Back tier

The main goal of this tier is to provide data access and exchange and to supply the required resources. The IaaS will provide resources support to the rest of the applications. The MB will facilitate the exchange of information and the CLC will provide persistent storage, while the CLC Feeder will act as a gateway for the messages from the MB to the CLC. The DL will provide the metadata for the components in the training scenarios. The EBSO will orchestrate the AS and VAT components in the infrastructure.

## 4.2 Software deployment perspective

The CYBERWISER.eu Platform consists of a set of components that will provide the overall functionality envisaged for the platform. Every component needs to be deployed in the systems to support the whole platform availability and operation. As the platform deals with virtualized environments for training scenarios as a major part of its functionality the deployment of the platform is divided into three distinct parts.

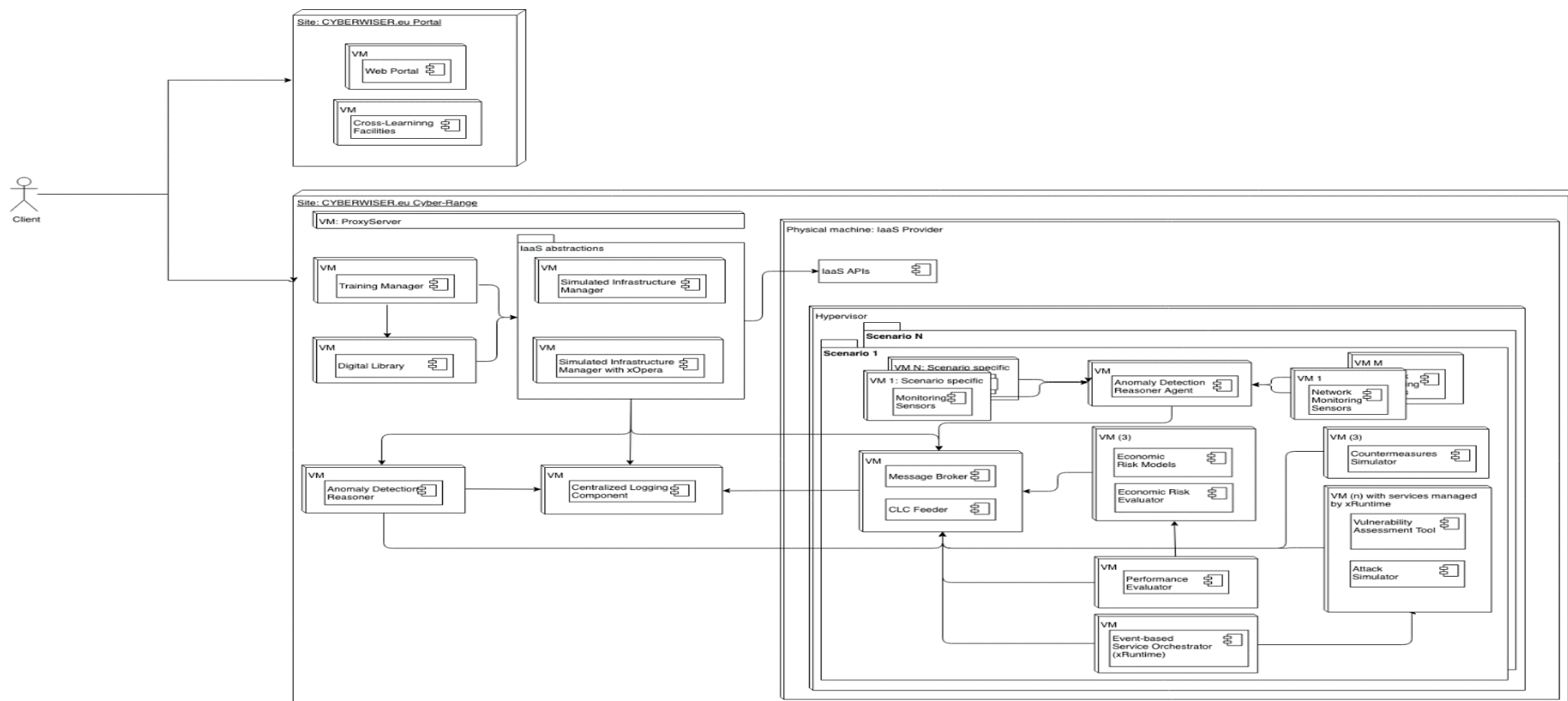


Figure 4. Deployment diagram

The first part consists of the web portal and cross learning facilities which are hosted together into two distinct virtual machines. These components operate stand-alone and are therefore also deployed as such.

The second part consists of the components required for designing and deploying individual training scenarios to be used by the clients. These components are the TM, SIM, DL, Pluggable Deployment Manager (xOpera). They are deployed on one or more virtual machines. The SIM is planned to be provided in two different versions. The baseline SIM implementation, which will support OpenNebula as the IaaS provider, and secondly a SIM that augments the Pluggable Deployment Manager (xOpera) in order to provide cross-IaaS support, initially focusing on OpenStack. Next to these machines we also have the ADR and the CLC deployed here on a set of virtual machines. They are deployed externally, out of the context of the individual scenarios to ensure the central storage encompassed all scenarios and the heavy ADR component only has to be deployed once. All these virtual machines for designing and deploying the training scenarios may themselves also be deployed on the OpenNebula Kernel-based Virtual Machine (KVM) hypervisor although this is not a requirement.

Finally, the third part consists of components that are part of each individual training scenario deployment. Specifically, the MB with the CLC Feeder, ADR Agents, Monitoring Sensors, CS, AS, VAT, ERE with its ERM, PE and the EBSO. These components are deployed alongside the other virtualized resources specific for each scenario where they gather information and allow (automated) controls for managing the training scenario as clients are using it. It is important to realize that these components are deployed multiple times, potentially in parallel for each running training scenario.

### 4.3 Data view

This section describes the Data view of the CYBERWISER.eu Platform. Understanding and modelling the data model is an essential part of any systems design and gives insight in how the various functions of the system will be supported in terms of data stored and exchanged within the system.

#### 4.3.1 Data model

The data model<sup>3</sup> for the platform is modelled on the different levels of abstraction. At the highest level of abstraction, only the main packages that have been identified are shown. At the next level each of the packages is detailed in terms of their main entities. At the most detailed level, the individual attributes and, where possible, their datatypes have been detailed as well. It is important to understand that each level adds additional details to the data model. This may result in the situation where certain entities are shown on a detail level diagram, that do not appear on a higher-level diagram. This usually indicates that such an entity is not considered the most important entity of that package. Alternatively, an entity may appear at a more detailed level to describe certain attributes of a relation between two entities at a higher level. Due to the introduction of these attributes, the relation is changed to an entity in between the related entities of the relation.

##### *Package level data model*

The data model has also been divided into a number of packages in order to reduce complexity and provide a logical grouping. The following diagram depicts the four identified packages at a high level of abstraction.

<sup>3</sup> Figures in this section are Entity Relationship Diagrams (ERD)



Figure 5. Package level ERD

The main packages that we have identified are 1) Scenario; describing all information related to the virtualized training scenarios in the cyber-range, 2) Monitoring; describing all information related to the monitoring of events in a running training scenario 3) Actions; describing all information related to executable actions during a training scenario and 4) Course; describing the information related to the overall training courses, excluding the cyber-range itself.

#### *Entity level data model*

Zooming in one level from the Package level data model, the individual main entities within each package are identified. The following diagrams show the main entities of each of the packages that have been identified.

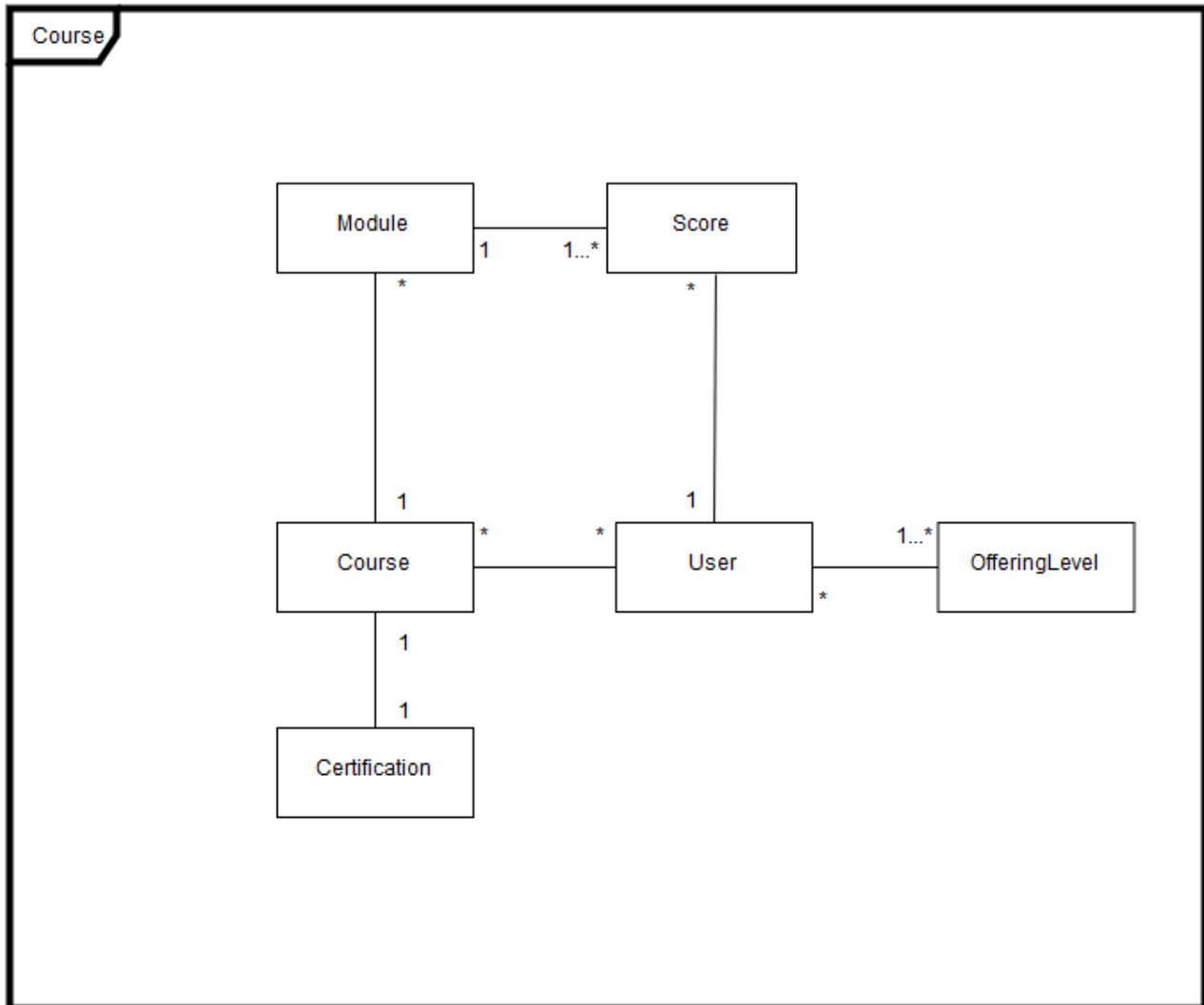


Figure 6. Entity level ERD – Course package

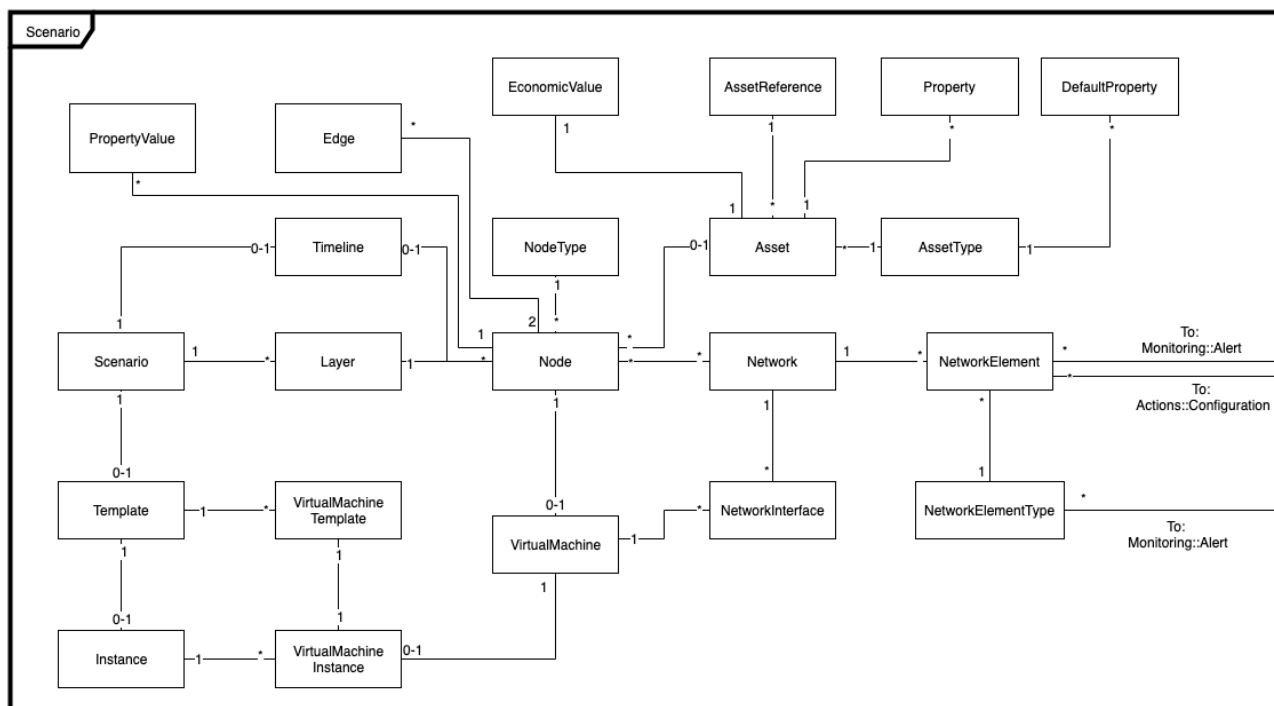


Figure 7. Entity level ERD – Scenario package

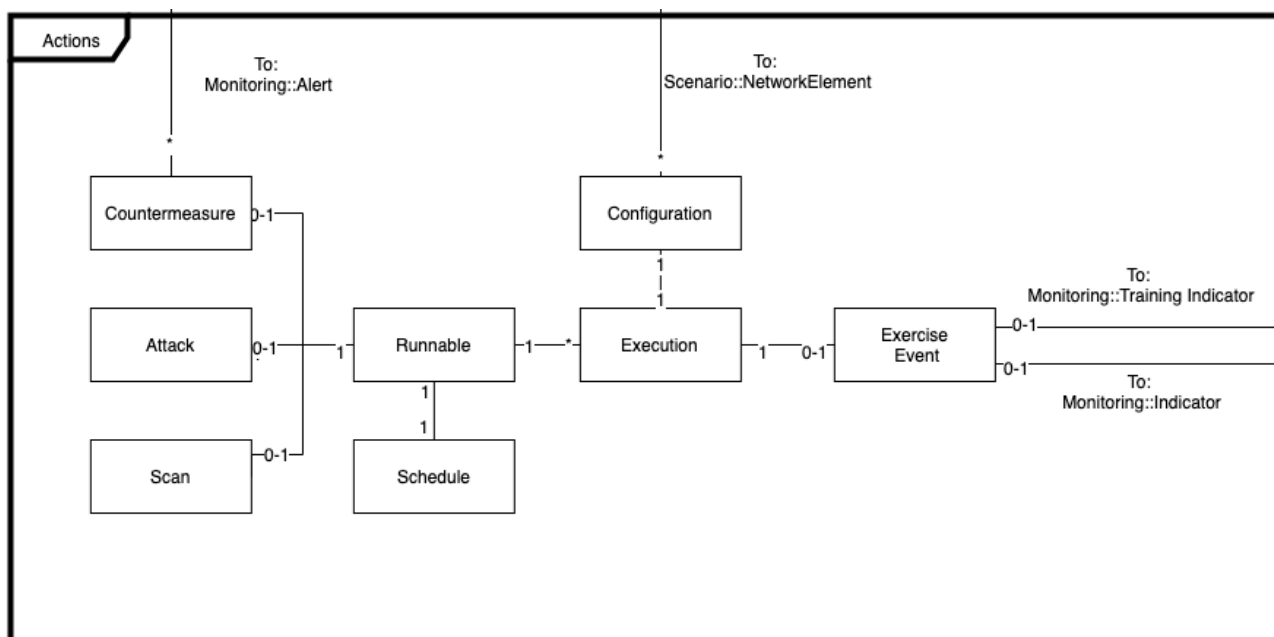


Figure 8. Entity level ERD – Actions package

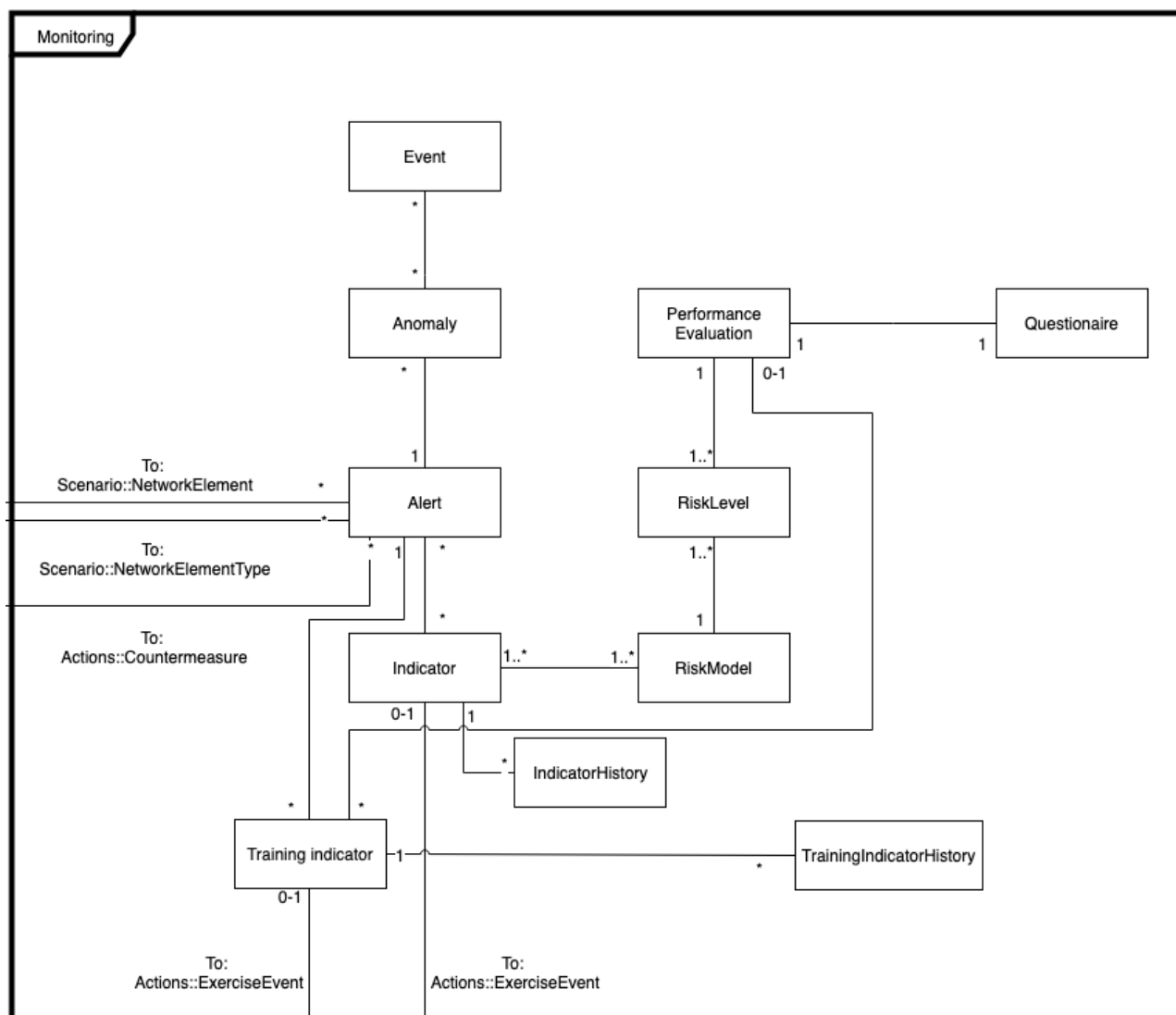


Figure 9. Entity-level ERD – Monitoring package

#### Attribute level data model

After the initial analysis that resulted in the higher-level entity ERD diagrams. The following detailed attribute ERD diagrams were identified for each of the packages.



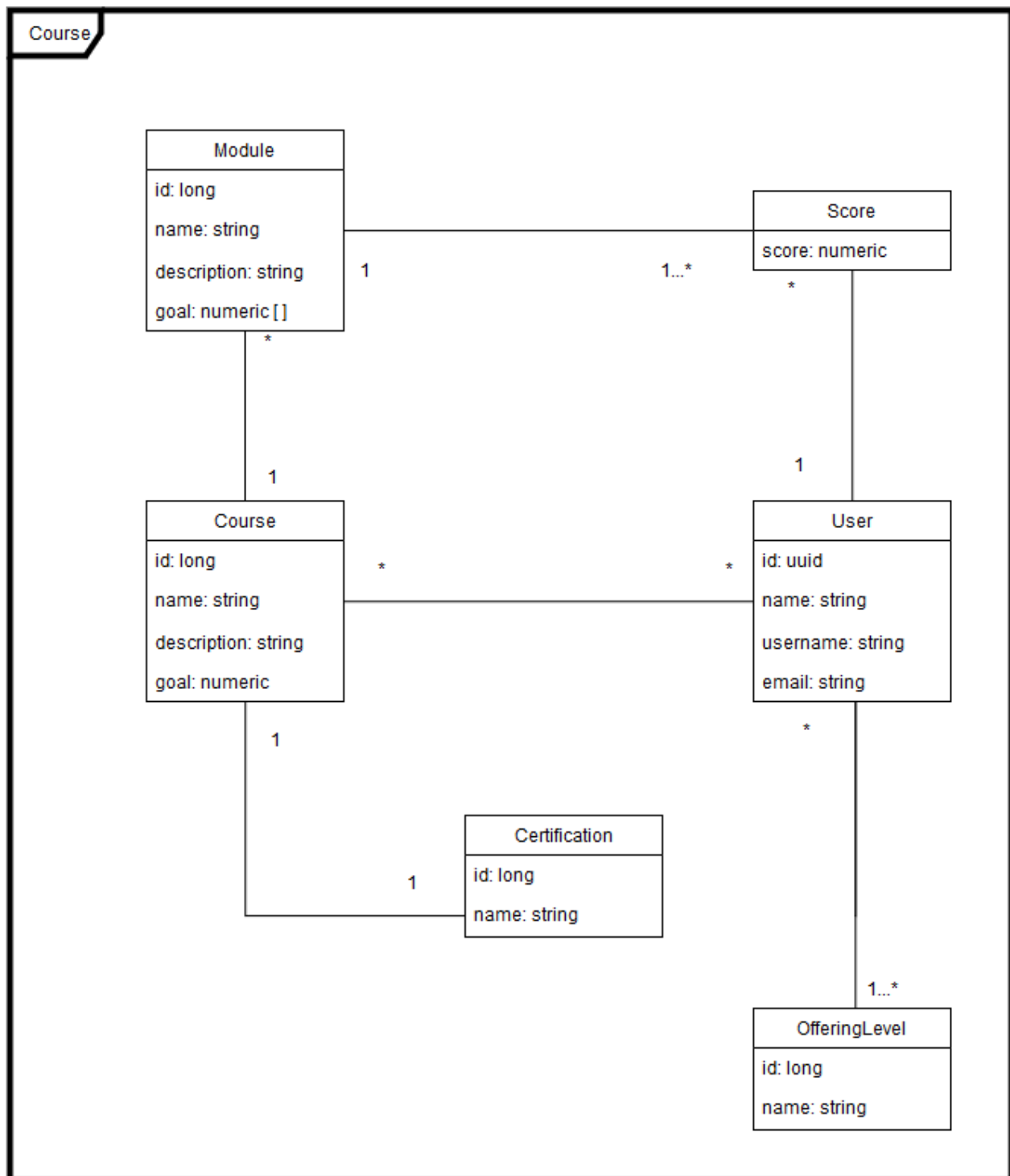


Figure 10. Attribute ERD – Course package

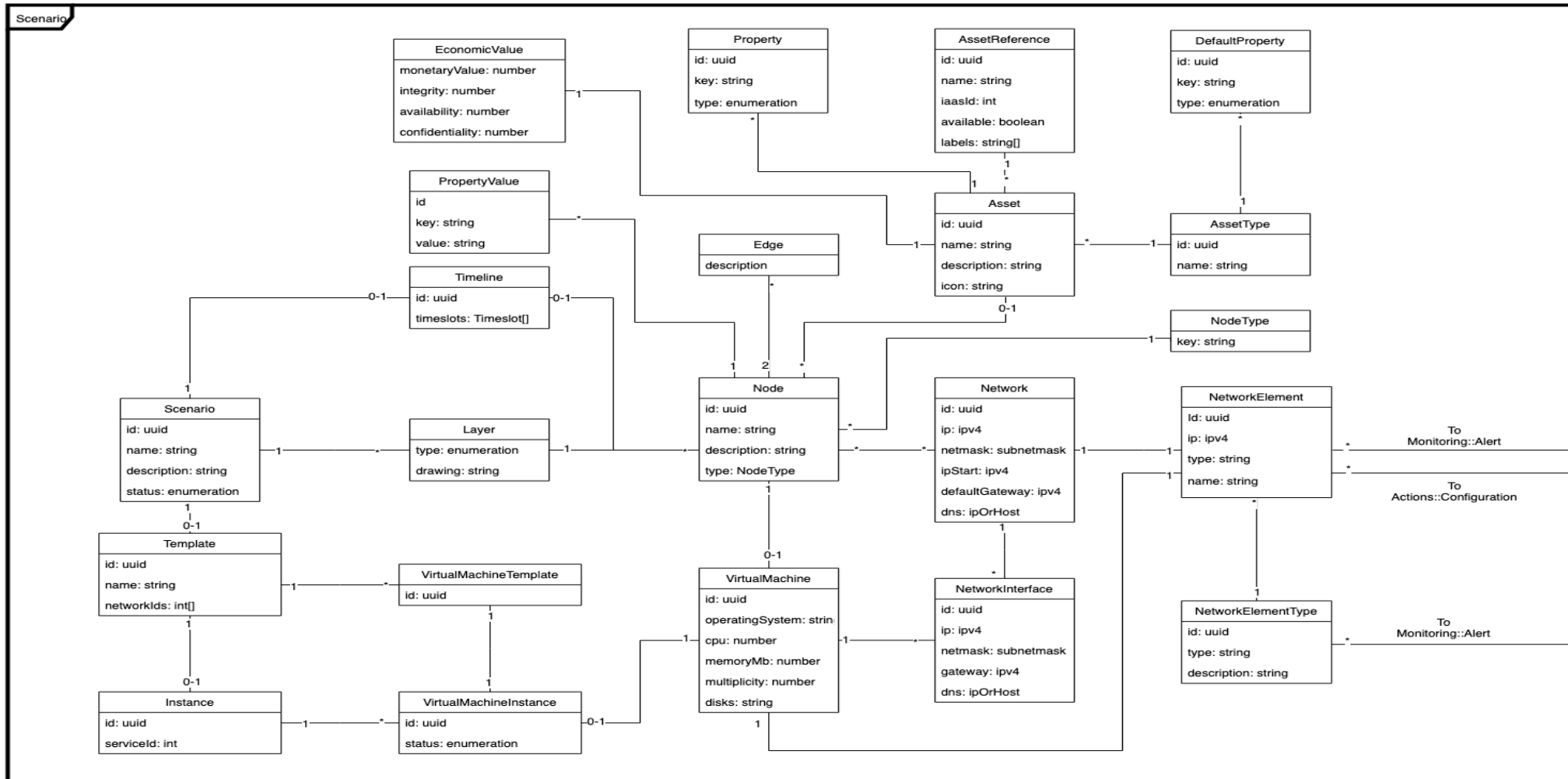


Figure 11. Attribute ERD – Scenario package

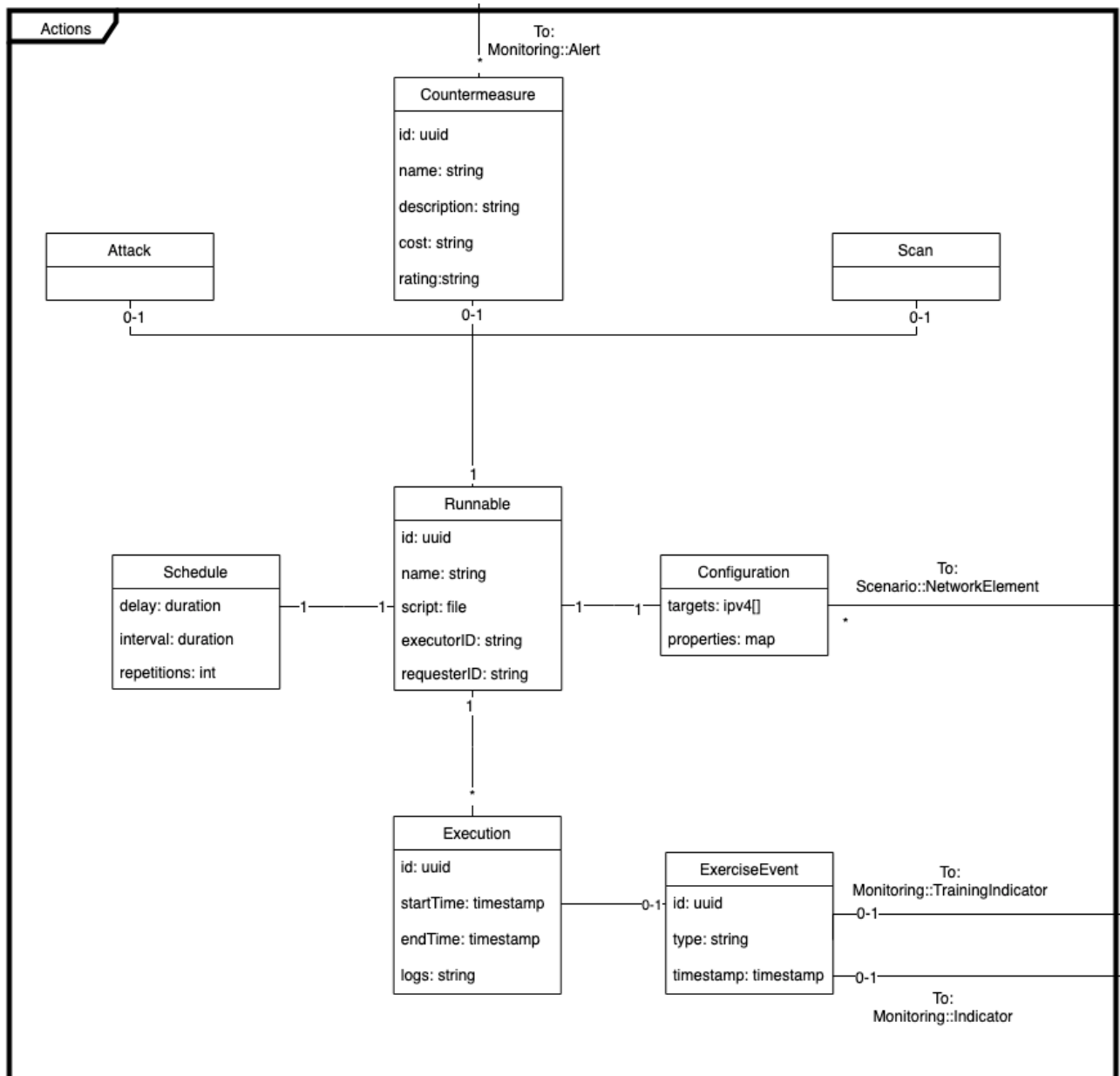


Figure 12. Attribute ERD –Actions package

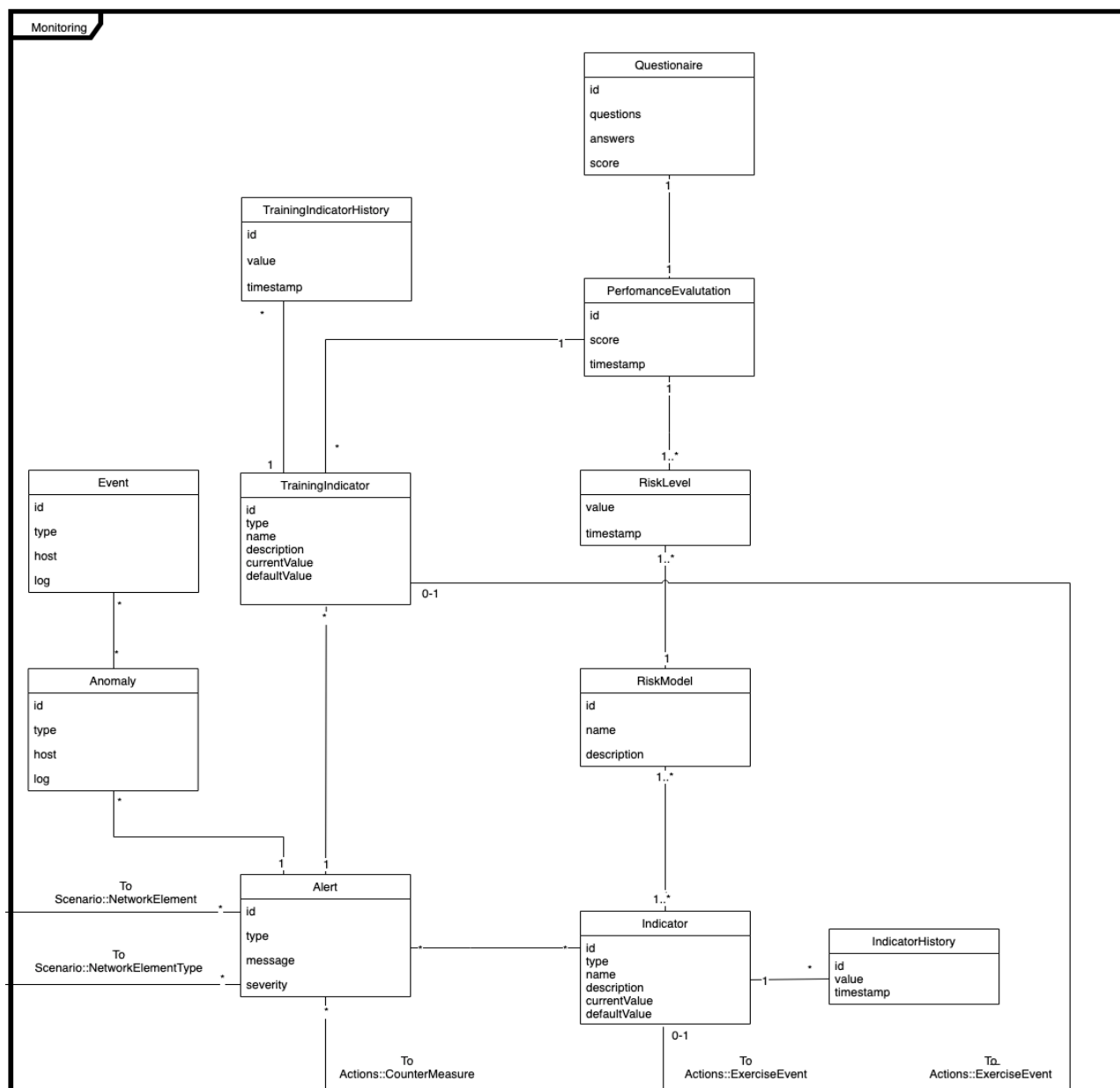


Figure 13. Attribute ERD – Monitoring package

#### 4.3.2 Data flows

While the data models detailed in the previous section determine which data is being created and used in the platform, it says nothing about the flow of information between the different components that make up the CYBERWISER.eu platform. The following diagram does exactly that. All the data flows between the different components are detailed, giving a better understanding of the overall functionality and at a high level, the data being exchanged across the interfaces of these components.

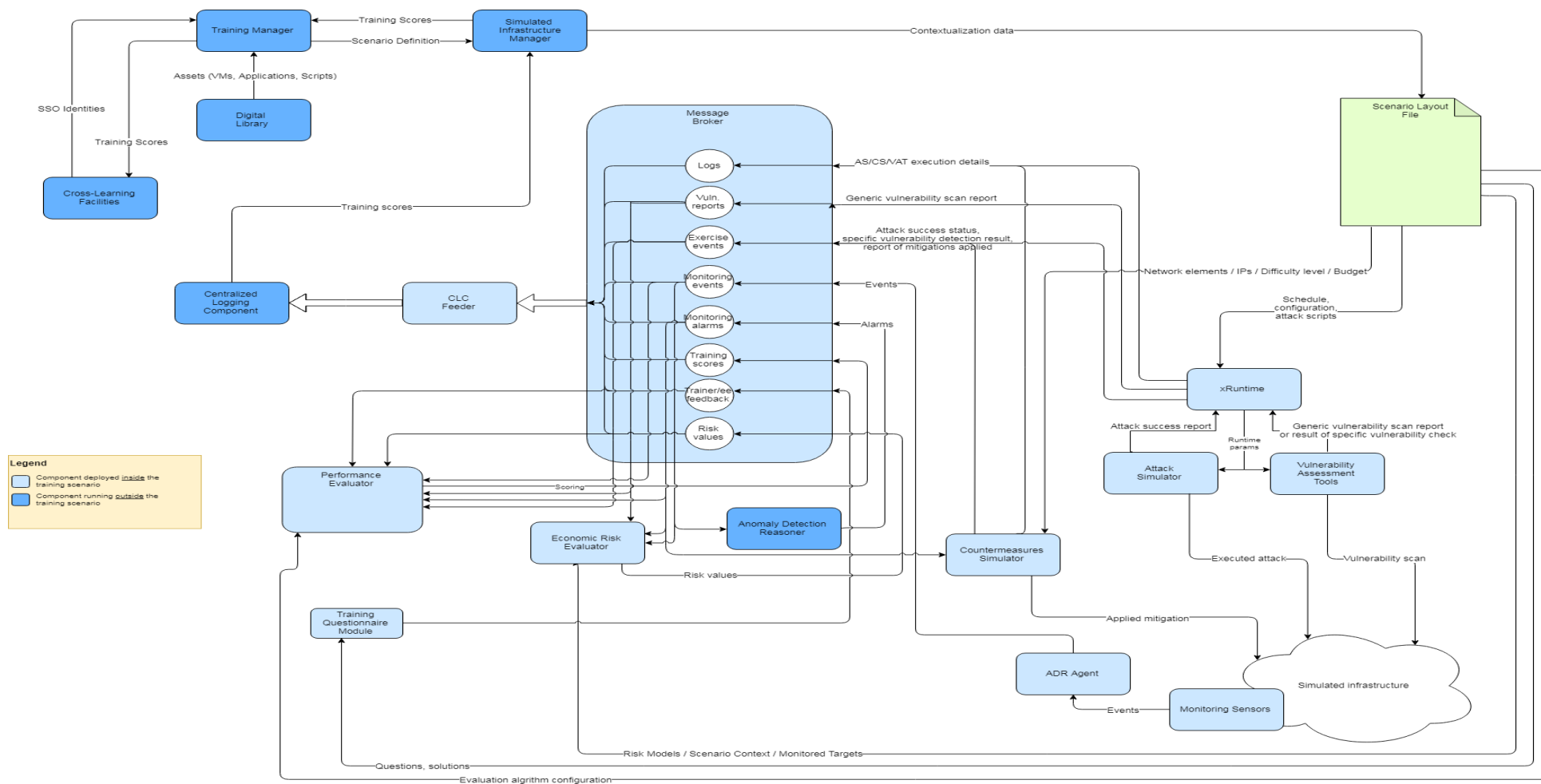


Figure 14. Data Flow Diagram<sup>4</sup>

<sup>4</sup> The Monitoring Sensors are part of the monitored infrastructure

The rounded rectangles represent the individual components of the platform, the arrows represent the data being exchanged from one component to another. The document in the top right corner labelled “Scenario Layout File” is a specific file that is made available by the SIM to virtual machines running in the scenario. It contains all relevant information for other components of the platform that are running inside the scenario itself.

#### 4.4 Technology view

This section presents the choice of technologies used for development of the CYBERWISER.eu Platform’s building blocks and interfaces between them. Table 6 summarizes the main programming languages, frameworks and other technologies in CYBERWISER.eu components. Please note that programming languages and frameworks are only stated for the components that required development effort from the CYBERWISER.eu consortium. In case an existing project or product fully supports a given CYBERWISER.eu component and merely needed to be properly configured and packaged, it is listed under the main technologies column, while the implementation details of such components are omitted. Thus, the second column of Table 6 is only relevant for the technology that was newly developed within CYBERWISER.eu.

Component	Main programming language(s) and framework(s)	Main technologies
<b>Event-Based Service Orchestrator (xRuntime)</b>	Go, JavaScript, Node.js	Docker, AMQP, JSON, RabbitMQ, OpenStack Swift, NFS
<b>Attack Simulator</b>	Bash, Python, Metasploit, Node.js, AngularJS, Swagger	Docker, xRuntime, JSON, AMQP
<b>Vulnerability Assessment Tools</b>	Python, Node.js, AngularJS, Swagger	Docker, xRuntime, OWASP ZAP, w3af, Cscan, JSON, AMQP
<b>Pluggable Deployment Manager</b>	Python	Ansible, TOSCA, YAML
<b>Centralized Logging Component</b>		Elasticsearch, ElasticHQ, Kibana, JSON
<b>Centralized Logging Component Feeder</b>	Python	AMQP
<b>Message Broker</b>	Python	RabbitMQ, AMQP, Docker, JSON
<b>Training Manager, Simulated Infrastructure Manager, Digital Library</b>	Java, TypeScript, Spring Boot, Angular	MongoDB
<b>IaaS</b>		OpenNebula
<b>Anomaly Detection Reasoner</b>	Python, Flask	OSSIM, Java, Spring Boot, Esper, Apache Storm, Nimbus, Apache ZooKeeper, AMQP, MariaDB, PHP, REST, Docker, JSON
<b>ADR Agent</b>	Python, JSON	Docker, AMQP, Syslog
<b>Monitoring sensors</b>	Python	OSSEC, Suricata, Cowrie, Nagios, Syslog, JSON
<b>Economic Risk Evaluator</b>	Python, Django, Django REST Framework,	AMQP, Docker, Docker-compose, PostgreSQL, Unicorn, Nginx, JSON
<b>Performance Evaluator</b>	Django REST Framework, Python, Weasyprint	AMQP, Docker, PostgreSQL
<b>Countermeasures Simulator</b>	Python, Django, Django REST Framework	AMQP, Docker, PostgreSQL, JSON
<b>Economic Risk Models</b>	R	CORAS

Component	Main programming language(s) and framework(s)	Main technologies
<b>Web Portal</b>	PHP, JavaScript, jQuery.js, Bootstrap.js	Drupal, JSON
<b>Cross-Learning Facilities</b>	PHP, JavaScript, jQuery.js, Bootstrap.js	Moodle, Drupal, JSON

Table 6. A summary of main technologies used in CYBERWISER.eu components

The subsections that follow describe the technologies used by each individual component in greater detail.

#### 4.4.1 Event-based Service Orchestrator (xRuntime)

The development name of EBSO is xRuntime. Its main part is a composition of various software components built on top of different software stacks: *the event processing logic and scheduling service*, and *the Docker interface*. They communicate with one another via the AMQP protocol. The xRuntime also requires some additional *infrastructural elements* that support its workflows.

- The CYBERWISER.eu exercise event processing logic and scheduling service are comprised in a single Go application. Go was chosen for its simplicity and robustness when it comes to implementation of concurrent logic which is essential for efficient task scheduling. The application understands requests formulated as JSON. Beside the Go standard library, it depends only on the library with implementation of the AMQP protocol and RabbitMQ extensions<sup>5</sup>, needed for communication with the Docker interface part of xRuntime.
- The Docker interface is a JavaScript application built with Node.js<sup>6</sup> web application framework. It too, expects requests in JSON format and responds with status messages in JSON, with incoming requests and outgoing responses carried by the AMQP protocol. Intuitively, as the Docker interface's main task is management of Docker containers, it depends on a Node.js client library for Docker, and requires presence of the Docker engine on the machine hosting the containers. As the Docker interface was extended with various functionalities for easier management of input files for the created Docker containers, and for persistence of their standard output or arbitrary files within these containers, it also depends on Node.js clients for OpenStack Swift<sup>7</sup> and NFS.

The infrastructural elements of xRuntime include an NFS server for file sharing across different hosts and OpenStack Swift for object storage. In addition, xRuntime relies on a RabbitMQ server for internal AMQP communication between the two parts described above. Note that this is not the same RabbitMQ instance as the one behind the CYBERWISER.eu's MB.

#### 4.4.2 Attack Simulator

As described in Section 5.12, the AS does not function as a standalone component but is represented by 1.) the attack tools and attack scripts which may be managed through 2.) a frontend web application that communicates with 3.) the EBSO backend.

<sup>5</sup> <https://godoc.org/github.com/streadway/amqp>

<sup>6</sup> <https://nodejs.org/en/about/>

<sup>7</sup> <https://wiki.openstack.org/wiki/Swift>



AS contains three attack tools, working as interpreters for the attack scripts in different scripting or programming languages: Bash<sup>8</sup>, Python<sup>9</sup>, and Metasploit<sup>10</sup>. The attack tools are packaged as Docker<sup>11</sup> images with the relevant interpreter and possibly other required software and libraries installed:

- Bash attack tool is a simple Docker image containing only the Bash interpreter and basic Linux networking tools like cURL<sup>12</sup>.
- Python attack tool is a Docker image containing Python 3.7.
- Metasploit attack tool is a Docker image containing the open-source (BSD-3-Clause license) Metasploit Framework and standard Metasploit modules that are included by default. It is able to run scripts written for the Metasploit console. The Metasploit Framework and its modules are written in Ruby<sup>13</sup>. Custom Metasploit modules for specific attacks can be added as well. In this case, they must also be written in Ruby and adhere to the standard interface format of Metasploit modules.

Attack scripts, which can be provided by both trainers and trainees, may introduce additional technologies, communication standards, etc., not listed above or included in the attack tools. Further development of attack scripts for CYBERWISER.eu training scenarios will determine whether additional software needs to be added to any of the attack tools.

It is anticipated that the implementation of the web application for managing attacks (Attack Configurator) will rely on JavaScript frameworks, likely AngularJS<sup>14</sup> and Node.js. Swagger<sup>15</sup> will be used to aid the development of the REST API. For internal communication with the EBSO, a library with implementation of the AMQP protocol will be used.

The attack tools are instantiated as Docker containers by the EBSO that interfaces the Docker engine. The input and output files are transferred by mounting appropriate directories into the file system of the attack tools, and via NFS in case of communication between different virtual machines. Please refer to Section 4.4.1 for details concerning the technology supporting this workflow.

#### 4.4.3 Vulnerability Assessment Tools

The operation of VAT varies depending on the type of vulnerability scan being executed.

As detailed in Section 5.9, vulnerability detection scripts detect very specific vulnerabilities. These scripts work in a similar manner as attack scripts and are also executed in the same Docker images (attack tools) and orchestrated by the EBSO, thus the technologies used in this case are the same as in the case of AS. Note that despite similarities in implementation, custom vulnerability detection scripts differ from the attack scripts in their function: instead of actively launching attacks, they passively determine the presence of vulnerabilities, minimizing (to the extent possible) the changes made and traces left in the target machine.

To detect generic vulnerabilities (detectable using the popular scanning engines and their common knowledge-bases), VAT relies on a generic suite of vulnerability scanners. The suite consists of several standard scanning tools, connected and orchestrated by the open-source Cscan framework<sup>16</sup>, written mostly in Python (and

<sup>8</sup> <https://www.gnu.org/software/bash/>

<sup>9</sup> <https://www.python.org/>

<sup>10</sup> <https://www.metasploit.com/>, <https://github.com/rapid7/metasploit-framework>

<sup>11</sup> <https://www.docker.com/>

<sup>12</sup> <https://curl.haxx.se/>

<sup>13</sup> <https://www.ruby-lang.org/>

<sup>14</sup> <https://angularjs.org/>

<sup>15</sup> <https://swagger.io/>

<sup>16</sup> <https://github.com/infobyte/faraday/tree/master/scripts/cscan>

licensed with GNU GPL v3). The scanning engines currently integrated are OWASP ZAP<sup>17</sup> (Apache 2 license) and w3af<sup>18</sup> (GNU GPL v2 license). Their output is gathered and merged into a JSON object, passed to and understood by other CYBERWISER.eu Platform components. The generic suite of scanners is packaged as a Docker image and also orchestrated by the EBSO.

It is anticipated that the implementation of the web application for managing vulnerability scans (Scan Configurator) will rely on JavaScript frameworks, likely AngularJS and Node.js. Swagger will be used to aid the development of the REST API. For internal communication with the EBSO, a library with implementation of the AMQP protocol will be used.

#### 4.4.4 Pluggable deployment manager (xOpera)

The development name of the Pluggable Deployment Manager is xOpera. As detailed in Section 5.3.1, the orchestrator and deployment manager is composed of four parts: the orchestration engine (also referred to as orchestrator), a REST API service, CSAR (Cloud Service Archive) compiler and TOSCA (Topology and Orchestration Specification for Cloud Applications) libraries of components.

The xOpera also offers a command-line tool called *opera*, which is, just like the orchestration engine<sup>19</sup>, software written in Python 3 with Python standard library as its main dependency. The tool works with an Ansible<sup>20</sup> environment and a small set of Ansible modules, which are all Python-based and come with the installation of the orchestration engine. The orchestrator engine and the command-line tool are open-source, Apache 2.0 licensed works. The CSAR compiler, too, is a Python tool that will be released under the same license. The implementation and licensing details of the REST API service to control the orchestration engine are not yet settled at the time of writing, however it is likely that the service will also be a Python application hosted on top of a web serving engine such as Nginx<sup>21</sup>.

TOSCA Libraries of components are comprised of files in YAML format compliant with the TOSCA YAML Simple Profile v1.2 standard<sup>22</sup>. The libraries also include Ansible playbooks to implement specific lifecycle operations carried out during orchestration workflows (for instance create, start, or stop). Certain elements in the library require third-party Python modules or other tools to be installed in the orchestrator's environment (e.g., OpenStack client SDK, AWS client tools, etc.). The TOSCA libraries may include definitions of base types, connectors to target IaaS-es, or custom types derived for the target application, i.e. the application it deploys and orchestrates (in the case of CYBERWISER.eu, target applications are training scenarios). Base types and IaaS connectors will likely be open-source and permissively licensed.

Technology-wise, xOpera poses no constraints on the target application. However, the orchestrator currently supports only the Ansible executor for the application's lifecycle operations, meaning that Ansible playbooks need to be provided for every resource managed by xOpera.

#### *TOSCA Translator*

The programming language and libraries to be used for implementation of the TOSCA Translator are undecided at the time of writing. Nevertheless, as the component will translate training scenario requests (represented as JSON files) into TOSCA service templates (YAML files), it depends on at least these three technologies – JSON, YAML and TOSCA.

<sup>17</sup>[https://www.owasp.org/index.php/OWASP\\_Zed\\_Attack\\_Proxy\\_Project](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project)

<sup>18</sup><http://w3af.org/>

<sup>19</sup> <https://github.com/xlab-si/xopera-opera>

<sup>20</sup> <https://www.ansible.com/>

<sup>21</sup> <https://www.nginx.com/>

<sup>22</sup> <https://docs.oasis-open.org/tosca/TOSCA-Simple-Profile-YAML/v1.2/os/TOSCA-Simple-Profile-YAML-v1.2-os.html>

#### 4.4.5 Message Broker

The MB is an instance of RabbitMQ server, an open-source message brokering solution implementing the AMQP protocol. RabbitMQ has support for TLS (SSL). All client components will communicate with this message broker using secure channels. To establish secure connections the server needs a CA (Certificate Authority) certificate and a certificate/key pair; and the clients shall use a certificate/key pair. The certificates and keys are created by using OpenSSL which is an open source toolkit.

RabbitMQ server is licensed under Mozilla Public License 1.1.

##### *CLC Feeder*

The CLC Feeder is a microservice written in Python 3. As its purpose is to collect messages from the MB (a RabbitMQ instance), transfer and persist them to the CLC (backed by Elasticsearch), its dependencies are Python AMQP library with RabbitMQ extensions and Elasticsearch client.

#### 4.4.6 Centralized Logging Component

The data coming into the CLC is stored in the form of JSON documents in Elasticsearch<sup>23</sup>, a fast, distributed search and analytics engine exposing a RESTful API. To allow user-friendly display of and insight into the data stored in the CLC, Elasticsearch is connected to Kibana<sup>24</sup> service for navigating and visualizing the data. In addition, the deployment of CLC will also include an ElasticHQ<sup>25</sup> service supporting web-based monitoring and management of the Elastic instance.

All parts of Elasticsearch, Kibana and ElasticHQ used in CYBERWISER.eu are open-source and licensed under the Apache 2.0 license.

#### 4.4.7 Digital Library, Training Manager, Simulated Infrastructure Manager

The DL, TM and the SIM are all built on the same technological stack. Frontend components are implemented with Typescript and Angular and backend components are implemented using Java and the Spring Boot family of libraries. Both frontend and backend application components use a large number of open source libraries all which are licensed such that they remain suited for usage in a commercial context. For persistence of data all components make use of a NoSQL, MongoDB document database.

Although all components are suitable for cross-platform deployment, the main target operating system is Linux, specifically Ubuntu.

The interfaces between these components themselves and between these components and other components of the CYBERWISER.eu platform are all based on RESTful principles where data is exchanged in JSON format. One exception exists though. This is the SSO interface between the web portal SSO provider and the TM. The SSO solution is based on the OAuth2, OpenID connect protocol.

#### 4.4.8 Anomaly Detection Reasoner

Section 5.11 provides details about the ADR. Most of this information is obtained from the publication referred below<sup>26</sup>. The reader can refer to such publication, in which this information is elaborated, for further details. The ADR is based on the XL-SIEM solution brought by ATOS to the project, available with license GPLv2. It is built on top of the Open Source SIEM called OSSIM<sup>27</sup>. It can run on Ubuntu 16. The technologies are enumerated in the bulleted list below:

<sup>23</sup> <https://www.elastic.co/products/elasticsearch>

<sup>24</sup> <https://www.elastic.co/products/kibana>

<sup>25</sup> <https://www.elastichq.org/>

<sup>26</sup> CIPSEC deliverable D2.2 "CIPSEC Unified Architecture Internal Release"

<sup>27</sup> <https://www.alienvault.com/products/ossim> (latest accessed on 19/06/2019)

- OSSIM as a baseline Open Source SIEM
- A set of Java processes that are integrated in the ADR.,
  - Such processes include the high-performance correlation engine Esper<sup>28</sup> library, packaged in a topology to be deployed in an Apache Storm<sup>29</sup> cluster.
- Apache Storm is an open source distributed real-time computing system for processing large volumes of data.
- Apache Zookeeper<sup>30</sup> running along with Apache Storm to support scalability and distributed real-time processing of events. Apache Zookeeper provides distributed synchronisation through the Storm cluster, while maintaining centralized configuration information.
- AMQP to implement the communication between the ADR and the ADR Agent through the MB
- MySQL relational database to store both events gathered by the agents and alarms generated by the server.
- PHP to implement the user interface.
- A REST API to access the application data.

#### Agent

The ADR Agent is implemented in Python and can run on Ubuntu 16. It has two interfaces: one to the ADR itself which uses AMQP through the MB, and the other one to the Monitoring Sensors, to receive the information they send, that uses Syslog<sup>31</sup>. It is available with license GPLv2.

#### 4.4.9 Monitoring Sensors

The Monitoring Sensors communicate with the ADR Agent by means of Syslog, as stated in section 4.4.8. The following monitoring technologies are considered: Suricata as a Network Intrusion Detection System, OSSEC as a Host Intrusion Detection System, the Cowrie Honeypot and Nagios for Network Monitoring. The three first were successfully used in the WISER Project and the Consortium therefore counts on previous experience to incorporate them. The last one is very popular and likely to be useful in the context of the exercises. During the implementation new candidates may arise to be integrated, what means that the list shown here may change.

#### 4.4.10 Economic Risk Evaluator

The ERE is based on the RAE solution brought by ATOS to the project. It is an evolution of the RAE application with an independent dashboard application and adapted for a production environment with the PostgreSQL database, the Unicorn server and the Nginx as a reverse proxy server.

The technologies used in the implementation of the Economic Risk Evaluator can be summarized as follows:

- Django as a framework leveraged for implementing the user interface.
- Python to code the intelligence running in the backend.
- Django REST Framework to implement the endpoints through which the external components or user can access to the database.

<sup>28</sup> <http://www.espertech.com>

<sup>29</sup> <http://storm.apache.org>

<sup>30</sup> <https://zookeeper.apache.org>

<sup>31</sup> <https://tools.ietf.org/html/rfc5424>

- PostgreSQL is the technology on which the ERE database is based and it is available for Django to have direct access to the information.
- Docker allows the containerization of the ERE elements, easing automatic deployments.
- Docker-compose orchestrate the docker containers that compose the ERE application.
- Gunicorn is the server supporting the ERE dashboard application.
- Nginx is the reverse proxy that collect the requests to the ERE
- AMQP is the interface used to collect the information from the MB, originally generated by the ADR and its Agent, the VAT.
- The interface provided by the ERE to the PE is based on RESTful and data is exchanged in JSON format. JSON format is also managed for the incoming RabbitMQ messages.

### Models

The ERE evaluates the economic risk exposure in the running scenario in real-time. The calculation of the risk is based on cyber risk models. These models are first created in terms of graphical risk models using the CORAS risk modelling tool. The CORAS tool is a diagram editor based on Eclipse EMF/GMF and it is designed to support on-the-fly modelling using all kinds of CORAS diagrams, including CORAS threat diagrams. Once a model is created and validated, a methodology can be followed to derive the corresponding calculation algorithm. As an intermediate step, and to ease the task, the user can use some kind of pseudocode creation technique to better understand the algorithm to implement. Then any kind of language can be used to actually implement the algorithm, nevertheless the ERE is prepared to execute algorithms written in the R language which are executed through the R script binary<sup>32</sup>.

#### 4.4.11 Performance Evaluator

The PE is a pure backend component developed in Python and using the Django REST Framework to obtain the needed information from a PostgreSQL database. The evaluation reports are produced using Weasyprint<sup>33</sup>. It communicates with the MB by means of the AMQP protocol. It is containerized using Docker to ease the automation of the deployments.

#### 4.4.12 Cross-Learning Facilities

The Cross-Learning Facilities integrate:

- A set of Workspace areas programmed using PHP language and developed using Drupal<sup>34</sup> version 7, GNU GENERAL PUBLIC LICENSE, version 2, June 1991.
- A Learning Content Management System (TRUST-LCMS) based on Moodle 3.5.4, GNU GENERAL PUBLIC LICENSE, Version 3, June 2007.
- For the operating system and the programming language the distribution is PHP 7.1.30-1+ubuntu16.04.1+deb.sury.org+1 (cli).

<sup>32</sup> To learn more about the R language, the reader can consult WISER Deliverable D3.2 entitled “Cyber risk modelling language and guidelines, preliminary version”

<sup>33</sup> <https://weasyprint.org/>

<sup>34</sup> <https://www.drupal.org/>

A software called iSpring <sup>35</sup>is used to transform the Training Material into SCORM so that can be read by Moodle<sup>36</sup>.

#### 4.4.13 Web Portal

The web portal hosted at [www.cyberwiser.eu](http://www.cyberwiser.eu) has been programmed using PHP language and developed using Drupal<sup>37</sup> version 7, GNU GENERAL PUBLIC LICENSE, version 2, June 1991. For the operating system and the programming language the distribution is PHP 7.1.30-1+ubuntu16.04.1+deb.sury.org+1 (cli).

The module Organic Groups has been used and customised for the per organisation and management of the different groups, to share the different workspace areas and manage the interactive communications.

To guarantee the integration of the Cross-Learning facilities, and therefore the integration of the other CYBERWISER.eu components, the web portal host a SSO mechanism based on OpenID Connect module that provides a pluggable client implementation for the OpenID Connect protocol.

#### 4.4.14 Countermeasures Simulator

The intelligence of the Countermeasure Simulator is developed using Python. The business tier communicates with the internal PostgreSQL database using Django REST Framework. The presentation layer is developed using Django. The component is containerized using Docker to ease its automatic deployment. It communicates with the ADR by using the MB basing on AMQP.

<sup>35</sup> <https://www.ispringsolutions.com>

<sup>36</sup> <https://moodle.org>

<sup>37</sup> <https://www.drupal.org/>



## 5. Detail of the building components

### 5.1 Web Portal

The web portal is the single access point to the CYBERWISER.eu Platform and all training resources for both trainers and trainees thanks to a single Login/Registration page.

The Login/Registration page will grant users access the Cross-Learning Facilities and the other CYBERWISER.eu components according to the permission level granted to the user. This means the user will have different rights whether he's a trainer or trainees, if he's part of a certain pilot or another and if he chooses a certain CYBERWISER.eu Offering Level or another.

The web portal guarantees the integration of the Cross-Learning facilities, and therefore the integration of the other CYBERWISER.eu components, through a SSO mechanism based on OpenID Connect module that provides a pluggable client implementation for the OpenID Connect protocol.

In the coming months the website will be enriched with dedicated sections about the CYBERWISER.eu platform which will explain the different Offering Levels available, their features and benefits from a trainee's perspective.

### 5.2 Cross-Learning Facilities

Once a user is logged in into the Web portal, a dedicated button is used to reach the Cross-Learning Facilities. The Cross-Learning Facilities are composed by dedicated Workspace areas that can be based on the CYBERWISER.eu Offering Level chosen by a particular group of user or on the specific group/s a user belong to (for example the Full Scale Pilots or the Open Pilots).

A user belonging to the PRIMER Offering Level will find in its Workspace area the e-learning platform and a file repository that can be used to store documents or files, together with other functionalities that are detailed in section 6.

A user belonging to the BASIC, INTERMEDIATE and ADVANCED level will find the same resources as above with the addition of the access to the cyber range platform.

### 5.3 Simulated Infrastructure Manager

The SIM is responsible for converting a designed training scenario into a template that can be instantiated by the IaaS provider. It controls the starting and stopping of training scenarios and provides access to the machines in the instantiated scenario. Figure 15 shows its internal components.



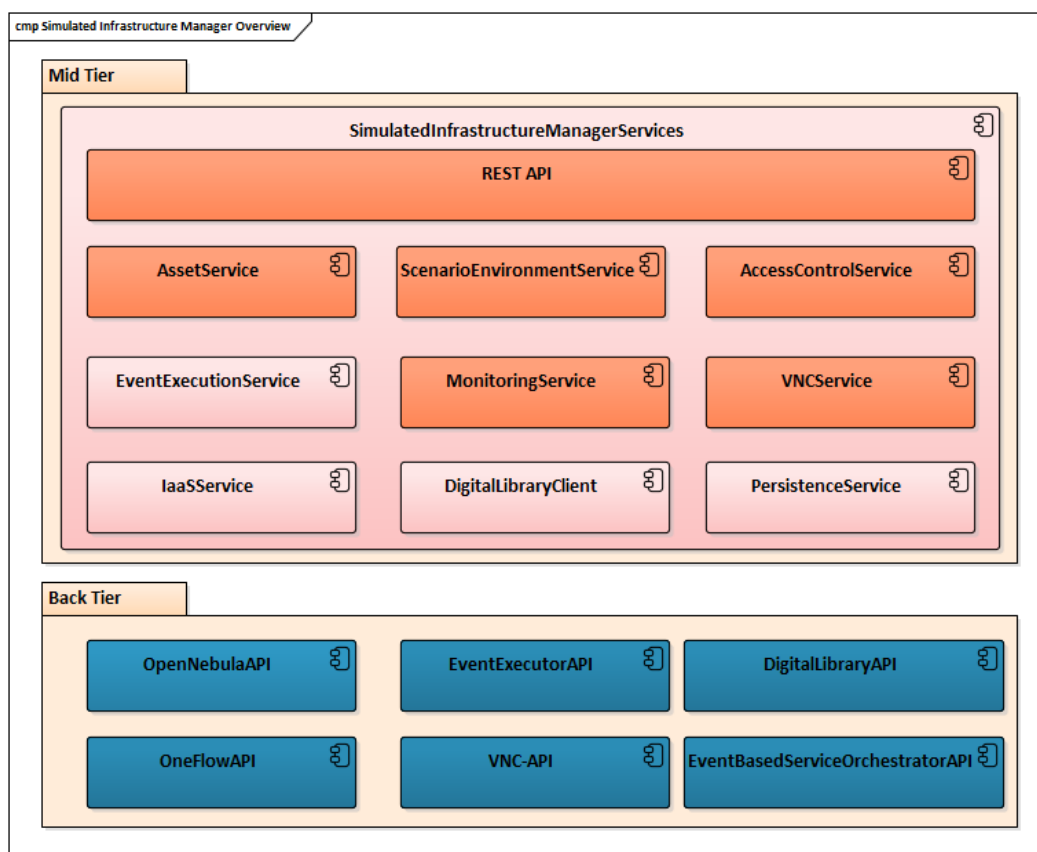


Figure 15. Simulated Infrastructure Manager components

The SIM's internal design is comprised of three different tiers. The Back Tier contains the data storage and external components that the SIM interacts with for managing the training scenarios running on the IaaS provider. The OpenNebulaAPI and OneFlowAPI, both provided by OpenNebula provide the necessary functionality for creating scenario templates consisting of virtual machine templates and service templates as well as providing the capability to create network subnets. The VNC-API also provided by OpenNebula allows the SIM to expose access to the instantiated machines in the form of VNC connections exposed over HTTP. The SIM also interacts with the DL to determine which elements a scenario is based on and to determine if all needed elements are available at the IaaS provider.

The Mid-Tier of the SIM contains all the necessary business logic to orchestrate the external APIs and provided a coherent set of functionalities to the TM and the end users of the training scenarios. This tier is decomposed into a set of services that each deal with a distinct subset of the overall functionality. The AccessControlService is responsible for providing proper authentication and authorisation on all request to the SIM. The ScenarioEnvironmentService is responsible for converting the training scenario request into a template and allowing instantiation of that template and subsequent management (deploy / undeploy / provide access). The AssetService provides functionality related to the management (creation, updates and removal) of the actual assets (virtual machines, scripts) that are available at the IaaS provider. The EventExecutionService is responsible for the scheduling and executing of events that are part of the definition of the training scenario. It interacts with the EBSO to realize these events. The MonitoringService is capable of processing relevant monitoring events from the running scenario in order to expose them to end users. The VNCSERVICE exposes the VNC connections to clients.

The Front Tier of the SIM is not pictured in this diagram because it is embedded in the frontend of the TM. Re-using the designed drawings of the scenario, functionality of the SIM is contextually shown in the user interface of the TM.

The SIM will be improved to allow for a high level of automation with respect to the virtual networking of the simulated environment. The SIM will support the integration of the overall performance evaluation of the trainees by integrating with the other components of the CYBERWISER.eu Platform. This requires the automatic deployment of monitoring sensors and other components and allowing them to exchange information with components that are deployed outside of the context of a running scenario. Additionally, the SIM will be improved to provide configuration information to each virtual machine running in the scenario about the information of the scenario itself (which machines are present, how can they be located and identified).

### 5.3.1 Pluggable Deployment Manager (xOpera)

The Pluggable Deployment Manager expands on the functionalities of SIM's ScenarioEnvironmentService and AssetService and can serve as an optional replacement of these services, allowing SIM to support simulated environments beyond OpenNebula. It aims to support a range of the IaaS providers such as OpenStack, OpenNebula and Amazon Web Services (AWS). When it is used, Pluggable Deployment Manager alters SIM's default means of scenario instantiation by providing an alternative path to a supported IaaS.

The Pluggable Deployment Manager aims to enable automated management of complete lifetime of training scenario instances in simulated environments. This includes the initial consistent and repeatable provisioning of the required virtual resources, deployment and configuration of the simulated environment for the training scenario, dealing with scaling up or down of the environment during its runtime, and final teardown. Internal design of the component is depicted in Figure 16.

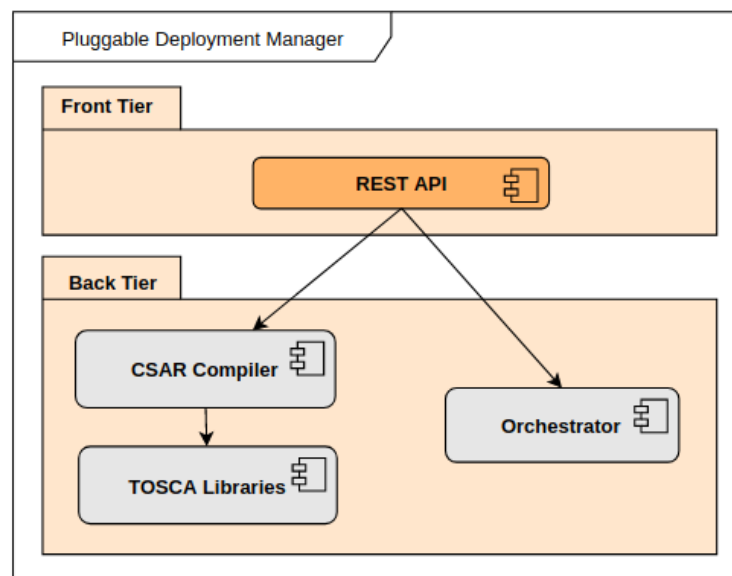


Figure 16. Pluggable Deployment Manager component diagram

SIM will interface with the Pluggable Deployment Manager via its REST API service. Most importantly, the API will expose endpoints for preparation and management of deployments. Specifically, it will allow for creation of a new deployment (instantiation of a training scenario), monitoring the status of the deployment workflow (checking whether the training scenario is ready), and tearing down the deployment (uninstantiation of a scenario). The basis of every deployment is a description of the simulated environment represented as a service template expressed according to TOSCA standard. Prior to actual deployment, the CSAR Compiler checks the validity of the service template and combines it with external dependencies referenced in the service template (for instance imported TOSCA files from TOSCA libraries of components, artefacts located in local or remote environments, etc.), producing a CSAR package as a result. The resulting CSAR package contains all the data required to create a new deployment and is used as an input to the orchestration engine (Orchestrator).

### TOSCA Translator

In CYBERWISER.eu, TOSCA is not the native format for the description of deployments associated with the training scenarios. Thus, the usage of Pluggable Deployment Manager introduces the need for an auxiliary component, the **TOSCA Translator**. The sole purpose of this component is translation of the training scenario descriptions used by SIM to equivalent TOSCA service templates consumable by the Pluggable Deployment Manager (instead of to OpenNebula virtual environment templates, as is the case when SIM is used without Pluggable Deployment Manager). Conceptually, the TOSCA Translator is a simple component that maps the data types used in training scenario requests to their TOSCA counterparts with the help of TOSCA libraries of components, as illustrated in Figure 17. Note that TOSCA libraries of components may include definitions of base types (for instance generic representations of VMs, virtual networks, security groups, etc.), connectors to target IaaS-es, or custom types derived for the target application, i.e. CYBERWISER-specific types (for instance CYBERWISER management network).

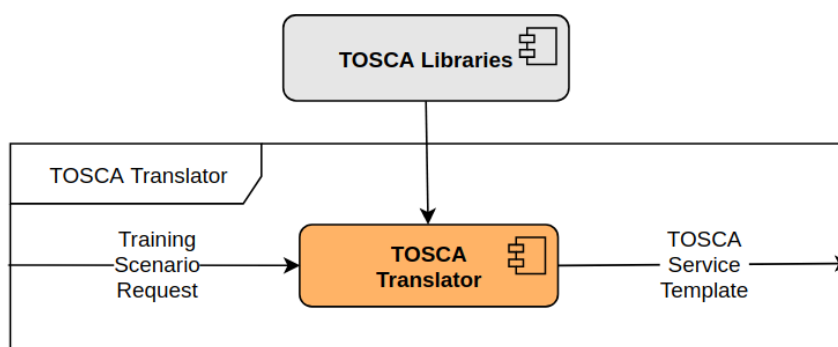


Figure 17. Conceptual design of the TOSCA Translator

## 5.4 Training Manager

The TM is responsible for providing an environment where users can design training scenarios and use those scenarios as part of training activities. A training activity can make use of multiple scenarios. The TM facilitates users that want to design a scenario by providing a four-layer Scenario Designer. It allows users to authorize other users to help design a scenario or take part in using the scenario once it has been instantiated. Fine grained access control rules allow us to expose only parts of the scenarios to individual users.

The TM as a whole will be further expanded by adding a timeline functionality. This timeline will be used for designers to plan out the different events that will occur during the execution of the scenario. These events will be automatically executed by the overall platform. In addition to this, the TM will allow for the definition of additional meta-data required for monitoring capabilities as well as monetary and security (integrity, availability and confidentiality) definitions of the scenario's virtual infrastructure to be used in the runtime evaluation of trainees.

### 5.4.1 Scenario Designer

The Scenario Designer is an embedded part of the TM and it is the component responsible for providing an intuitive designer where users can model real world Information and Communication Technology (ICT) infrastructures which will be replicated in a virtual or hybrid (containing physical elements) scenario.

The Scenario Designer allows to user to design the scenario from four different perspectives, called layers. There is the Training or Business layer that describes how this scenario is (expected) to be used by the users. For training scenarios, the expected attack or defense goals can be outlined. The application layer is used to define the relevant application landscape in the scenario. It also serves as a mechanism to determine which application should be installed on which machines of the scenario. The network layer is used to define the infrastructural layout of the scenarios. It contains among others the relevant virtual machines in the form of workstations servers or gateways and the interconnecting networks that make them a complete scenario.

Finally, there is the timeline layer. The timeline allows users to define a timeline of events that should happen during the execution of the scenario. Events on the timeline should be automatically executable in the form of automated attacks, scans or the application of countermeasures.

The information from all these layers can be linked to create a rich model describing all relevant aspects of the scenario.

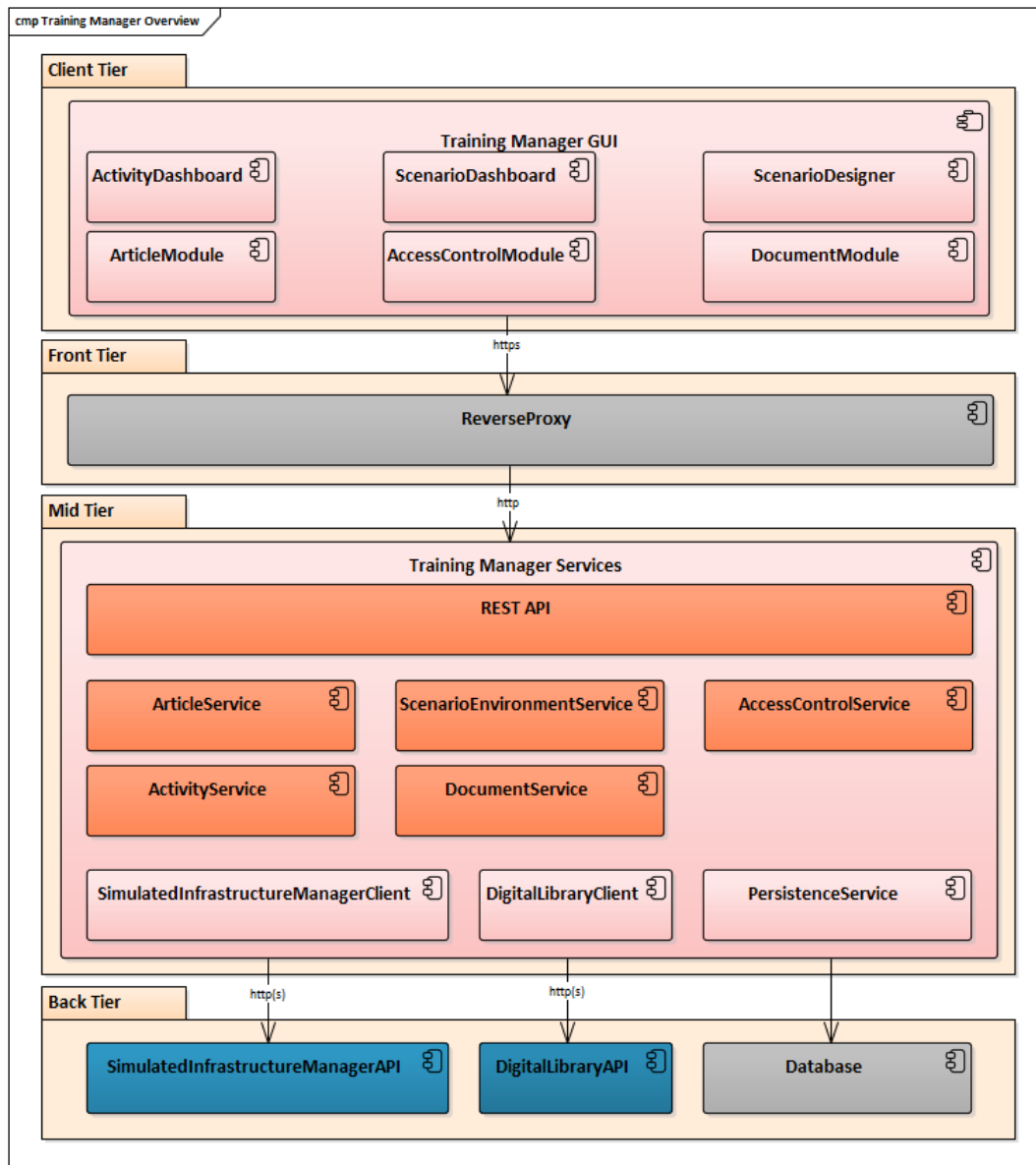


Figure 18. Training Manager components

The TM's internal design (see Figure 18) is comprised of three tiers. The back tier consists of the storage components and external application interfaces the TM needs to realize its capabilities. These consist of the API of the SIM as well as the API of the DL. The API of the SIM is to submit a designed scenario request to the SIM for the creation of an instantiable template as well as providing an interface for controlling and using the scenario at runtime.

The Mid-Tier consists of the business logic required for the TM and is realized by several separated services each providing a specific subset of the overall required functionality. The ArticleService is concerned with the management of the various textual content required in the TM. The DocumentService is responsible for the

management of Documents throughout the TM. The ActivityService is responsible for the management of the services and activities in the TM. The main functionality is provided by the ScenarioEnvironmentService which is responsible for everything related to the design and usage of training scenario.

The Front Tier represent the graphical user interface that is exposed to end users. It is decomposed into a set of modules each providing a specific subset of the overall frontend logic and presentation capabilities. The modules in the Front Tier correlate one-to-one to the individual services in the Mid-Tier and have the same responsibilities albeit now related to frontend logic and user interface display.

## 5.5 Performance Evaluator

As presented in the Description of Action, the PE is a component in charge of assessing how well the participants in the training are performing / performed. This component will be built from scratch during the project.

The PE evaluates the progress of the trainees during the different exercises carried out in the cyber range, basing on a list of inputs which are relevant for the performance evaluation that need to be monitored.

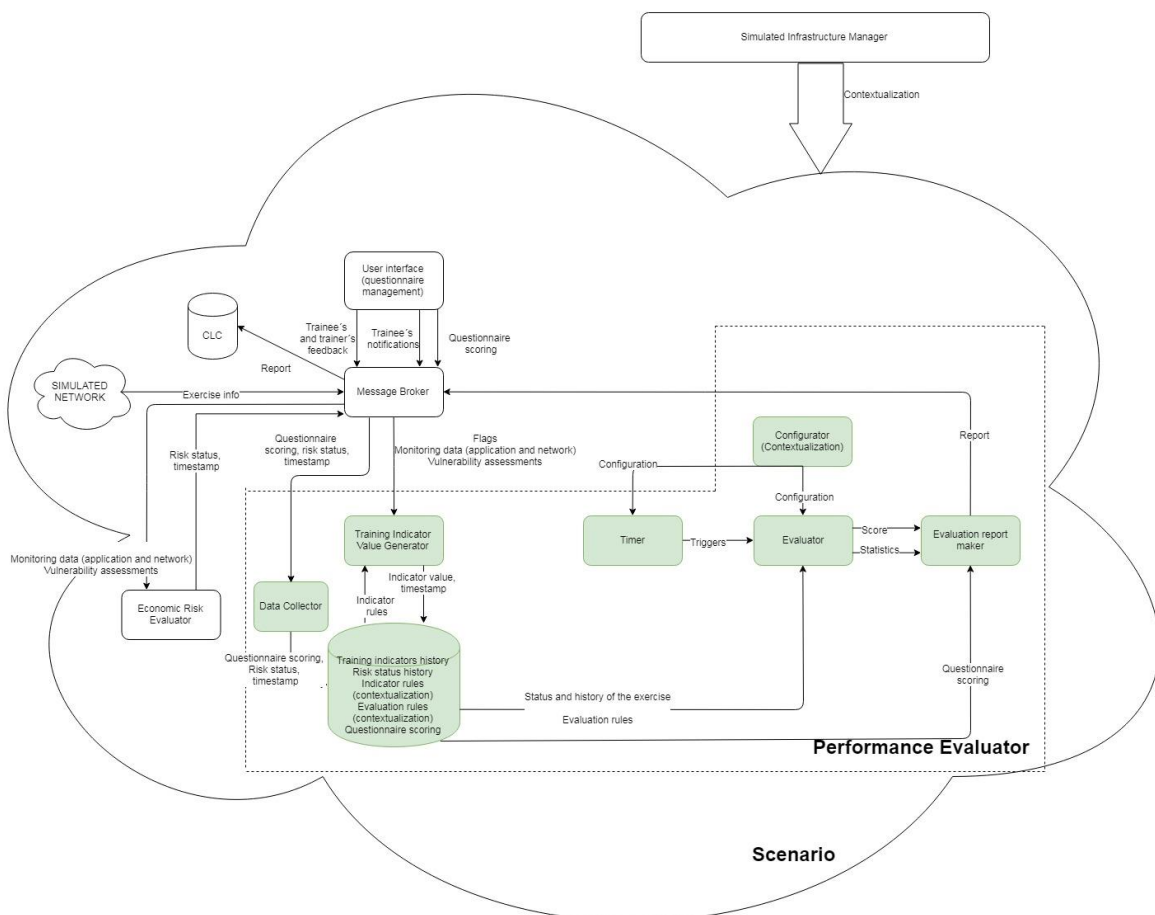


Figure 19. Internal composition of the Performance Evaluator and interfaces from/to the outside

Figure 19 shows the internal composition of the PE and the interfaces to other components. The PE has an internal *Evaluator* that needs as input both the current status and the history of the ongoing exercise. Such

status is composed by a series of training indicators<sup>38</sup> defined in advance and being part of the contextualization of the exercise, that takes values from a certain set of possible values; and by the cyber risk status calculated by the ERE. Not only the current value of the training indicators and the cyber risk status is stored, but also all the previous history. This is, when their value change, the new value is stored along with a timestamp representing the moment in which such change took place. In this way, not only the current values of the indicators and the cyber risk exposure can be taken into account, but also all their previous evolution which in many cases is relevant to evaluate performance<sup>39</sup>.

To produce the training indicator values, there is a block called *Training Indicator Value Generator* that obtains the corresponding value to an indicator (appending a timestamp) considering three different types of inputs coming from the MB: flags generated within the exercise (relevant events, mainly achievements of the trainee), monitoring data coming from the network and application sensors, conveniently transformed into events and especially alarms by the ADR, and vulnerability assessments. The rules to calculate such values are provided by contextualization and stored in the internal database of the PE. The *Training Indicator Value Generator* will query this database to obtain such rules. To calculate the cyber risk status, as it happened in the WISER project, the ERE needs the monitoring data and the vulnerability assessments, provided by the ADR and the VAT via MB. The internal composition of the ERE is addressed in section 5.7. Its outputs are stored in the internal database of the PE by means of a *Data Collector* block taking care of this mission.

The *Evaluator* is configured by means of the contextualization, this is, in the contextualization file there must be basic information defining the way the internal algorithm will operate. The evaluation rules are stored in the internal database and are retrieved when a new evaluation needs to be done. The evaluation algorithm will be invoked periodically. The block in charge of invoking it is the *Timer*, with a frequency that is configured thanks to the contextualization as well. The *Evaluator* uses the stored data about training indicators history and the cyber risk status history to produce two types of outputs: the score itself and several statistics of the exercise. These data are passed to a block named *Evaluation Report Maker* in charge of producing the performance evaluation report which is sent to the CLC via the MB. As presented earlier in the document, part of the scoring is calculated outside the PE, in the scenario itself, and is provided via the MB and stored in the internal database thanks to the *Data Collector*. This way, the *Evaluation Report Maker* can query the internal database to obtain this information.

Figure 20 shows the internal design of the PE.

<sup>38</sup> In CYBERWISER.eu we inherit the concept of “indicator” as input to an evaluation algorithm. This concept was developed in the WISER project for the evaluation of cyber risk. In this project we also use indicators for training performance evaluation, calling them “training indicators”

<sup>39</sup> The time a defender takes to execute a certain mitigation action makes necessary to store the timestamps of the indicators and cyber risk changes. For example, the introduction of input validation mechanisms in the event of SQL injection attacks

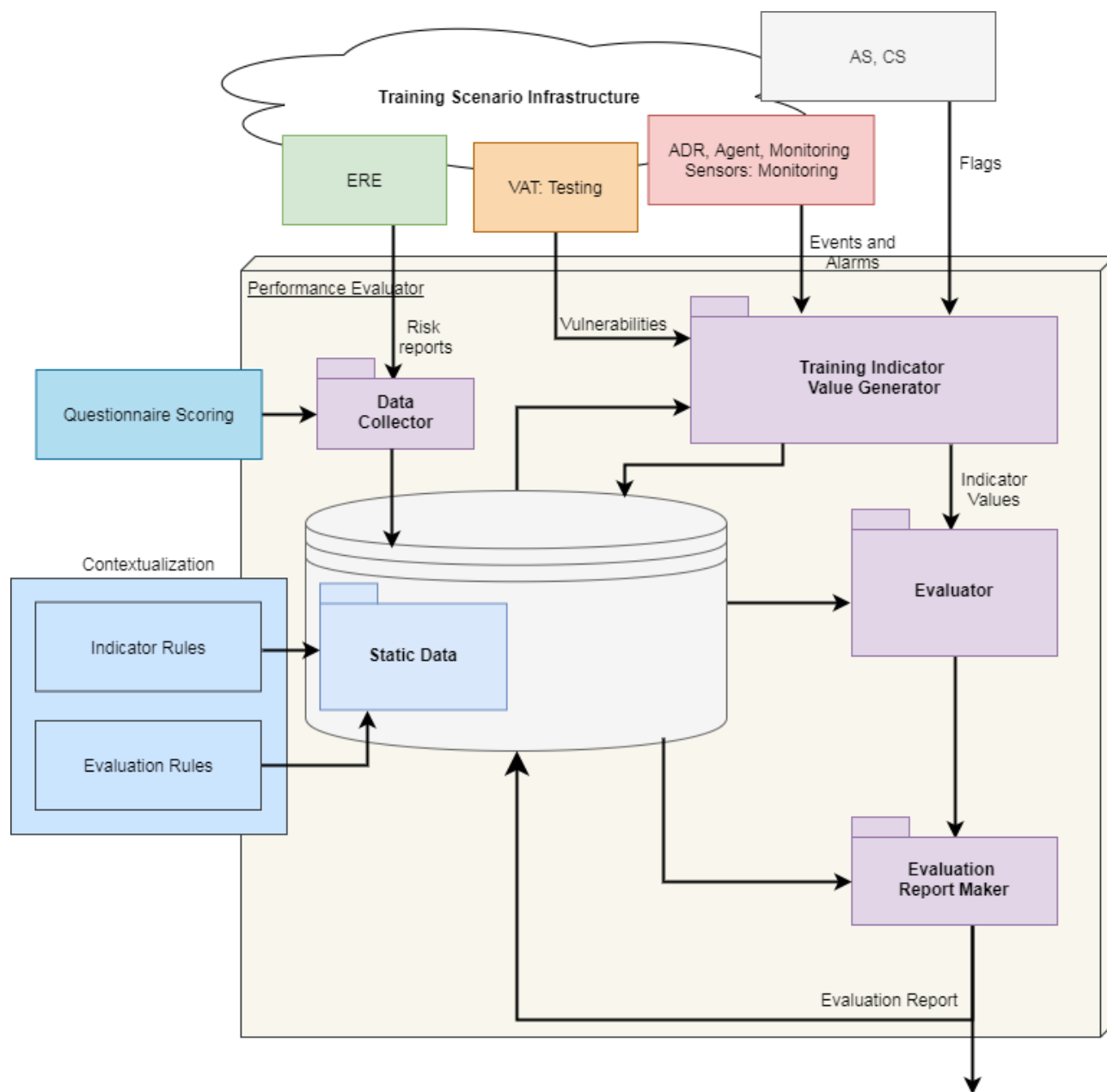


Figure 20. Internal functional design of the Performance Evaluator

The two main challenges for the implementation of the PE are the collection of the different events needed by the Exercise Evolution Evaluator and developing the internal algorithm of this component itself. Specific research will be done on how to produce these events. This is, how to detect automatically that the events are happening and produce appropriate logs that can be analyzed properly on the PE side to extract the events. Once the events are in place, what to do with this information and what criteria to use to figure out a score basing on the evolution of the exercise is not trivial. This will require specific research and the creation of innovative algorithms to bridge the gap and solve the challenge. As this component issues a report with the evaluation of the exercise and the corresponding mark, this report needs to be visually appealing with the key information clearly identified and well-structured, with direct messages highlighting the strengths and weaknesses of the trainee.



## 5.6 Digital Library

The DL is a storage component that stores three main categories of information relevant for the design and creation of virtual environments. The first is the meta-data describing different Simulated Assets such as the images containing operating systems and complete virtual machines or the scripts that define the installation of software applications on those machines to simulated users. The meta-data consists of a set of properties which together define what each Simulated Asset is dependent on and what kind of capabilities it provides. The second are the actual Training Scenarios, which may be stored in the DL as well, ready for re-use. Finally, the DL holds meta-data related to Physical Assets that may be part of certain scenarios.

In the overall architecture of the platform, the DL fulfills the role of central storage location for this information. The TM uses it to query for available Training Scenarios and Simulated and Physical Assets which are used in the creation of Training Scenarios. When these scenarios are sent to the SIM for deployment, it too uses the information in the DL to validate and deploy the scenario.

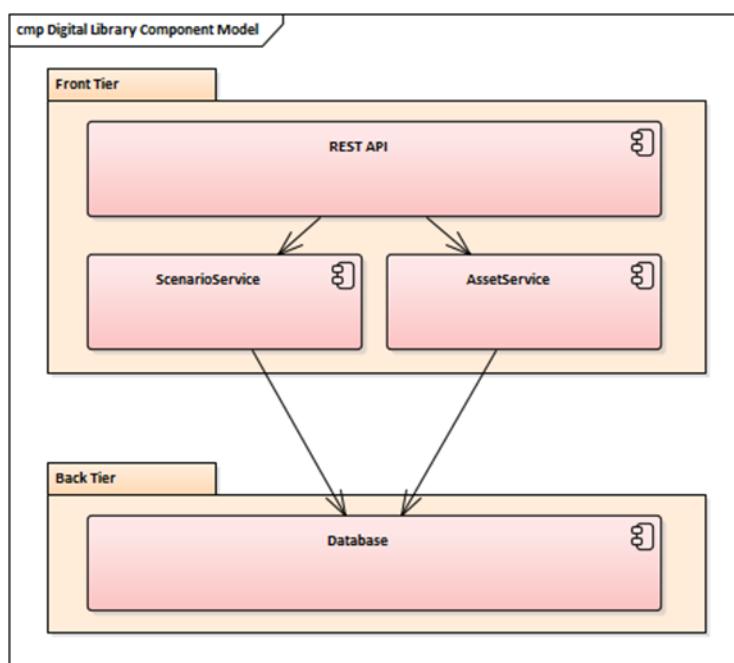


Figure 21. Digital Library Component Diagram

The above diagram (see Figure 21) shows the design of the DL itself. It is comprised of two layers, a database layer and a business logic layer. The business logic is exposed to the other parts of the platform by a RESTful API.

The DL will be expanded to allow for the storage of scenario definitions and attack and countermeasure scripts. It will also be improved with a property mechanism that allows the user to clearly see which objects in the DL can be used on top of other objects. For example, an application configuration of application A can be used on top of application A which itself can be used on top of a 64bit, Windows 7/8/10 virtual machine.

## 5.7 Economic Risk Evaluator

The ERE application is an evolution of the Risk Assessment Engine (RAE) that was born in the WISER Project<sup>40</sup>. This application can estimate in near-real time the economic exposure of the simulated infrastructure. The ERE calculates and assesses the monetary risk level of the scenario through economic risk model-based algorithms whose inputs are defined in the shape of indicator<sup>41</sup> values. These indicator values are generated from different sources of information:

- Configuration questionnaire, based on the training scenario environment that should be provided as a deployment configuration.
- Targets configuration based on the scenario definition, such as the IP addresses, port (if the target in question is an application) name and description, and how they would be impacted by an incident about confidentiality, integrity and availability from which the typical and worst economic loss will be estimated. This information should also be provided during the scenario configuration.
- Events and Alerts that would be collected from the MB (presented in section 3.2.1) and generated in the monitoring infrastructure (ADR, ADR Agent and Monitoring Sensors). This information will be provided in real-time during the execution of the training scenario.
- Vulnerabilities detected in the training scenario infrastructure. This information is sent to the RabbitMQ when a vulnerability scan is performed by the VAT.

From the point of view of the ERE application, the economic risk models (further presented in section 5.7.1) are a set of algorithms that provide the economic risk level in the training scenarios based on the infrastructure status, the likelihood of occurrence of certain incidents and their associated impact in terms of economic loss. Each training scenario should apply one or more risk models depending on the training goals, so the ERE application will execute the risk models applicable for each training scenario.

In the same way, not all the indicator values are applicable in every risk model. These indicator values are assigned from the sources of information detailed above and should match the pre-configured rule condition in their associated indicators which are taken out from the risk models. Please note the difference between indicators and indicator values; the indicators are pre-configured in the application according to all risk models and represent conditions that should match the incoming information (questionnaires, targets, events, alarms or vulnerabilities) while the indicators values are the result of applying the indicator conditions to the incoming information. In summary, the indicator is the configuration of what incoming information should be considered and the indicator value is the real input for the risk model to support the automated assessment of economic risks.

The ERE application works in near real-time since it takes some seconds to manage the incoming information and to execute the corresponding algorithms associated to the models. The concept real-time does not mean it is executing the algorithms recurrently, but when some of the inputs match an indicator condition that produces a change in the status of the indicator values, and that can lead to a change in the risk models' inputs. In such way, the computational efficiency allows a better performance and lower resources requirements.

Figure 22 shows the internal functional design of the ERE application. The ERM and their related algorithms, the training scenario environment with the responses of a questionnaire and the targets configuration should be provided initially to the ERE application as static data. This information generates the related indicator values, along with this, the events, alarms and vulnerabilities arriving to the ERE are also translated by the

<sup>40</sup> H2020 Project WISER. Grant Agreement n. 653321. [www.cyberwiser.eu](http://www.cyberwiser.eu)

<sup>41</sup> The concept of indicator is envisioned and presented in the WISER project, in particular in deliverable D3.1: "Cyber risk patterns"

*Indicator Value Generator* into indicator values, as long as they match the indicators configured according to the risk model's definition.

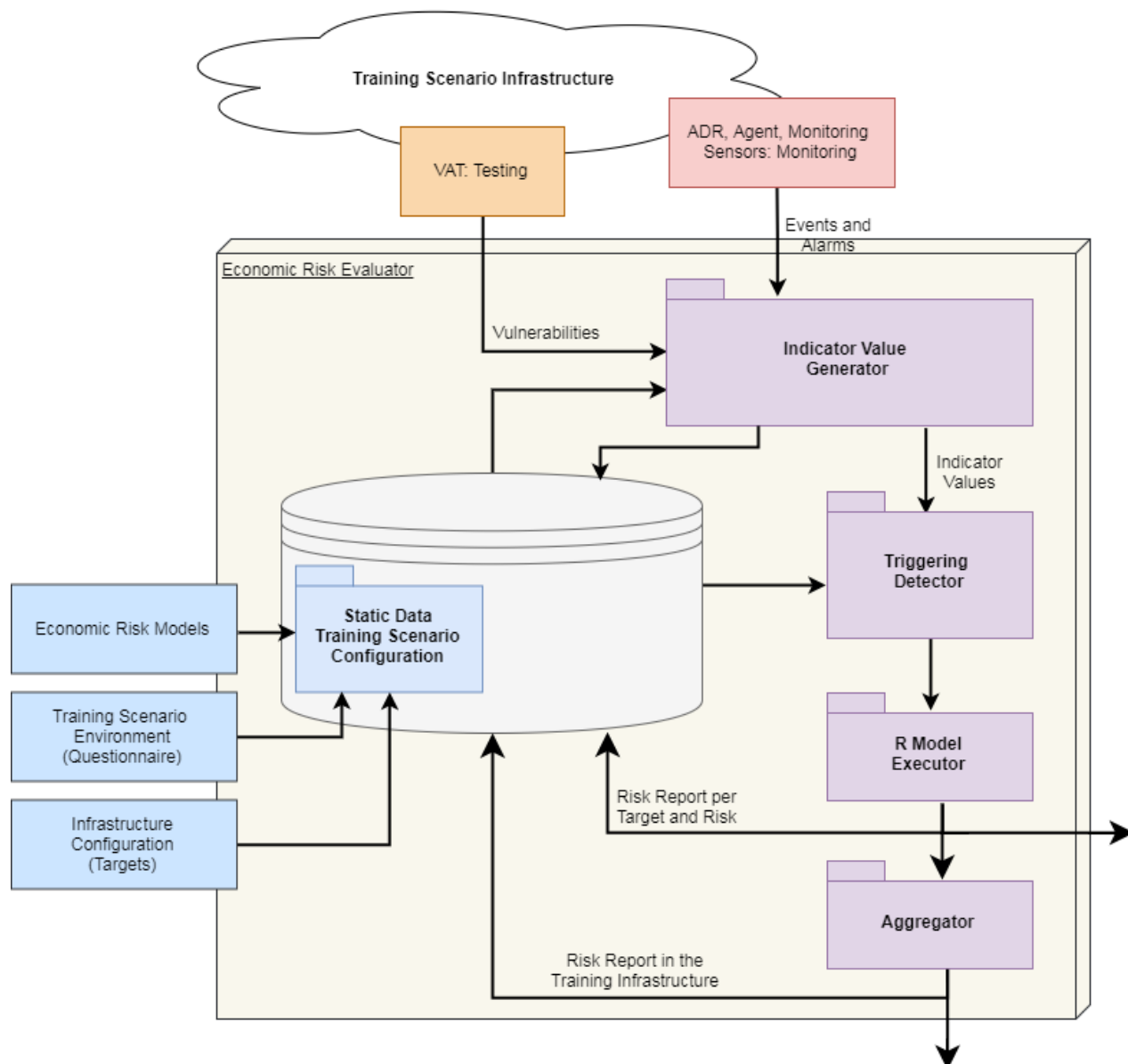


Figure 22. Economic risk evaluator internal functional design<sup>42</sup>

The *Triggering Detector* receives the indicator values and checks if their status has changed, and in such case, it will launch the algorithms associated to the applicable risk models in the corresponding training scenario.

The *R Model Executor* is in charge of executing the risk model algorithms which will provide quantitative risk level assessments, in terms of economic loss, for each target in the infrastructure and for each risk inside a specific risk model. The algorithms should be provided in the form of R scripts that represent the corresponding risk model. The inputs for each R script come from the indicator values that are involved in their corresponding

<sup>42</sup> This figure is an adaptation of that included in WISER deliverable D5.2 "WISER Real-time assessment infrastructure"

risk model definition. The R Model Executor runs the algorithms when the Triggering Detector notices a change in some indicator values that affects the risk models.

Finally, the *Aggregator* will be responsible for grouping the risk assessments per targets and risks, provided as output by the R scripts, into global assessments, i.e. assessments for the all infrastructure, assessments for each risk model and assessments for each target.

The user is informed about the money being exposed in terms of cyber risk. Also, the ERE can propose one or more mitigation measures triggered by the model algorithms according to the infrastructure status (reflected in the indicator values). The ERE will offer a feature consisting in simulating what would happen if a certain mitigation measure was applied. The risk will be re-calculated, and it will be possible to assess the effect such measure will have. This completes very well what is offered by the CS presented in section 5.13.

ERE application consists of three tiers, the presentation, the business logic and the back tier, which are represented in Figure 23. The back tier provides data storage and data exchange resources to send, receive and save the required information. The business tier will expose the report results and the component configuration to the rest of the platform through the DataService via a REST API. This layer also contains the AssessmentService which is responsible for all the processing logic to achieve the risk level assessment. The Presentation tier provides the access point for the end-users to configure and consult the corresponding information.

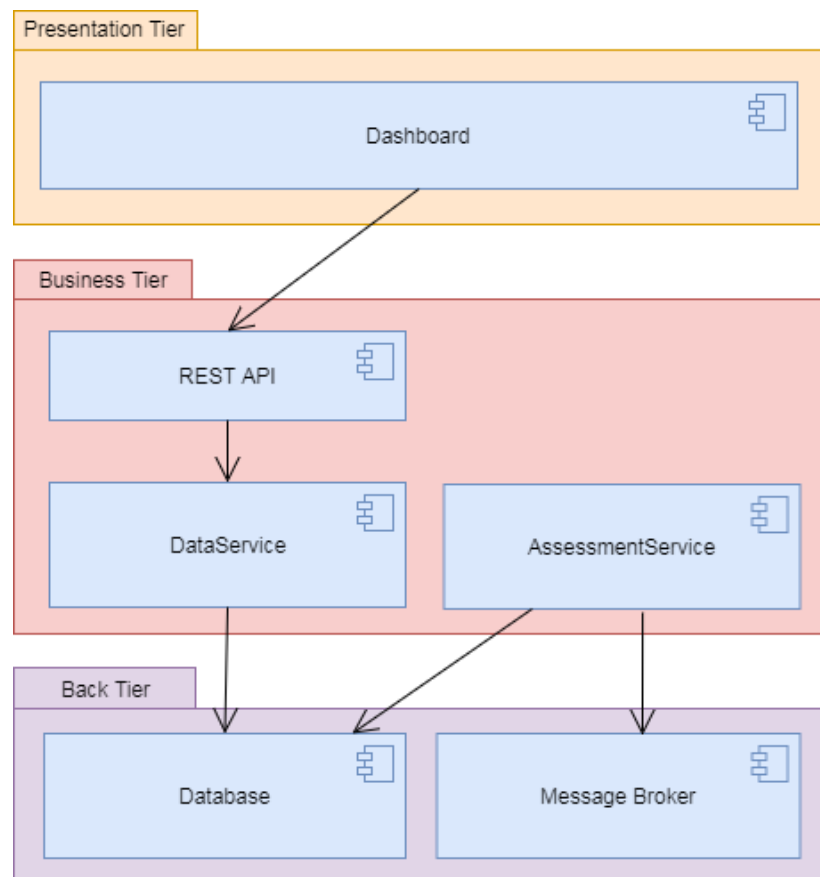


Figure 23. Economic Risk Evaluator layer design

As mentioned before, the ERE in CYBERWISER.eu is an evolution of the RAE application presented in WISER. The main improvements that will be emphasized for this component are listed below:

- In WISER project, the application was purely a backend component, and it was not able to work independently of the rest of the platform, so it needed to be integrated in the WISER platform. In this

sense, in order to enhance the standalone capacity of this component, the ERE will be integrated with a Django application which will provide data storage to save incoming and output data, as well as intermediate data; GUI that allows users to interact with the application.

- Convert Python 2.7 code to Python 3.6.
- The new version of this application should allow an initial data load during boot-up to configure the ERE according to the training scenario configuration.
- The updated application will improve the performance as a result of an architecture optimization.
- We will explore the possibilities of integrating the ERE with other third-party applications, such as commercial Security Information and Event Management (SIEM) solutions.
- Usage of more sources, like new vulnerability scanners

### 5.7.1 Models

The overall goal of risk modelling in CYBERWISER.eu is to offer support for development of: a) graphical cyber risk models to establish and communicate understanding among (human) stakeholders with a wide range of different background; and b) corresponding machine-readable cyber risk assessment algorithms that can be executed in real-time by the ERE in order to provide a list of risks along with a risk level assessment for each risk, in terms of monetary loss.

Graphical risk models are created using the CORAS risk modelling language<sup>43</sup> as well as the CORAS risk modelling tool adapted for WISER<sup>44</sup>. The risk assessment algorithms are developed schematically with respect to the graphical risk models mainly following guidelines also provided by WISER<sup>38</sup>. In the following we provide a high-level explanation of the cyber risk modelling approach in CYBERWISER.eu and how produced models and corresponding algorithms are related to the ERE.

As illustrated in Figure 24, the risk modelling process in CYBERWISER.eu consists of two overall steps. These two overall steps consist of further detailed sub-steps, but the detailed steps are outside the scope of this deliverable and will therefore be described within the training materials generated in the context of WP4. In Step 1, we establish and document a good understanding of the relevant risks, scenarios in which the risks may materialize, the relationship between the scenarios, as well as indicators used to collect information about the simulated infrastructure that can be used to assess the risks and the involved, vulnerabilities, threat scenarios and unwanted incidents. The first step produces graphical CORAS risk models with indicators which is used as input to Step 2. In Step 2, we schematically translate the CORAS risk model with indicators to a machine-readable risk assessment algorithm expressed in the **R** programming language<sup>45</sup>. Thus, the output of Step 2 is R scripts capturing the risk model produced in Step 1, where the indicators identified in Step 1 act as expected input parameters to the algorithm.

<sup>43</sup> M. S. Lund, B. Solhaug and K. Stølen: Model-Driven Risk Analysis. The CORAS Approach. Springer, 2011.

<sup>44</sup> A. Refsdal, G. Erdogan, G. Aprile, S. Poidomani, R. Colgiago, A. Alvarez, P. Lombardi, R. Mannella. WISER public deliverable D3.2 – Cyber Risk Modelling Language and Guidelines, preliminary version. Available online: [www.cyberwiser.eu](http://www.cyberwiser.eu) (accessed 08.01.2019).

<sup>45</sup> The R Project for Statistical Computing. Online: <https://www.r-project.org/> (accessed 08.01.2019)

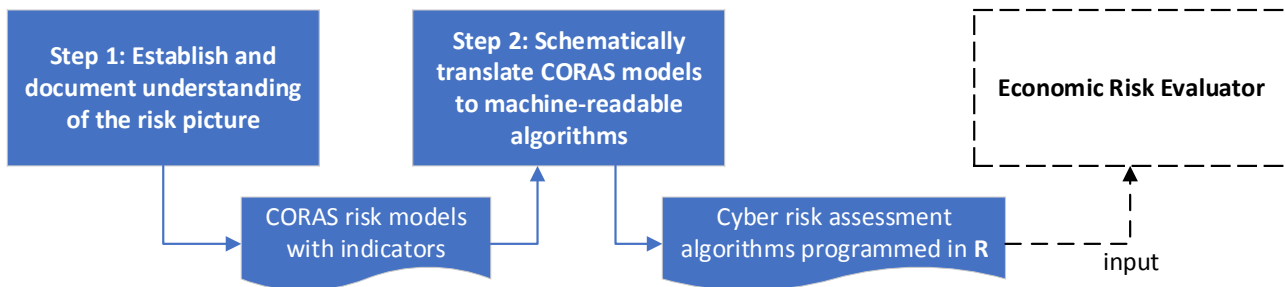


Figure 24. Outline of overall method for cyber risk modelling and its relationship to the Economic Risk Evaluator

The relationship between the R scripts and the ERE is illustrated in Figure 24 by the dashed arrow and rectangle. The produced R scripts are fed into the Economic Risk Evaluator which is responsible of executing the R scripts (risk assessment algorithms). This includes collecting information from the simulated infrastructure with respect to the indicators and feeding this information to the corresponding parameters in the R scripts.

The approach to create risk models in CYBERWISER.eu is based on the approach provided by WISER. This includes the existing risk models, as well as the method for cyber risk modelling. The main improvements that will be emphasized with respect to risk modelling in CYBERWISER.eu over the WISER approach are the following:

- The risk modelling and calculation methods will be refined and presented pedagogically considering a wide range of target group, and not only experienced cyber risk professionals as in WISER.
- Adapt and possibly extend the WISER risk models such that they can be applied in the cyber range developed in CYBERWISER.eu.
- Place the risk modelling and calculation method in an overall risk analysis process. This includes the consideration of context establishment, risk assessment, and risk treatment.
- Develop new risk models following the abovementioned new overall risk analysis process.

## 5.8 Event-Based Service Orchestrator (EBSO)

In its core, the EBSO is a component capable of orchestrating services running in Docker containers. We refer to these services as *managed services*, since EBSO manages all their configuration, instantiation and also teardown. By designing attacks and vulnerability scans around this technology and logic, we can use EBSO as a shared backend supporting the AS and VAT components, essentially making launching of attacks and vulnerability scans managed services of EBSO. The runtime environment (supported by Docker) provided by EBSO to the managed services is central to the component, however it has two other relevant parts: 1.) the scheduling logic dealing with when (at what time) and how (how many times, with what interval in between) to execute managed services, and 2.) custom result processing and reporting, which is CYBERWISER.eu-specific and may vary between the managed services.

Figure 25 depicts the internals of EBSO in a three-tier view with the front tier acting as a gateway to domain logic in the mid-tier, and the back tier representing the supporting services/infrastructure required by the mid/front tiers.

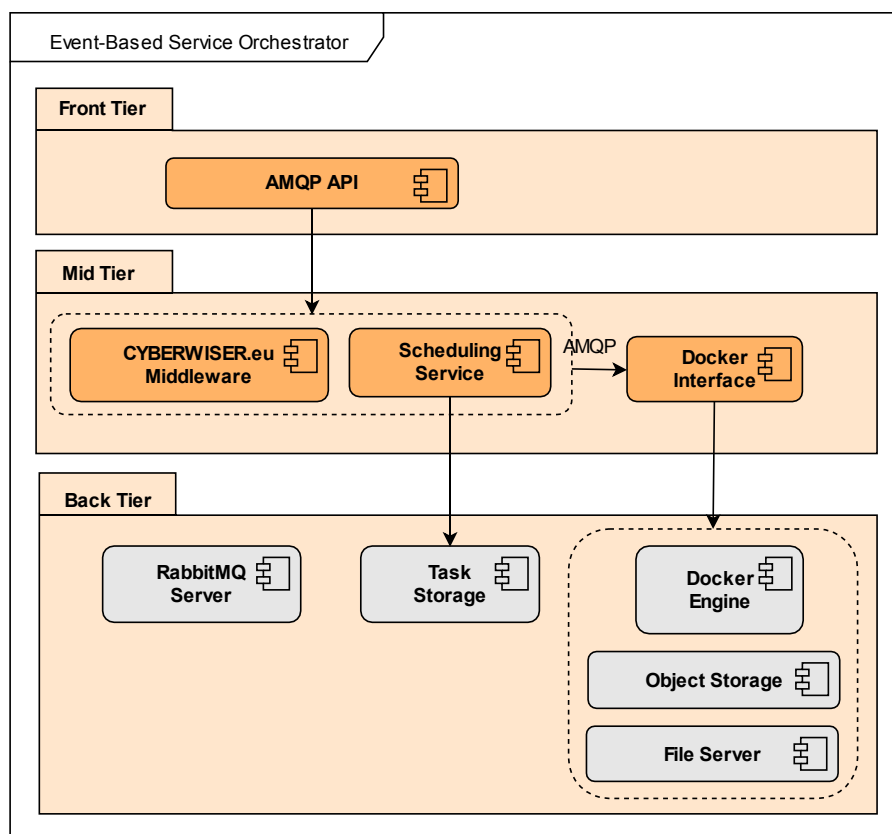


Figure 25. Event-Based Service Orchestrator component diagram

The front-facing part of the EBSO is a REST-like API served via AMQP, indirectly providing control over managed services, i.e. launchers of attacks and scans, and providing access to the data related to their executions. As further described in the next paragraph, orchestration of attacks and scans revolves around the concept of tasks, thus the API provides methods for submission, modification, cancellation and retrieval of data about tasks. The API is exposed to the front-end parts of VAT and AS (please refer to Sections 5.9 and 5.12, respectively).

The mid-tier of EBSO comprises CYBERWISER.eu Middleware, Scheduling Service and the Docker Interface. The **Scheduling Service** handles execution of tasks according to the associated schedules. It is generic in nature and works with an abstract notion of tasks. For the purpose of supporting the envisioned workflows for the AS and VAT components, concrete tasks revolve around attacks and scans at a high-level. At the low level, a task corresponds to communication with the Docker Interface (to whom we delegate the responsibility of runtime preparation for and execution of attacks/vulnerability scans), which is done via AMQP.

Upon receiving a request via AMQP from the Scheduling Service, the **Docker Interface** prepares the dynamic runtime environment for the managed service (i.e. for the Docker container that will launch the attack/scan against the target) on the host. This preparation may involve configuration of environment variables, mounting of input data, etc. To prepare said runtime environment, the Docker Interface communicates with the Docker Engine, as well as with other back tier components providing data to containers (managed services), for instance object storage server or network file server. Once the runtime environment is prepared, the Docker Interface runs the managed service in a Docker container, waits until the container exits, and persists the data generated by the container and its standard output to the object storage. This information is compiled into a response that also includes the return code of the containerized process, which is propagated to the Scheduling Service via AMQP.



Once the Scheduling service collects a response from the Docker interface, it passes it to the **CYBERWISER.eu Middleware** part. The latter processes the return code of the containerized service (included in the response) and emits an event that may or may not be significant for the exercise. Examples of such events are failure of an attack or presence of a particular vulnerability on the target. The same middleware layer that extracts information meaningful for the CYBERWISER.eu exercise from the responses coming from the Docker Interface also transforms requests from the front-tier API to requests comprehensible by the Docker Interface.

## 5.9 Vulnerability Assessment Tools

VAT provides two general vulnerability scanning options:

1. **Scanning for generic vulnerabilities.** Using a generic suite of vulnerability scanner modules, the VAT can be used to scan target infrastructure for common vulnerabilities. VAT will offer several scanning modules (e.g. OWASP ZAP<sup>46</sup> and w3af<sup>47</sup> scanners for web vulnerability scanning). The integration of other modules (e.g. Nmap<sup>48</sup>, OpenVAS<sup>49</sup>, etc.) is possible and will be further studied during the project lifetime. According to the configuration of a specific scan, multiple vulnerability scanner modules can be used, affecting the types of vulnerabilities that can be detected.

In case of a generic scan, the output of VAT is a vulnerability report containing a list of detected vulnerabilities. The report is provided to the ERE to be considered in the risk assessment.

**Generic scanning can be used automatically by the platform (to aid subsequent economic risk assessment), manually by the white team, and, if allowed by scenario-specific rules, also by the red and blue teams.**

2. **Detection of specific vulnerabilities with custom vulnerability detection scripts.** To detect the presence of specific vulnerabilities deliberately introduced in the exercise infrastructure as a part of a training scenario, custom vulnerability detection scripts that check for the presence of such vulnerabilities can be implemented. They allow us to automatically determine whether the trainees successfully mitigated the vulnerabilities. This is useful in training scenarios where the goal is to defend the exercise infrastructure, as it helps the platform to assess trainees' progress during the exercise.

Custom detection scripts will be implemented when necessary, according to the definitions of scenarios and the types of vulnerabilities represented in the simulated infrastructure. The scripts are executed by the attack tools offered by the AS component (see Section 5.12) and can be written in one of the supported languages.

The output of VAT in case of scanning a specific vulnerability with a custom detection script is an interpretation of a single boolean value indicating the presence or absence of the vulnerability. This information is provided to the PE. By observing the presence of vulnerabilities using periodic detections, the platform can determine whether and when the blue team was successful in mitigating them, which affects their performance evaluation.

**Vulnerability detection scripts can be used automatically by the platform (to aid in trainee performance evaluation) and manually by the white team to help keep track of trainees' progress during the exercise.**

<sup>46</sup> [https://www.owasp.org/index.php/OWASP\\_Zed\\_Attack\\_Proxy\\_Project](https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project)

<sup>47</sup> <http://w3af.org/>

<sup>48</sup> <https://nmap.org/>

<sup>49</sup> <http://www.openvas.org/>

VAT supports scheduling of both types of scans. They can either run once (at a particular time) or on a recurring basis with customizable time intervals. Schedules and other configuration can be defined in advance (as a part of the scenario definition) or during the training exercise.

The internal component diagram of VAT is presented on Figure 26. Its subcomponents and interactions are detailed below.

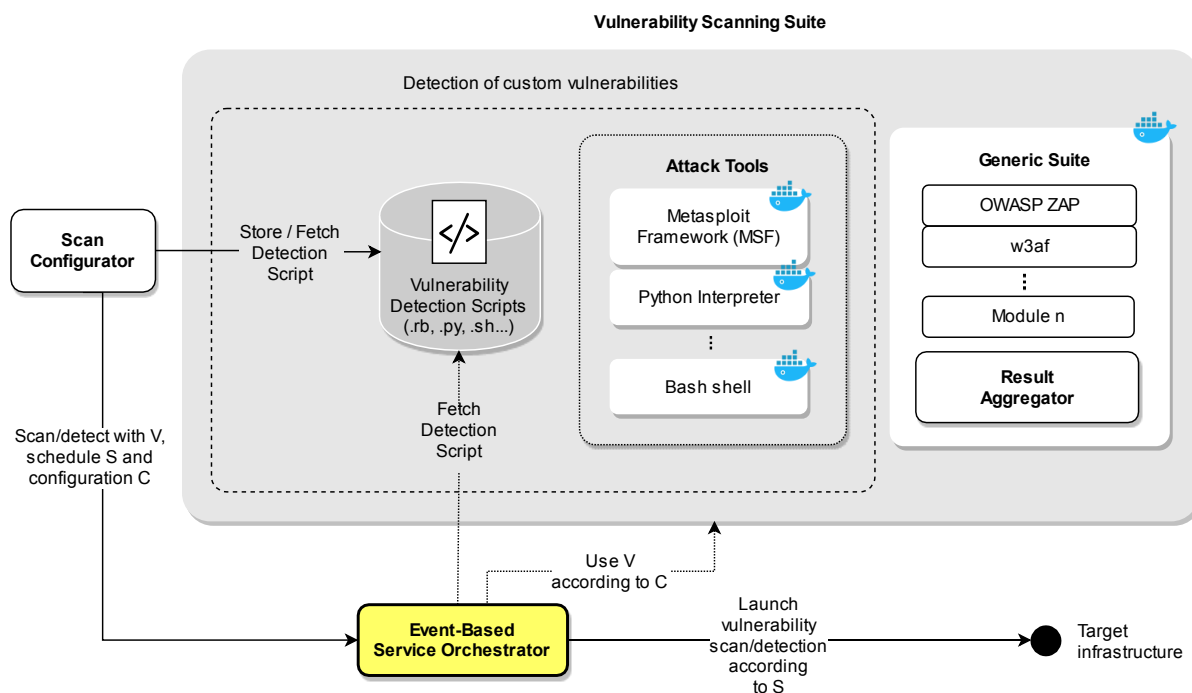


Figure 26. Logical view of vulnerability assessment tools

**Generic Suite** of vulnerability scanners incorporates various generic scanning modules. In the context of the Generic Suite, a single scan may involve invocation of one or more of these scanning modules. This introduces the need for an auxiliary component, the **Result Aggregator**, that combines the results of individual scanning modules into a vulnerability report in a format comprehensible to other parts of the CYBERWISER.eu Platform. The entire Generic Suite is packaged in a single Docker image.

In addition to the Generic Suite, VAT also supports custom **Vulnerability Detection Scripts** that perform very specific automated checks. Vulnerability Detection Scripts do not fall in the same category as scanners in the Generic Suite, as they are much more simplistic and thus far less comprehensive. However, they provide the ability to check for vulnerabilities hand-tailored for a given training scenario that escape scanners in the Generic Suite, therefore supporting an additional degree of customization. Vulnerability Detection Scripts are executed by one of the Attack Tools, i.e. Docker Images also used by AS, depending on the script's language. Technical execution of Vulnerability Detection scripts is exactly same as for the Attack Scripts (refer to Section 5.12 for details). In contrast to Attack Scripts however, Vulnerability Detection Scripts execute passive attacks, meaning that their goal is to report only about the presence or absence of a vulnerability. In addition, they generally don't change the state of the target machine.

The Generic Suite of vulnerability scanners, Vulnerability Detection Scripts and the associated Attack Tools support vulnerability scanning capabilities of VAT. Together, they form the **Vulnerability Scanning Suite** – a collection of tools for detecting various types of vulnerabilities in target systems.

**Scan Configurator** is a component with a Web interface to be utilized by trainers and/or trainees during an ongoing cyber-range exercise. It allows management and configuration of vulnerability scanning and detection offered by VAT through its Vulnerability Scanning Suite. Specifically:

- target of the scan (by means of an URL or an IP address),
- which scanning module(s) (for Generic Suite) or which Vulnerability Detection Script to run,
- suite/module/script-specific configuration (if any), and
- scheduling of scans.

EBSO is a backend component that schedules and orchestrates vulnerability scanning capabilities of VAT according to a given timeline and configuration. The same component is also used in the scope of the AS (see section 5.12) and it is additionally described in Section 5.8. When vulnerability scans or detections are requested by the CYBERWISER.eu Platform, meaning that they are known in advance as they are a part of a training scenario, the configuration is file-based. When the training scenario is up and running, configuration for scanning tasks can also be provided on-the fly by the trainers and trainees through the Scan Configurator that communicates it to EBSO via AMQP. Note that in the context of a single scanning task, EBSO executes either a particular Vulnerability Detection Script or the Generic Suite of scanners. Figure 27 depicts a tiered view of the VAT component. The front tier comprises user-facing component for configuration and management of vulnerability scans. The mid-tier encapsulates core business logic of VAT, namely the Vulnerability Scanning Suite. The back tier is represented by the EBSO component that schedules and orchestrates the business logic (vulnerability scanning and detection). Note that the front tier communicates directly with the back tier to configure and execute the logic in the mid-tier. Due to the specifics of the presented design, the mid-tier is represented as unconnected in the figure below:

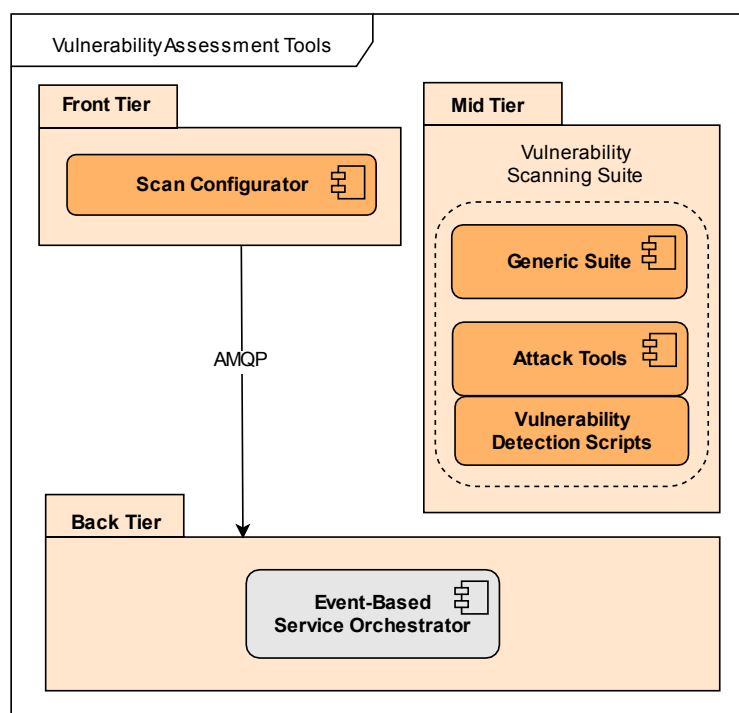


Figure 27. Vulnerability Assessment Tools component diagram

Regarding the innovation roadmap envisioned for VAT, the Generic Suite, as described above, has been developed and already used in the WISER project, and is brought into CYBERWISER.eu as background. It was connected to an HTTP API to manage the schedules and scan settings and reported the results via Syslog. Innovation for VAT addressed in the CYBERWISER.eu comprises changing the interfaces of the Generic Suite of scanners to fit into the described architecture, and integrating it with EBSO to manage scheduling more efficiently and generically (as EBSO is a shared backend used by both VAT and AS). Apart from the integration-related changes, we will study inclusion of new types of scanning modules into the Generic Suite. Further, new

functionalities will be added in the form of Vulnerability Detection Scripts. Thus, innovation in VAT is addressed mostly at the level of Vulnerability Scanning Suite extensions.

### 5.10 Monitoring Sensors

A sensor is a software entity capable of processing and analyzing information, eventually producing a useful output. Depending on where the information is collected, we distinguish between:

- Sensors working at the network layer level. They are installed at certain points of the network and their visibility scope is determined by factors like the topology or the configuration of the network. Depending on the network, given a specific type of network layer sensor only the installation of a single sensor would be required or, on the contrary, such installation should be replicated in different points of the network. These sensors collect information about the network activity, and work at the IP level, with data in transit. They can be added to the network in two different ways:
  - Offered as a package that is installed on a certain machine chosen by the network owner/administrator. This machine can be an existing one or a new one incorporated to the network with the purpose of hosting the sensor.
  - Offered within a Virtual Machine (VM) for which an IP address is assigned and added to the network.
- Sensors working at the host level: These sensors are installed on the machine where the application/s to be monitored is/are. These sensors collect information from the applications installed on such machine. They work with data prior to transmission or after reception. Application layer sensors usually work following a client-server model. For each machine with applications to be monitored, a client is installed, and on a separate machine the server side is installed and receives all the information coming from the clients. Depending on the network topology and configuration, a single server or more than one will be necessary.

Sensors do not perform highly complex processes over the information they collect, they are based on rather simple calculations that permit to obtain a first level of aggregation.

Sensors send their logs to an entity called *Cyber Agent*. This entity can be compatible to a wide range of sensors. The Cyber Agents have modules called *plugins* making possible to “understand” the data coming from the different sensor types. Plugins play the role of interpreting the logs related to certain types of events. There are as many plugins as number of sensor types to use. Plugins produce as output events that are sent to the ADR. Logs are collected by means of a *Rsyslog*<sup>50</sup> server. How the sensors collect information, send logs and these logs are collected by the Cyber Agent and interpreted to produce events to be in turn sent to the ADR is represented in Figure 28.

<sup>50</sup> Rsyslog website: <https://www.rsyslog.com/> (last accessed on 12/02/2019)

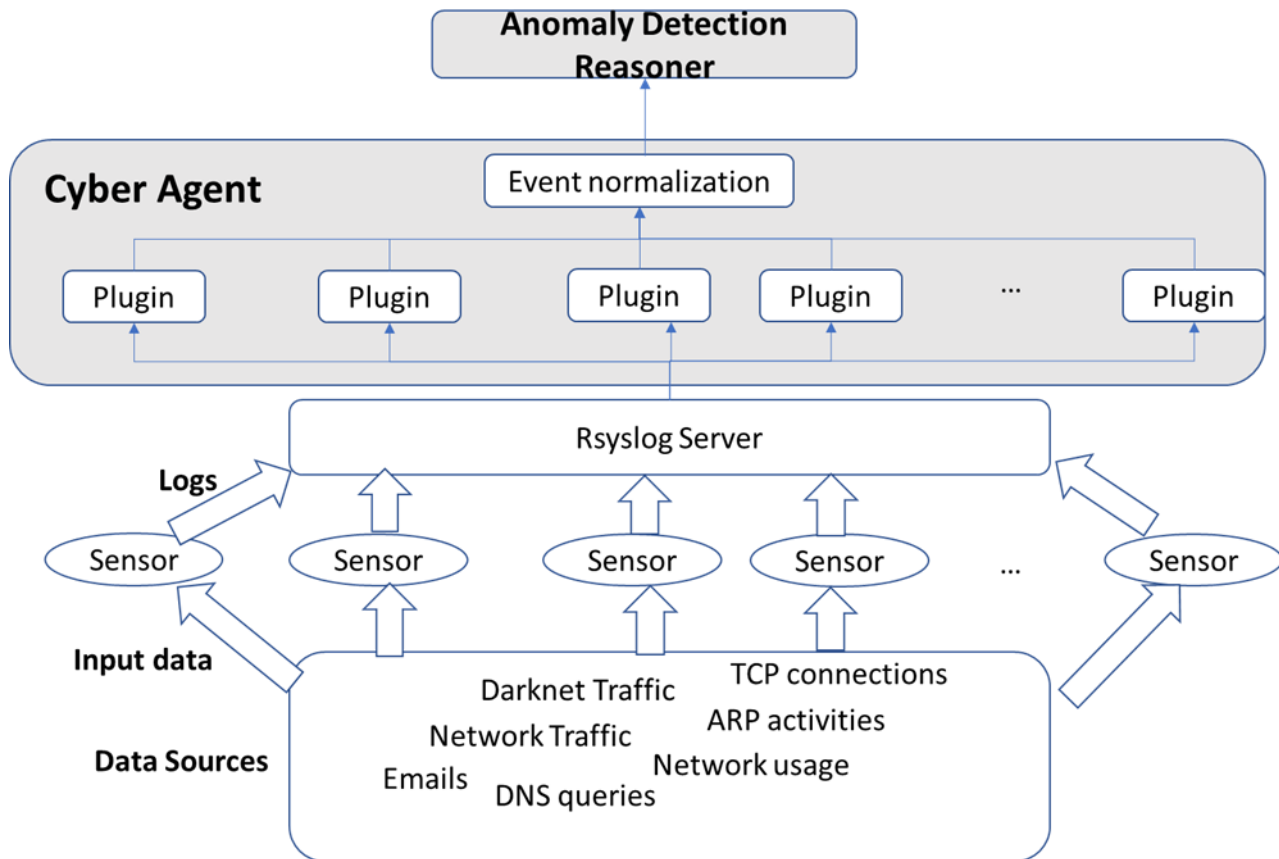


Figure 28. Monitoring sensors and Cyber Agent

There is a wide plethora of different monitoring sensors existing in the state-of-the-art. As the project evolves, the CYBERWISER.eu partners will decide which available open source sensors could be used in the exercise. Below some examples are listed:

- Suricata<sup>51</sup>: It is a Network Intrusion Detection and Prevention System which is open source. It can provide real-time intrusion prevention/detection and it can real-time monitor network traffic by structuring extensive rules using the Lua<sup>52</sup> signature language<sup>53</sup>. Suricata detects complex threats, supports multi-thread and multi-processor systems and inspects huge amounts of traffic data (gigabits).
- OSSEC<sup>54</sup>: It is a Host Intrusion Detection System. It monitors all aspects of UNIX system activity at the application level. OSSEC performs real-time application log and file integrity checks and, when an attack happens, it produces alert logs<sup>55</sup>.

<sup>51</sup> <https://suricata-ids.org/> Last accessed 15/01/2019

<sup>52</sup> <https://www.lua.org/about.html>. Last accessed 15/01/2019

<sup>53</sup> CIPSEC Project Deliverable D2.3: CIPSEC Products Integration in the Unified Architecture. Available on <https://www.cipsec.eu/content/d23-cipsec-products-integration-unified-architecture>

<sup>54</sup> <http://www.ossec.net>. Last accessed 15/01/2019

<sup>55</sup> WISER Project Deliverable D4.1: Design of the WISER Monitoring Infrastructure

- Cowrie honeypot: Honeypots are specially deployed servers, used to attract attackers and detect their presence while averting the attacks from other machines of the network. Cowrie exposes a Secure Shell (SSH) over the network. It logs connection attempts and commands executed by the attackers<sup>56</sup>.
- Nagios: it is a network / system status monitoring daemon. It is capable of measuring parameters such as Central Processing Unit (CPU) load, disk usage, number of current processes. Current SSH sessions, memory usage or running processes to name but a few. This information, conveniently stored, allows a myriad of applications such as forensic analysis of possible security or outage events in machines in the network.<sup>23</sup>

For sensors in general, an improvement to be accomplished during the project is the detection of configuration changes in the different machines of the infrastructure. This can be applied, for example to the detection of the application of certain mitigation measures. For the evaluation of the student, it is important to detect not only that he has carried out a specific action, but also when he did it. The Consortium will develop sensors to detect specific actions carried out by the red and the blue team which are relevant for the evaluation of the exercise.

An important cross innovation to be addressed in the short term is easing sensors' deployment by using containers (Docker or similar). This will be fundamental as the scenarios are deployed in an automatic way, and so far, in previous projects, sensors have been installed manually. This is not possible in CYBERWISER.eu, where the automation of this step is necessary.

### 5.11 Anomaly Detection Reasoner

The ADR is based on the SIEM solution called XL-SIEM brought by ATOS to the project. This is a SIEM solution with added high-performance correlation engine to deal with large volumes of security information. It is built on top of the Open Source SIEM called OSSIM<sup>57</sup>.

The objective of this asset is the real-time detection of security threats, distinguishing the normal network activity from the suspicious one. As previously said, it plays the role of anomaly detection reasoner. It normalizes, filters and correlates information coming from heterogeneous sources. It obtains valuable insights about the cyber climate of the monitored infrastructure. Starting with huge amounts of data, this asset produces meaningful events and then raises alarms following complex event correlation rules.

The XL-SIEM offers sophisticated real-time security analysis technology with highly interoperable, scalable and elastic, security events processing through a cluster of nodes. It is cross-layer, allowing for the convergence of physical and cyber security. Figure 29 shows the internal component diagram of the ADR.

<sup>56</sup> WISER Project Deliverable D4.2: WISER Monitoring Infrastructure

<sup>57</sup> <https://www.alienvault.com/products/ossim> (latest accessed on 19/06/2019)

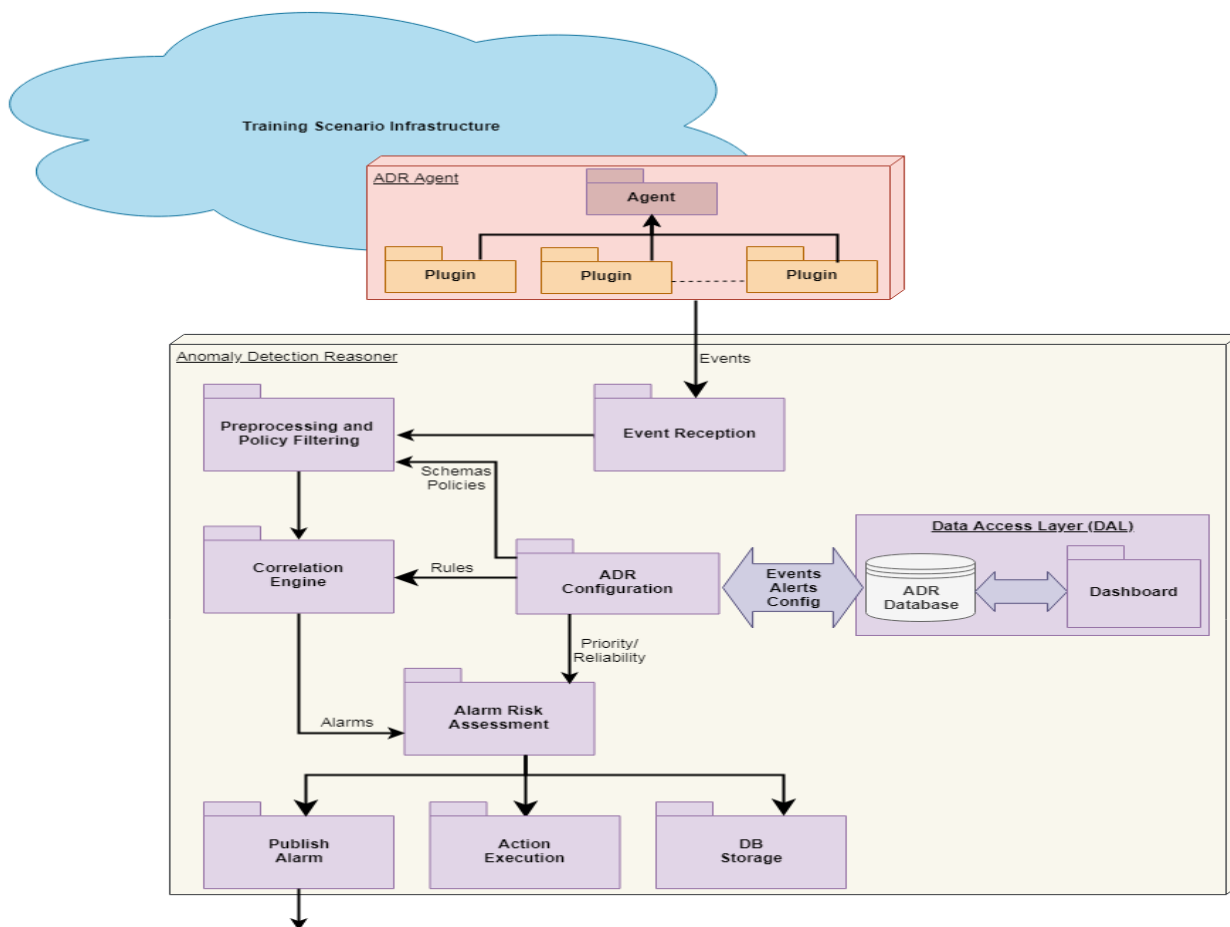


Figure 29. Anomaly Detection Reasoner diagram<sup>58</sup>

There is an internal module dedicated to the reception of the events coming from the monitoring sensors deployed on the client infrastructure. These events are pre-processed and filtered according to certain policies which are configured beforehand by the user leveraging the *ADR Configuration* module. This been done, the data is fed into the *Correlation Engine* which also needs to be configured by the user, who establishes the rules for the identification of the alarms. There is a block devoted to the risk assessment of the issued alarms, also needing to be configured by the user. Both events and alarms are published in a dashboard where the user can check the status and launch the execution of certain actions to address the problems informed by the alarm.

The architecture of the ADR is built to provide real-time distribution across different machines not only for the correlation process, but also for the support of different filtering policies, different rules and different data schemas associated with each process. This allows for greater flexibility in processing and improving processing capabilities and improving processing capabilities and optimization of the use of available resources.<sup>59</sup>

<sup>58</sup> This diagram is the adaptation of that included in CIPSEC Deliverable D2.2: “CIPSEC Unified Architecture, first internal release”

<sup>59</sup> CIPSEC Deliverable D2.2: “CIPSEC Unified Architecture First Internal Release”  
<https://www.cipsec.eu/content/d22-cipsec-unified-architecture-first-internal-release> (last accessed on 27/06/2019)



It is relevant to say that the events are basic pieces of information that should be considered at warning level, this is, something that requires heads up and follow up but no immediate reaction. Alarms are obtained by means of the aggregation and correlation of events. The more events participating in the correlation process, the more specific the alarm launched is, the higher the risk is and the more seriously this information must be taken. In fact, as a general rule an alarm should involve an immediate reaction, since it reports issues that must be treated on urgent basis.

ADR application works in a three-tier-architecture (see Figure 30), the presentation tier, the business tier and the back tier. The back tier provides data storage and data exchange resources, while the business tier will provide a REST API to access the application data through the DataService and its configuration using the Policies&AlarmsService. The Policies&AlarmsService component also controls and executes all the internal flows presented in Figure 29. The presentation tier provides the GUI for the end-users.

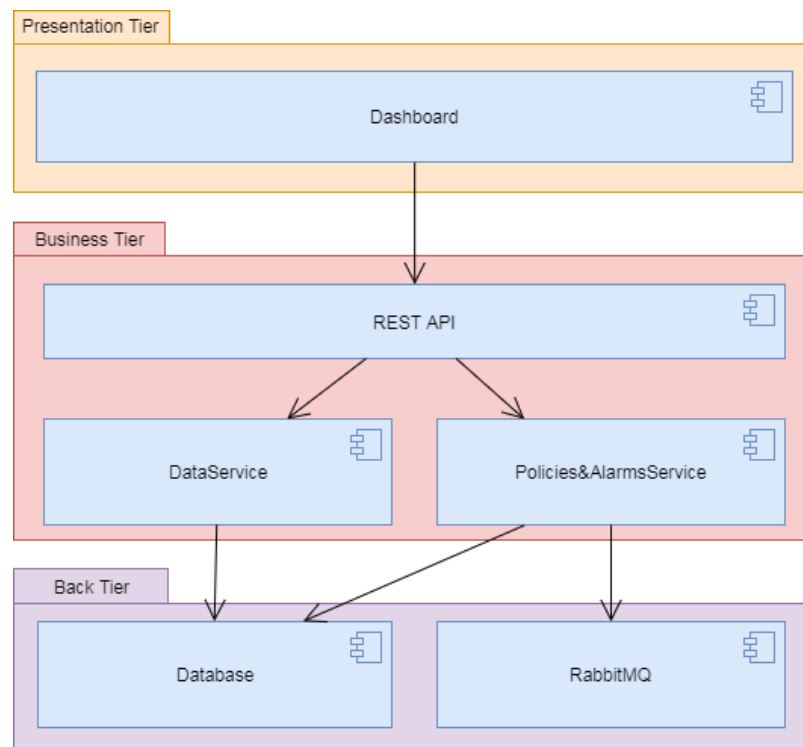


Figure 30. Anomaly Detection Reasoner, layer design

The ADR will be integrated into the CYBERWISER.eu Platform and will be leveraged for the training exercises. Besides, the asset itself will be evolved thanks to the participation in the project. The main work lines are listed below:

- Cross innovations
  - Improvement of the visualization capabilities, to provide a more attractive user interface.
  - Management of massive amounts of data and statistical process.
  - Multitenancy of high number of clients.
  - Recording of beginning and ending of anomalies (timestamps).
  - Containerisation to ease the deployment of the asset.
- Behavioural analysis
  - Avoid excess of information about alerts.

- Avoid false positives.
- Make reports and analysis simpler.
- Identify normal behaviour patterns to confront with abnormal ones.
- Use of machine learning techniques to achieve this.

## 5.12 Attack Simulator

AS is a component capable of automating attacks against targets in the simulated infrastructure using pre-defined scripts. It supports the following functionalities:

1. **Running pre-defined automated attacks.** As a part of scenario design (using TM), the entity designing the scenario configures the AS to run a particular attack script against some target. It's possible to configure the AS to run the script once (e.g. to schedule it at a particular point in the scenario) or on a recurring basis (e.g. to schedule it periodically with the desired interval). Once a training scenario is instantiated, the AS will automatically run the attack script, following the provided timeline and configuration.
2. **On-demand interventions by the white team.** A trainer closely following progress of the cyber-range exercise might want to intervene in the exercise to either aid the trainees, or to increase the difficulty on-the-fly. For instance, he might assess that the frequency at which an attack script runs is too low and decide to increase it to allow smoother progression of the exercise. He might even add new scheduled attacks or modify the existing ones. White team members control the AS through the Attack Configurator (a Web interface).
3. **Easier launching of automated attacks in red team exercises.** Knowledge and hands-on experience with offensive security and penetration testing (i.e. *pentesting*) software is essential for cybersecurity training. While in some exercises, trainees directly access such tools from their workstations, other exercises put more focus on risk assessment and don't require understanding of the (possibly complicated) attack specifics. AS can support those exercises, as it allows trainees to configure and launch attacks using pre-determined scripts and observe the infrastructure being attacked without a deeper knowledge of the attack itself.
4. **Access to attack script knowledge base in the red team exercises.** In the scope of an exercise, red team trainees must be given access to a set of *pre-defined* attack scripts or attack script templates. These can be used as a starting ground for configuring attacks, to avoid writing attack scripts from scratch.

The attack scripts, if so configured, can report about the success of attacks. When the attacks are run either by the CYBERWISER.eu Platform or by the white team (see 1. and 2. above), the information about success/failure of an attack is transferred to the PE, supporting the evaluation of blue team's success in defending the simulated infrastructure.

Figure 31 depicts a logical view of the component that can support the identified core functionalities, including its subcomponents and invocation paths. Details for each of the subcomponents and their interactions (required for interpretation of the figure) are provided below.

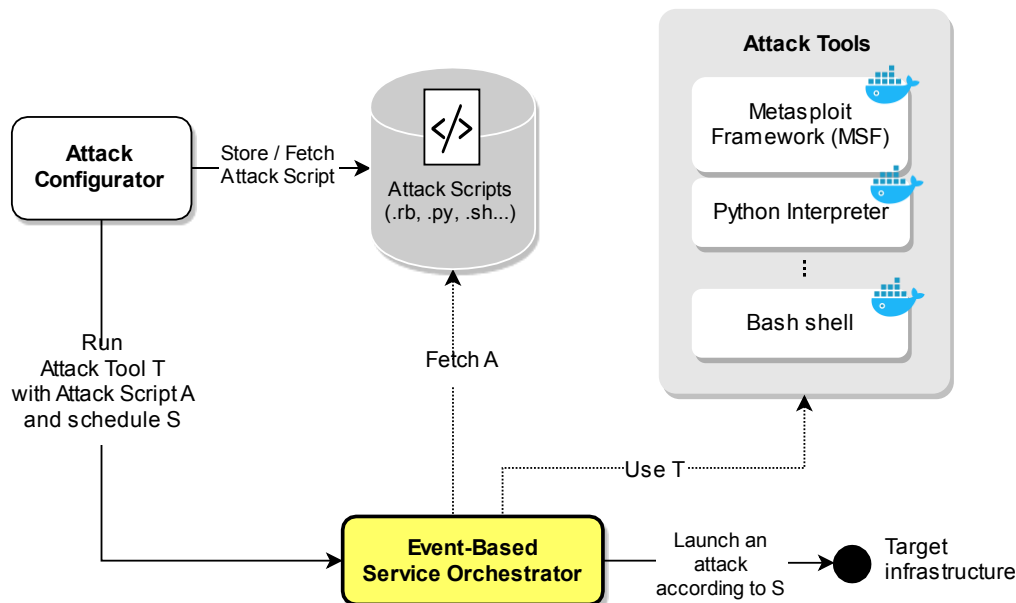


Figure 31. Logical view of the Attack Simulator

**Attack Script** is the description (source code) of an automated attack to be executed against the target infrastructure. Attack Scripts are written in one of the popular programming or scripting languages (for instance Ruby, bash). When an Attack Script outlines an attack but does not provide its complete description (for instance it has missing functionality or configuration), we instead talk about **Attack Script Templates**. These may not be executable, and the red team trainees need to modify them to produce valid Attack Scripts. During the training exercise, Attack Scripts and Templates must be stored in a place where EBSO can retrieve them. Technically, the component for storage of Attack Scripts is a part of the EBSO itself (see EBSO's back tier component Object Storage in Section 5.8). In addition, as designing automated attacks ahead of time (i.e. before the training scenario is running) is an integral part of the CYBERWISER.eu Platform, the attack scripts need have associated assets in the DL, so that attacks and their timelines can be configured in the TM.

**Attack Tools** power attacks. They are *lightweight* Docker images that package environments for running a particular kind of attack script. For instance, an example of an attack tool is an image with an installation of Metasploit Framework capable of running Metasploit exploits (scripts in Ruby language).

However, note that:

**Attack Tools are not to be confused with offensive security/pentesting software that may be installed on trainee's workstation** in the cyber-range exercise. Attack Tools abstract away, to the extent possible, the specifics of using said software. On one hand, this allows the red team trainee to focus on understanding the specifics of the attack (via access to the Attack Scripts) rather than learning how to use such software. On the other hand, it provides (to the CYBERWISER.eu Platform as well as trainer and red team trainee) the capability to run automated attacks in a simple manner.

Note that depending on the learning goals, some training scenarios may envisage the usage of Attack Tools through the AS (via Attack Configurator), while others require hands-on knowledge of offensive security tools available from the workstation.

**Attack Configurator** is a component with a Web interface to be utilized by trainers and trainees during the training scenario runtime. It allows management and configuration of all aspects of attacks, including the choice of an attack tool, attack scripts to run, and their scheduling. It also allows trainees to share their own attack scripts with their teammates in the scope of an ongoing training scenario

**Event-Based Service Orchestrator** is a backend component at the heart of the AS (please refer to Section 5.8 for details). In the context of the AS, EBSO oversees that attack scripts are launched against the target infrastructure according to a given timeline and configuration. When attacks originate from the CYBERWISER.eu Platform, meaning that they are known in advance as they are a part of a training scenario, the configuration is file-based. When the training scenario is up and running, configuration can also be provided on-the fly by the trainers and trainees through the Attack Configurator that communicates it to EBSO via AMQP.

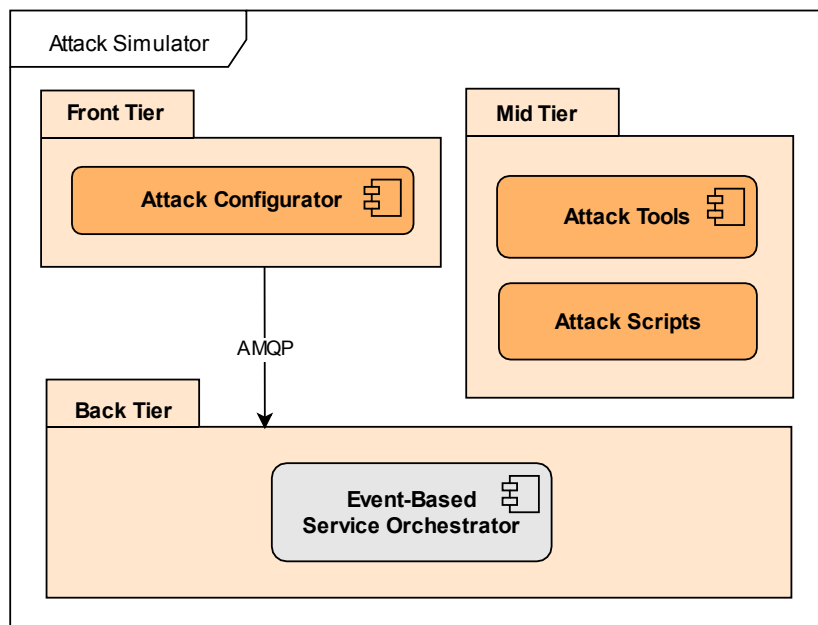


Figure 32. Attack simulator component diagram

Figure 32 depicts a tiered view of the AS component. The front tier comprises user-facing component for configuration and management of attacks. The mid-tier encapsulates core business logic of the AS, namely the tooling for running the attack scripts as well as attack scripts themselves. The back tier is represented by the EBSO component that schedules and orchestrates the business logic (attacks). Note that the front tier communicates directly with the back tier to configure and execute the logic in the mid-tier. Due to the specifics of the presented design, the mid-tier is represented as unconnected in the figure above.

### 5.13 Countermeasures Simulator

The CS will be offered to the trainees participating in the blue team as one of the applications deployed on the scenario in question.

The purpose of the CS is two-fold:

- It will be used when the red team role is played by human/s and the blue team role corresponds to the system.
- It will be used when the blue team is human, but they need support regarding the application of countermeasures (especially beginners' case).

The tool needs to store several data relevant for its internal operation which is provided by means of the contextualization process. Among this information we will have:

- The different kind of incidents each scenario element is exposed to in this exercise, in other words the kind of attacks they may suffer in the course of the exercise.
- The kind of actions that can be carried out on each scenario element, in this exercise, aiming at mitigating the effects of the attack.

- The price to pay to apply these actions.
- A rating linked to each action.
- The list of IPs and type of elements such IPs correspond to.

This been explained, the logical sequence (see Figure 33) this tool would follow is presented below:

- 1) The ADR sends through the MB information about an alarm notifying an incident for a certain scenario element with a certain IP.
- 2) The CS queries the database to know which scenario element corresponds to that IP.
- 3) Once it knows the kind of scenario element in question, it queries the database to know the kind of mitigation actions that scenario element can be applied.
- 4) With this information, it queries the database again to find which mitigations out of the obtained in the previous step would be applicable to the type of incident detected.
- 5) There will be two possibilities in this step:
  - a. If the blue team role is played by a human, the user will choose a mitigation among the available ones, and in the backend a script will be executed. These scripts will be fed the needed information leveraging the existing data about the scenario.
  - b. If the blue team role is played by the platform, this process will be performed automatically and the platform will use some criteria (cost, rating, best value for money) to choose what mitigation to go for.

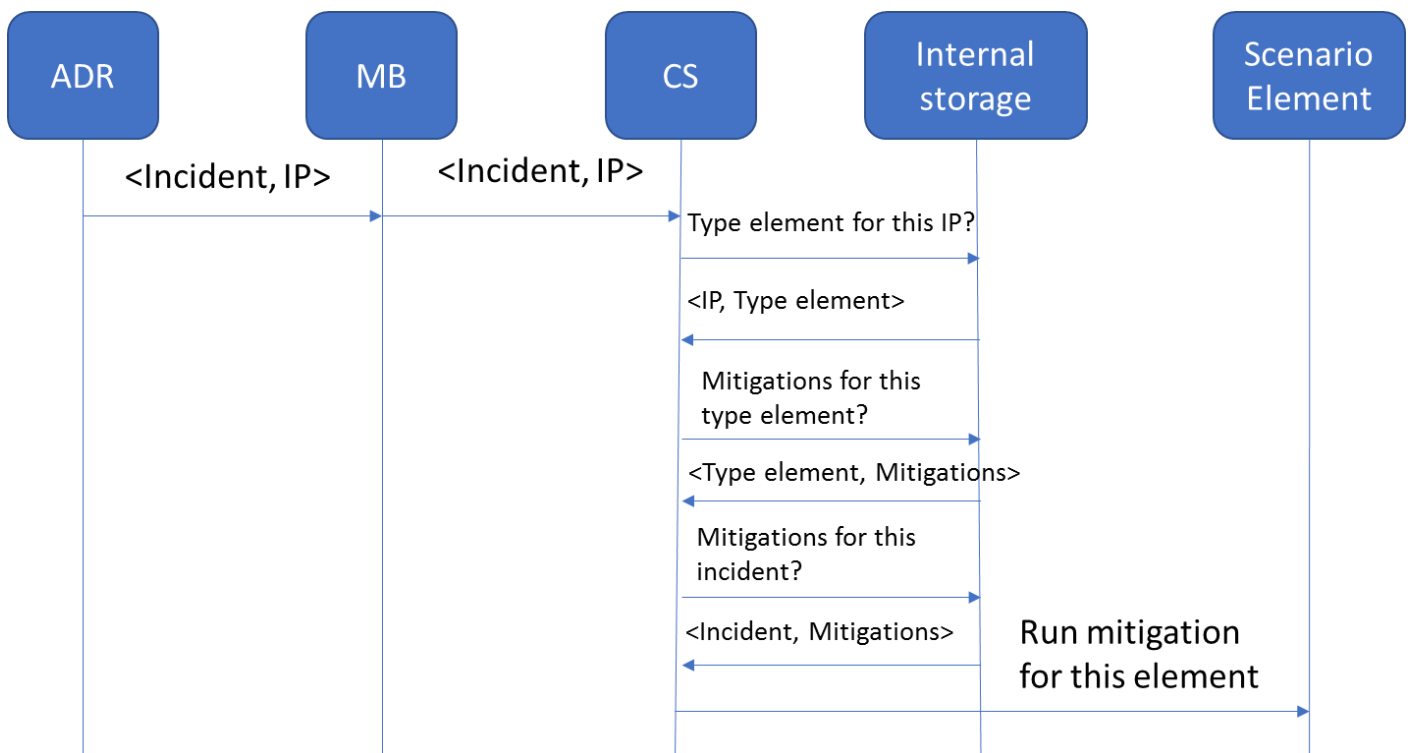


Figure 33. Concept of Countermeasures Simulator

Figure 34 represents the inter-relation of the concepts involved for mitigation identification. Figure 35 represents how the idea explained above is applied in a design. The green boxes inside the area marked by

the dotted line represent the internal components of the Countermeasures Simulator. The SIM provides contextualization information needed for the internal operation of this component. In particular, this contextualization indicates if the component will work in a full-automatic mode or the user will interact with it. Also, the internal storage is fed with this information: list of pairs <IP, type of element>, list of pairs <IP, incident>, list of pairs <Type of element, mitigation>, list of pairs <incident, mitigation> and list of three-element-groups <mitigation, cost, rating> for all the simulated network elements. The relations among these concepts are depicted below:



Figure 34. Inter-relation of concepts for mitigation identification

The attacker launches an attack against an element of the simulated network. By means of the sensors deployed, this attack is detected and sent to the agent, which produces an event that is sent to the ADR. The ADR will use the MB to transmit an alarm that indicates the kind of incident it is about and the IP undergoing such situation. The *Mitigation Identifier* uses this information to query the database following the process explained in Figure 34 to obtain the possible mitigation/s that can be applied. If the CS is working on a fully-automated mode, it passes this information to the *Mitigation Selector*. This module consults the database to establish a priority list with the possible mitigations. The criterion to follow<sup>60</sup> is configurable by contextualization and can be: 1) cost; 2) rating; 3) best value for money, combining information about the cost of the mitigation and the rating. Once the mitigation has been selected, the *Mitigation Executor* both sends logs to the CLC via the MB and prepares the call to the script, passing the needed parameters for its right execution. If the CS is used by human defenders, there will be a user interface that will show the identified mitigations and the related costs and rating. Then the user will pick one of the mitigations and through the user interface he will pass the needed parameters for execution. The user himself will run such execution. The level of difficulty the user will face can be configured by contextualization and considers the following cases:

<sup>60</sup> More criteria may be added if a need for it is identified in the course of the project

- Level 0: The CS will show the mitigations that apply to the machine and the incident under question, along with their corresponding price and rating. This way the user has all the needed information to choose.
- Level 1: The CS will show the mitigations that apply to the machine and the incident under question, but no information about price and rating will be provided. The user will have to apply his own knowledge to prioritize and choose the best one/s and his decisions will impact in a lowering of the budget.
- Level 2: The CS will show not only the mitigations that apply to the machine and the incident under question, along with their corresponding price and rating, but also some others that do not apply to (possibly) confound him and increase the difficulty.
- Level 3: The CS will show not only the mitigations that apply to the machine and the incident under question, but also some others that do not apply to (possibly) confound him and increase the difficulty. On top of that, no information about price and rating will be given.
- Level 4: This means directly not to deploy the CS, this is, not to make it available.

It is also relevant to remark that not all the mitigation measures that will be proposed will have a script ready to be executed. In some cases this is not possible and the user has to proceed manually. This does not mean that the mitigation cannot be suggested by the CS. The difference is that there will not be any script ready for execution.



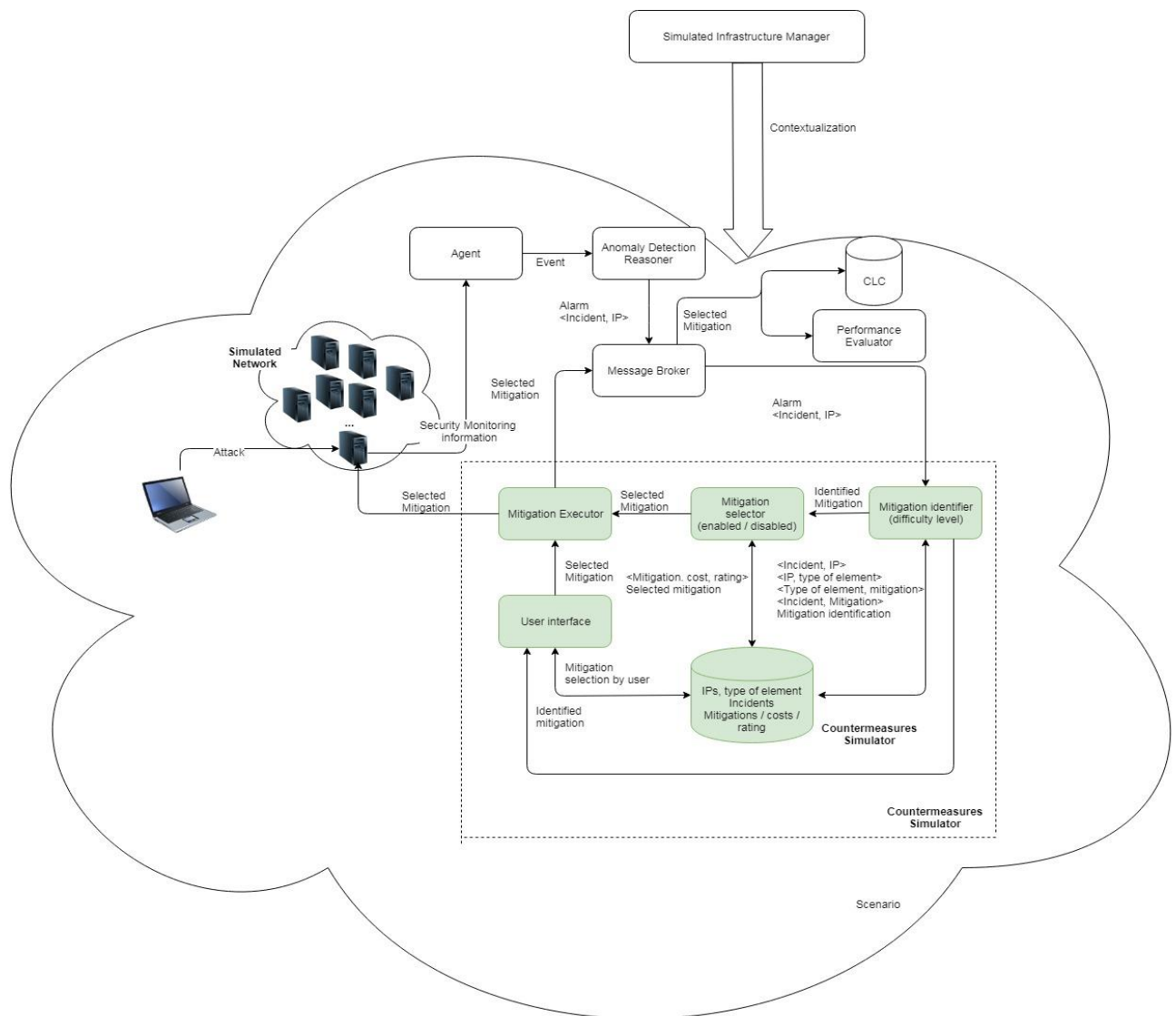


Figure 35. Design to apply the Countermeasures Simulator concept

The figure below shows the translation of the concept into an internal functional design.

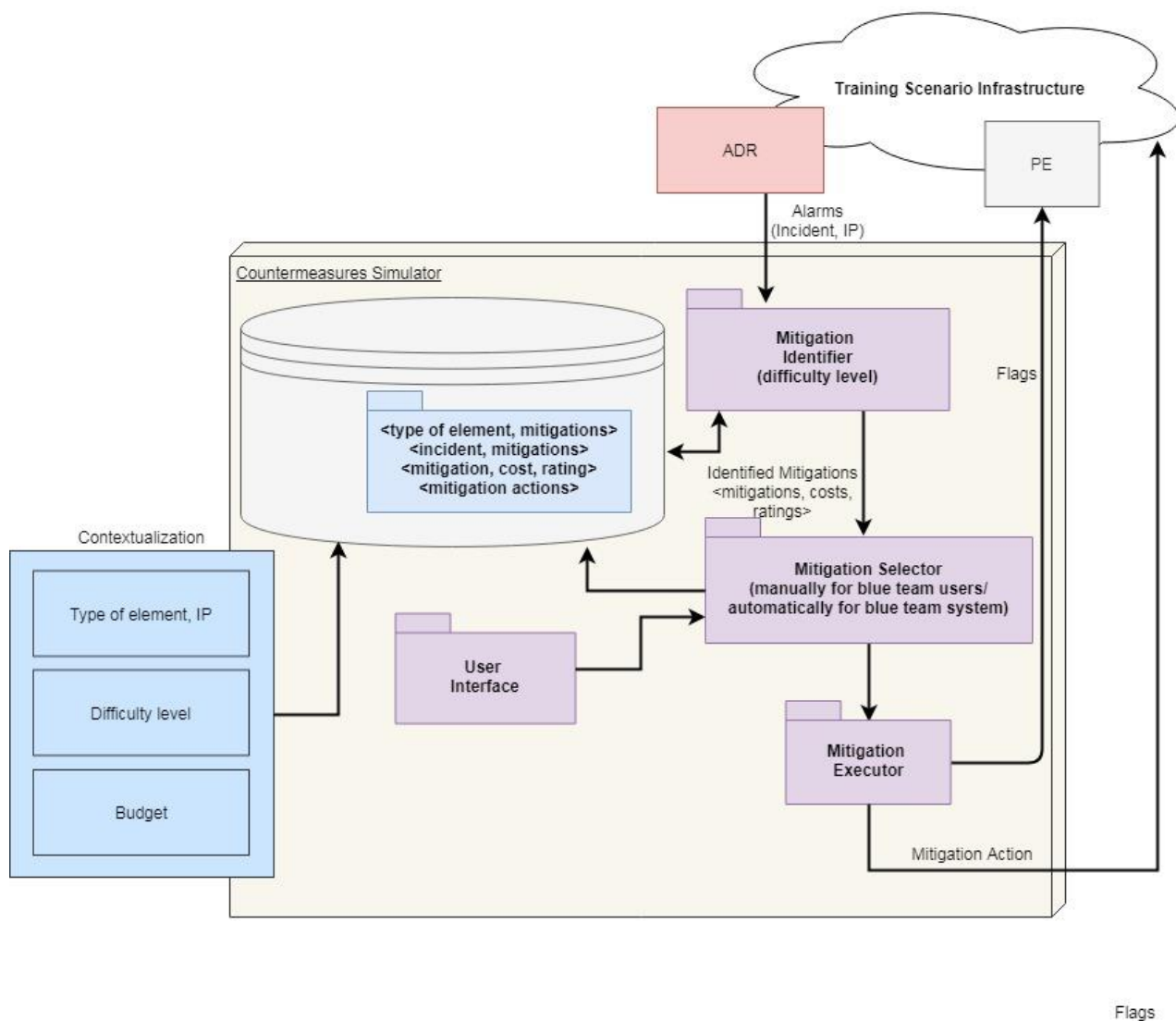


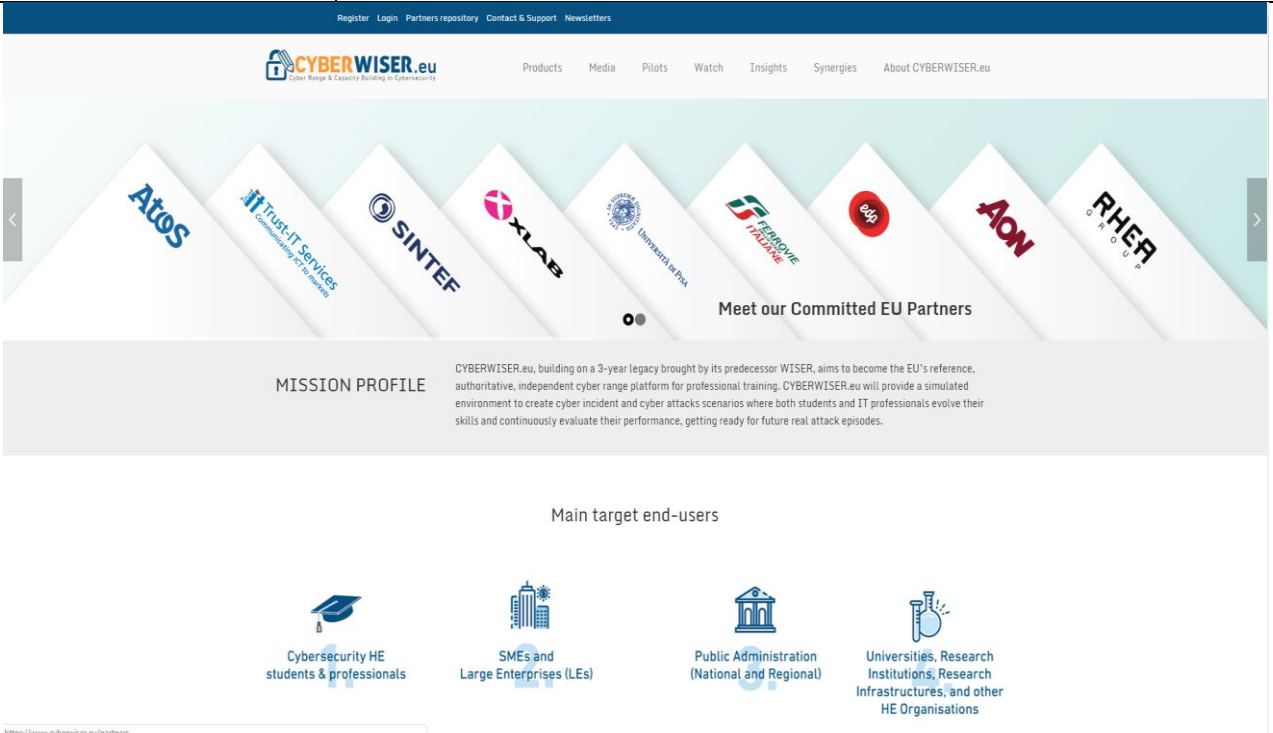
Figure 36. Internal functional design of the Countermeasures Simulator

## 6. Using the platform

In this section, the main relevant user interfaces are documented, as well as the relations existing among them (section 6.1). In addition, a list of representative use cases is included in section 6.2. In both cases we have followed a template, presented in section 2.3, to produce the contents in a systematic, methodical and structured way.

### 6.1 User flows through the user interface

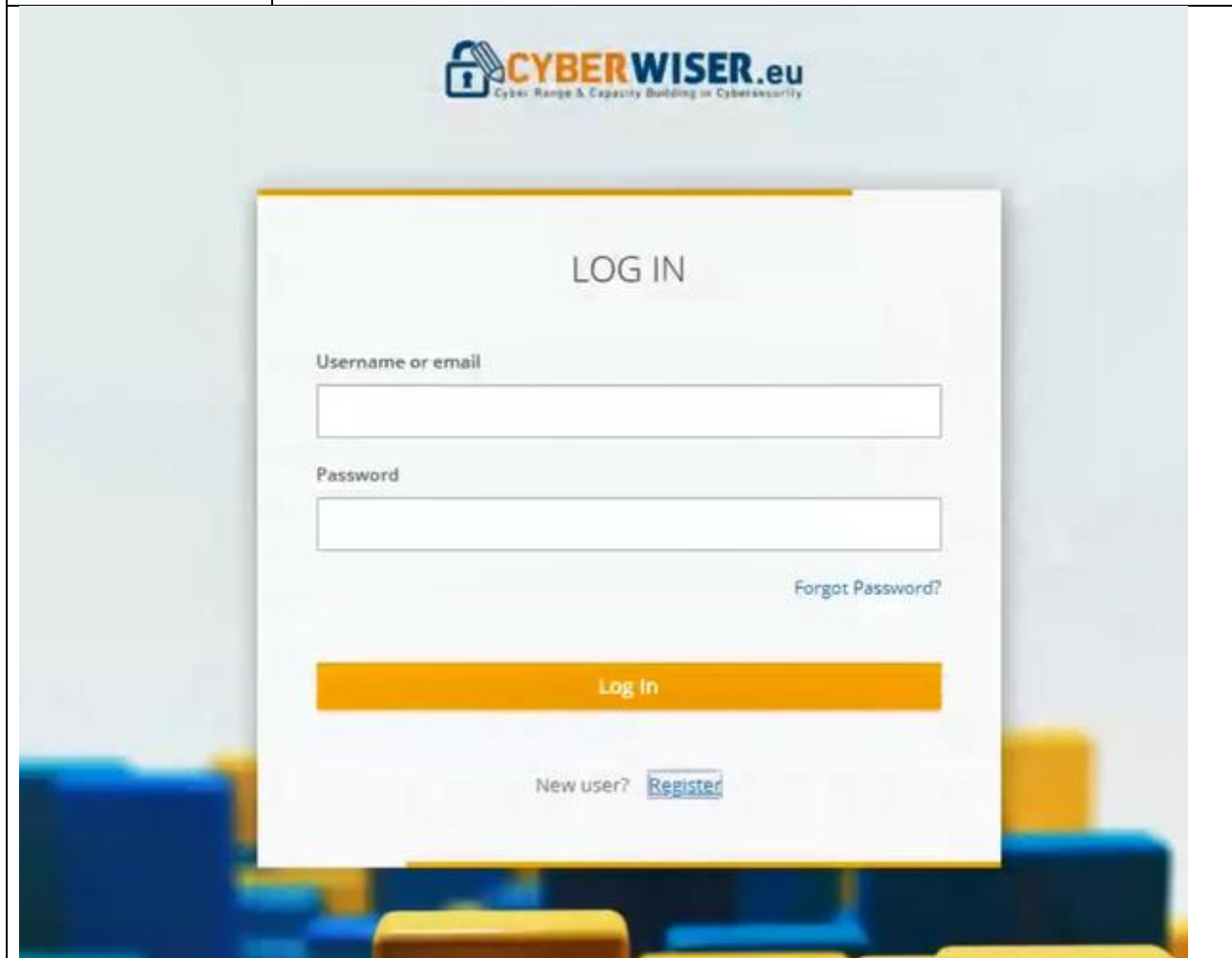
User Interface name	CYBERWISER.eu website
UI Id	UI.01.000
Description	This is the portal website, with all the information related to the project and is the on-line access point to the cybersecurity training platform.
Purpose	Following the model agreed on the Consortium for the provision of services, these are accessed through the project website.
Navigation and user interaction	<p>Among the different links available with all the information about the different aspects of the project, one is entitled "Login" and takes to the CYBERWISER.eu Login page which is required to access the CYBERWISER.eu platform.</p> <p>In the coming months the website will be enriched with dedicated sections about the CYBERWISER.eu platform which will explain the different Offering Levels available, their features and benefits from a trainee's perspective. This is necessary to ensure a proper dissemination and go-to-market strategy for the CYBERWISER.eu platform.</p>
Other comments	This interface presents to both trainees and trainers the same graphical layout and functionalities.



The screenshot shows the homepage of the CYBERWISER.eu website. At the top, there is a navigation bar with links: Register, Login, Partners repository, Contact & Support, and Newsletters. Below this is a header section with the CYBERWISER.eu logo and a list of menu items: Products, Media, Pilots, Watch, Insights, Synergies, and About CYBERWISER.eu. The main content area features a large banner with the text "Meet our Committed EU Partners" and a carousel of logos for various partners including Atos, IT Trust-IT Services, SINTEF, XLAB, Universitat de Pisa, Fedgovve, AOW, and RHEA. Below the banner, there is a section titled "MISSION PROFILE" which describes the platform's goals and its role as a reference for professional training. At the bottom, there is a section titled "Main target end-users" with four icons representing different user groups: Cybersecurity HE students & professionals, SMEs and Large Enterprises (LEs), Public Administration (National and Regional), and Universities, Research Institutions, Research Infrastructures, and other HE Organisations.

Table 7. UI.01.000 – CYBERWISER.eu website

User Interface name	CYBERWISER.eu Login page
UI Id	UI.01.001
Description	In the centre, a frame that contains two text boxes: one to type the login text and another to type the password. Under these, a button named "login". On top of the frame containing the two text boxes and the button, the CYBERWISER.eu logo appears. Inside the login frame there are two links, one entitled "Forgot password?" utilized to get a new password, and another one entitled "Register" which is utilized in case the user does not have user / password and they have to be created. The background is composed using colours closely related to those of the project branding.
Purpose	This interface has the goal of being the point to introduce the credentials to access the platform and its services (will be reachable on-line within the CYBERWISER.eu portal <a href="https://www.cyberwiser.eu">https://www.cyberwiser.eu</a> ), according to the permission level granted to the user.
Navigation and user interaction	The user introduces his credentials, if they are correct he is taken to the private part of the portal and if not he is requested to introduce his credentials, with a limit of attempts.
Other comments	This interface presents to both trainees and trainers the same graphical layout and functionalities.



The screenshot shows the CYBERWISER.eu login interface. At the top, the CYBERWISER.eu logo is displayed. Below it, the text "LOG IN" is centered. There are two input fields: "Username or email" and "Password". Below the password field is a link "Forgot Password?". A large orange button labeled "Log In" is positioned below the input fields. At the bottom, there is a link "New user?" followed by a "Register" button. The background is a blurred image of blue and yellow blocks.

Table 8. UI.01.001 – CYBERWISER.eu Login page

User Interface name	CYBERWISER.eu private area
UI Id	UI.01.002
Description	<p>In this private area of the portal, several links are enabled to the user, depending on the user profile. Among them, on the top left part, the user will see the banners of those products for which he has been granted access. Among them, one that may appear is the banner linking to the Cross Learning Facilities through which both the training resources and the cyber range itself are available to the user, always depending on the offering level he went for.</p> <p>At the moment of writing the same region of the interface hosts the Cross-Learning facilities button as well as the buttons for the services produced by the CYBERWISER.eu predecessor WISER. To avoid confusion, these will be moved in another region of the interface before the official launch of the CYBERWISER.eu Platform.</p> <p>It must be stressed that the label "Cross Learning Facilities" is to be considered just a placeholder as this will probably be renamed in a more market-oriented way to support the go-to-market strategy.</p>
Purpose	The purpose of this interface is to make available resources addressed to the members of the CYBERWISER.eu Community upon authentication of the user, with a personalized offer linked to the level of engagement
Navigation and user interaction	Among the different links available, if the user clicks on the "Cross Learning Facilities" banner on the top part, he is taken to the training main menu with no need to introduce credentials thanks to the SSO system.
Other comments	This interface presents to both trainees and trainers the same graphical layout and functionalities.

Table 9. UI.01.002 – CYBERWISER.eu private area

User Interface name	CYBERWISER.eu workspace area
UI Id	UI.02.001
Description	<p>In this page a user can see the groups he is registered in. Depending on the specific group/s a user belongs to, he will see on the right column the types of resources allowed to the group: the e-learning platform, the cyber range platform and a file repository that can be used to store documents or files.</p> <p>It must be stressed that the label “Workspace” is to be considered just a placeholder as this will probably be renamed in a more market-oriented way to support the go-to-market strategy.</p>
Purpose	<p>The workspace area is a summary of the different groups in which a user is involved, and links to the different resources that the user is made available as a consequence of belonging to a specific group. There is a dependency between the resources that are made available and the group in question</p>
Navigation and user interaction	<p>By clicking on a group a user is taken into a specific area in which he can see:</p> <ul style="list-style-type: none"> <li>• Group details.</li> <li>• Messages from other users (both trainers and trainees) in the same group. When a message in a group is posted each user will receive an automatic email notification.</li> <li>• Link to the e-learning platform (My Training Path label) handled with the SSO so there's no need for the user to do another login process.</li> <li>• Link to the file repository handled with the SSO so there's no need for the user to do another login process.</li> <li>• Link to the group's email address.</li> <li>• Actions that can be performed by each user i.e. post a message or an event.</li> <li>• List of the people included in the group.</li> </ul>
Other comments	<p>This interface presents to both trainees and trainers the same graphical layout and functionalities.</p>

User Interface name	CYBERWISER.eu workspace area
	<p>The screenshot displays the CYBERWISER.eu workspace interface. On the left, a sidebar lists three groups: EDP - FSP3, FFSS - FSP2, and UNIFI - FSP1. The main area shows a detailed view of the FFSS - FSP2 group. This view includes the group's logo (Ferrovie dello Stato Italiane), a description of the pilot project, a 'Recent Activity' section with a 'Test New Post' entry, and a 'Members' list showing 'Rec every' and 'Mario Rossi'. A right-hand sidebar contains links to 'My Training Path', 'File Repository', and 'Add new content'.</p>

Table 10. UI.02.001 – CYBERWISER.eu workspace area



User Interface name	CYBERWISER.eu Moodle dashboard – Trainee perspective
UI Id	UI.02.002
Description	<p>This is the main dashboard of the e-learning platform.</p> <p>It firstly presents the CYBERWISER.eu offering level(s) to which a user is subscribed to. If the users already performed some of the courses inside an offering level a percentage of completion for the entire offering level is given here as a summary for the user.</p> <p>The user can also see at a glance offering levels (and as a consequence courses) in progress and taken in the past.</p>
Purpose	<p>This interface is the summary of the course assignments the user has, listing the ongoing offering levels (reflecting the % of progress) and those completed in the past.</p>
Navigation and user interaction	<p>When a user clicks on an offering level, a detailed view is shown with the list of all the courses available in that specific offering level. At the moment of writing this deliverable, partners are still discussing if quizzes (or exercises) and certification of completion of the courses should be shown here.</p> <p>The organization and hierarchy of the courses and relative modules is shown as defined into D4.1.</p> <p>A user can simply click on a course to enter it. Some courses are composed by more than one module so in this case an intermediate pop up page presenting the module is shown to the users. When a user successfully completes a course this is indicated in a graphical way in the list of courses and the percentage of completion of the offering level is shown as a summary.</p>
Other comments	

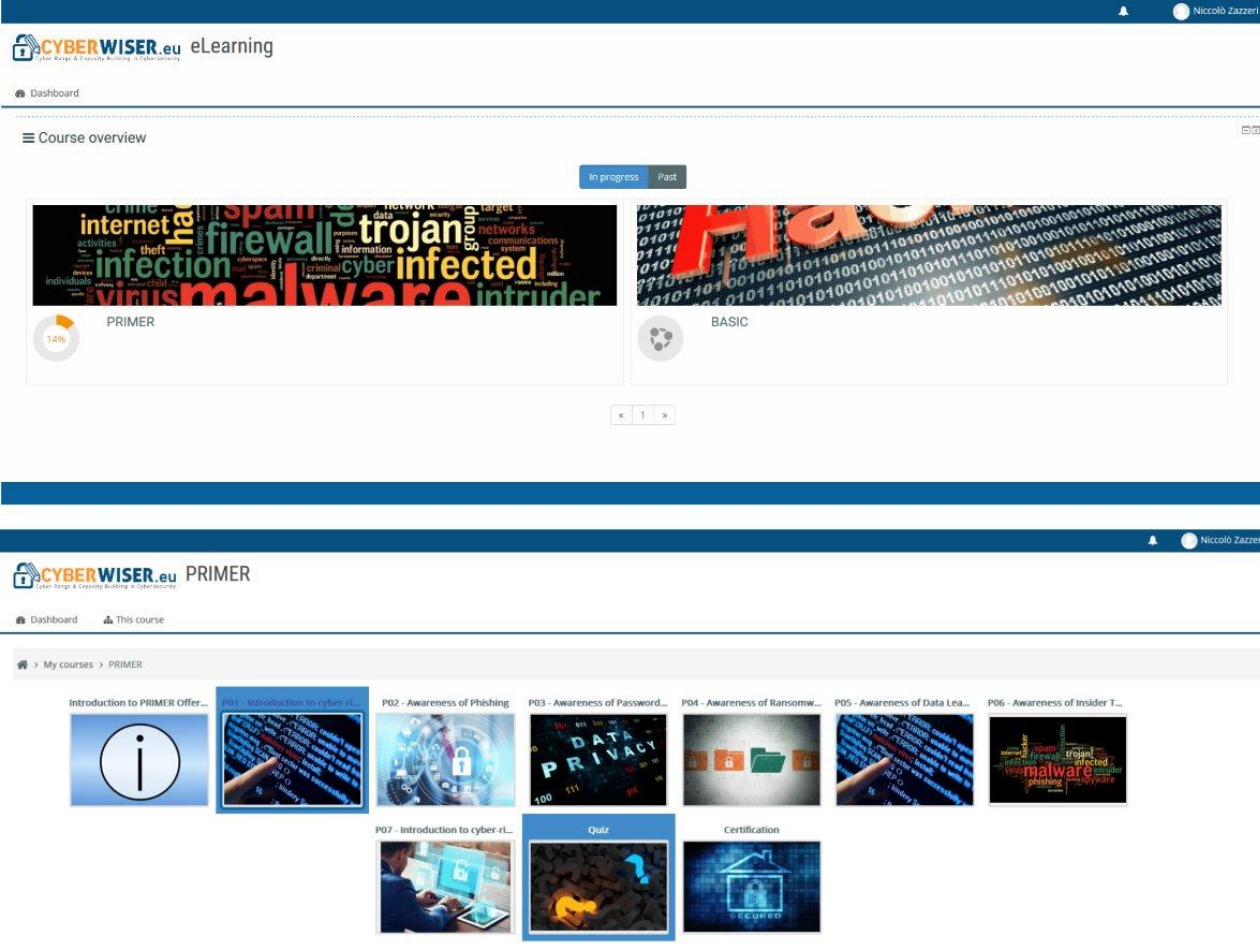
User Interface name	CYBERWISER.eu Moodle dashboard – Trainee perspective
	

Table 11. UI.02.002 – CYBERWISER.eu Moodle dashboard – trainee perspective.

User Interface name	CYBERWISER.eu Moodle dashboard – Trainer perspective
UI Id	UI.02.003
Description	<p>The graphical layout is the same as for the trainee perspective but with added functionalities.</p> <p>Just as for the trainee perspective, the interface presents the CYBERWISER.eu offering level(s) to which a trainer is assigned to. The trainer can see at a glance offering levels (and as a consequence courses) in progress and taken in the past. The real changes in respect to the trainee perspective are inside the offering levels and courses in which the trainer is presented with buttons and functionalities to shape the offering levels, the courses and monitor the trainee's activities.</p>
Purpose	<p>The interface lists the ongoing offering levels and those done in the past to which a trainer is assigned to.</p> <p>The interface also allows the trainer to shape a course (i.e. adding descriptive texts and images, uploading training materials, exercises etc.) and monitor the progress of the trainees.</p>
Navigation and user interaction	<p>Just as for the trainee's perspective, when a trainer clicks on an offering level, a detailed view is shown with the list of all the courses available in that specific offering level. The trainer is presented with some administrative functionalities. At the moment of writing this deliverable, the interface presents a lot of functionalities that may be restricted or cancelled to ease the trainer user experience at a later stage. Anyway, we list here below the basic functionalities that will be surely present in the final version of the interface and that allow the trainer to:</p> <ul style="list-style-type: none"> <li>• Edit a single course and relative modules in all its aspects so name, description, image, training materials and exercises.</li> <li>• Manage the organization and hierarchy of the courses and relative modules.</li> <li>• Monitor the trainees progresses. Many ways of monitoring are possible: a trainer can view the completion of a single module, the completion of a single course, can have an overview of the progress of all the trainees or a single trainee overview.</li> </ul>
Other comments	

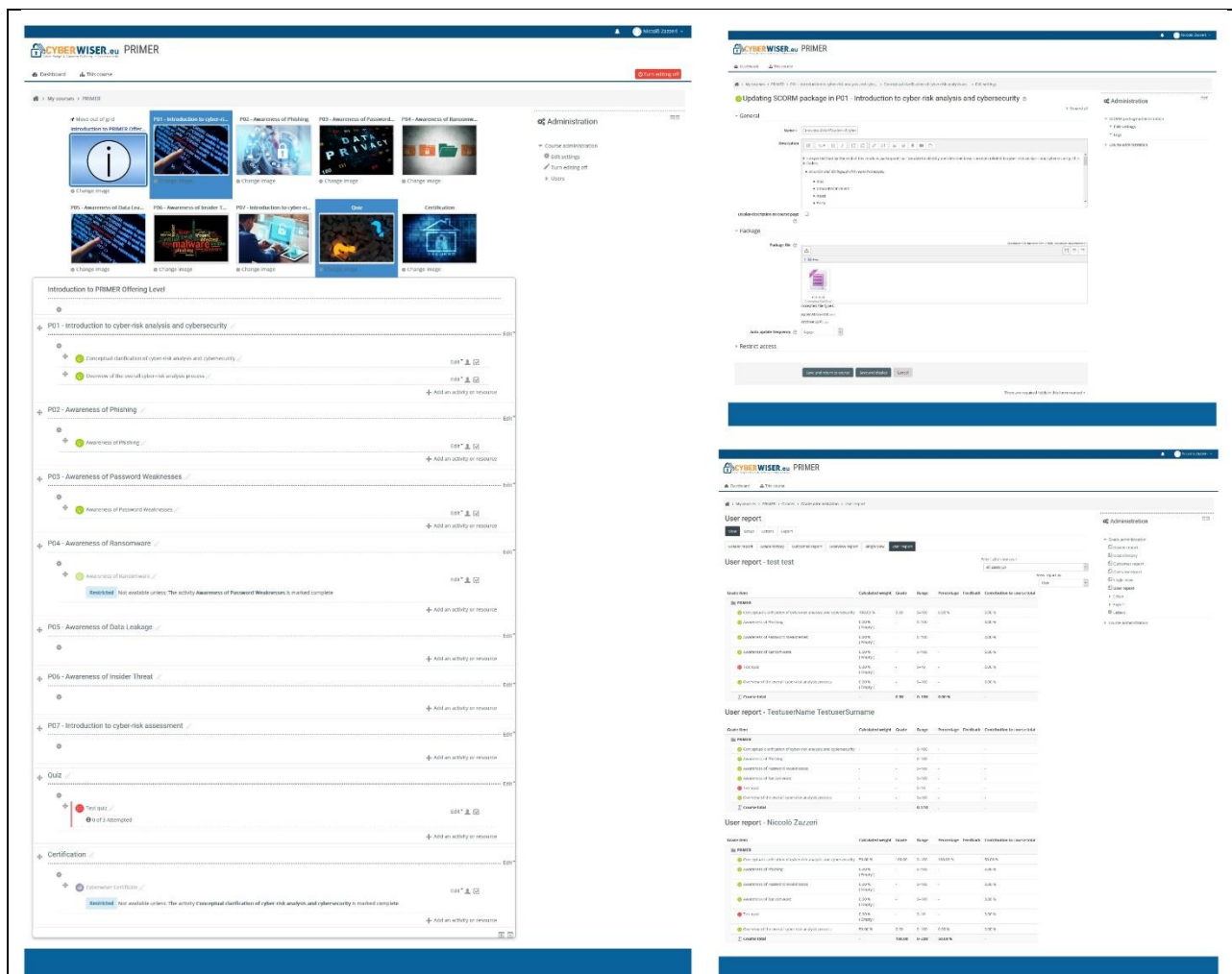



Table 12. UI.02.003 - CYBERWISER.eu Moodle Dashboard – Trainer perspective


User Interface name	Simulated Infrastructure Manager
UI Id	UI.03.000
Description	
Purpose	
Navigation and user interaction	
Other comments	All main interfaces are presented in the following. No other (explicit) ones for the SIM. SIM features are integrated in the TM as far as user interfaces are concerned.

Table 13. UI.03.000 – Simulated Infrastructure Manager

User Interface name	CYBERWISER.eu Scenario listing page
UI Id	UI.04.001
Description	In this page the user will see the listing of all scenarios he or she access to. Depending on user permissions, different actions will be either available or hidden.
Purpose	The purpose of this screen is to allow the user to see to which scenarios they have access and to either open a specific scenario or create a new one.
Navigation and user interaction	The user can navigate to all visible scenario, opening them in the editor. Qualified users can also navigate to the page for creating a new scenario or alternatively, editing an existing one. The user can also navigate to the version history of the scenario creation.
Other comments	



CYBERRANGE



ADMINISTRATION

Users

SERVICE

Services

ACTIVITIES

Activities


SCENARIO

Scenarios

CONFIGURATION

Digital Library

Administrator Adm





 Scenario

Scenarios

Scenarios

Add

To start search items type at least 3 characters and press Enter

Name	Version	Created	Type	Status	Diagram	Actions
DemoScenario	A1	Jun 24, 2019, 9:06:05 AM	TRAINING	VALIDATED	Edit	 
testGabBusinessScenario	A1	Jun 18, 2019, 2:59:54 PM	BUSINESS	CREATING	Edit	 

2 total

Powered By DUNE Group

Table 14. UI.04.001 – CYBERWISER.eu Scenario listing page

User Interface name	CYBERWISER.eu Scenario versions page
UI Id	UI.04.002
Description	In this page the user will see the listing of all scenario versions of the chosen scenario.
Purpose	The purpose of this screen is to allow the user to see the version history of changes to the scenario.
Navigation and user interaction	The user can navigate to all of the older versions and the current version. Old versions can be deleted by eligible users.
Other comments	

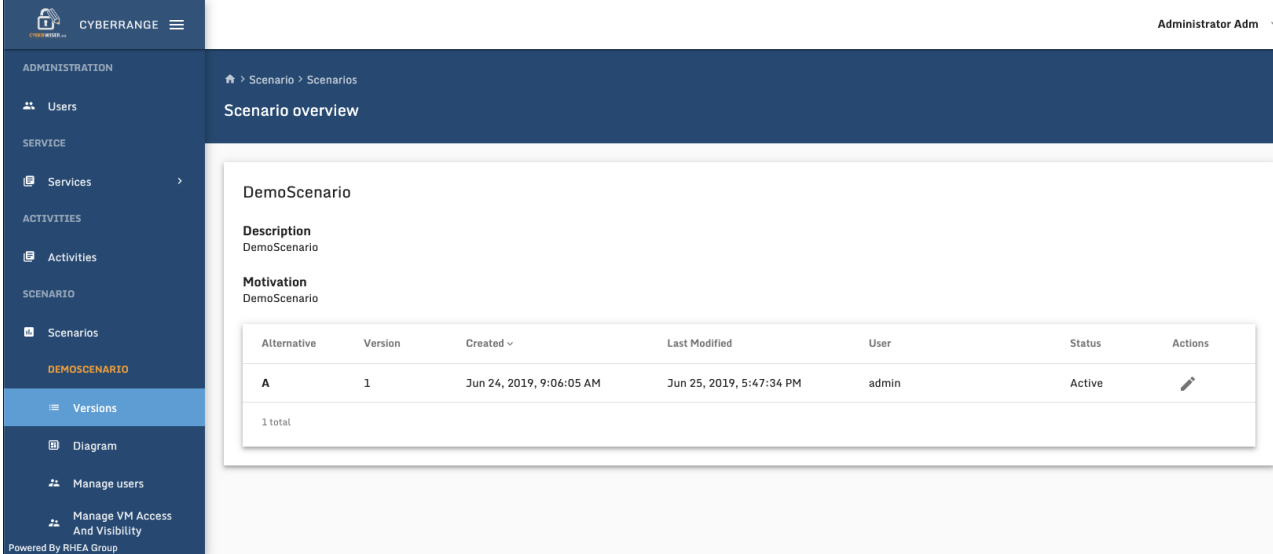
 <p>The screenshot shows the 'Scenario overview' page for 'DemoScenario'. It includes a sidebar with navigation options like 'Users', 'Services', 'Activities', 'Scenarios', and 'Versions'. The main content area displays the scenario details and a table of versions. The table has columns for Alternative, Version, Created, Last Modified, User, Status, and Actions. There is one version listed: Alternative A, Version 1, created on Jun 24, 2019, 9:06:05 AM, last modified on Jun 25, 2019, 5:47:34 PM, by user 'admin', with status 'Active'. A '1 total' summary is shown below the table.</p>	
---	--

Table 15. UI.04.002 – CYBERWISER.eu Scenario versions page

User Interface name	CYBERWISER.eu Scenario editor page
UI Id	UI.04.003
Description	<p>In this page all of the details of a specific scenario can be viewed and managed, all based on the users' permissions. In design mode, the scenario can be constructed using the palette on the left-hand side, dragging in the required elements of the network, the application landscape, and the training scenario. Details for each node can be managed or viewed using the panels that appear on the right side of the screen when a node is selected. These panels differ depending on what kind of node is selected. For virtual machine nodes, the asset can be selected, the VM dimensions can be specified as well as the network details for each of the connected network nodes. For network nodes the network information can be viewed or managed.</p> <p>In view mode, or when the scenario is already instantiated the design view is used to see the overall status of the different virtual machines in the network layer as well as allowing users to access those VM that they have been granted access to.</p>
Purpose	The purpose of this screen is to provide a complete management or insight of a single scenario.
Navigation and user interaction	From this screen, when the scenario is instantiated, authorized users can open up the VNC access window to selected VMs. The network details page can be opened, as well as the network policy window.
Other comments	

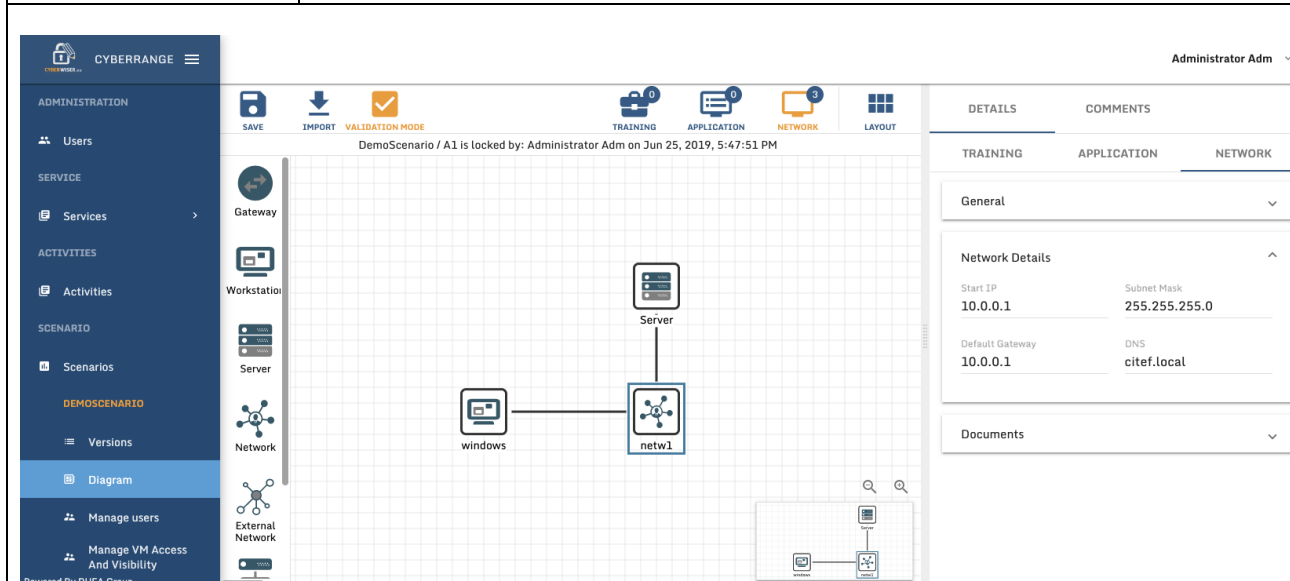


Table 16. UI.04.003 - CYBERWISER.eu Scenario Editor Page



User Interface name	CYBERWISER.eu – Network policies
UI Id	UI.04.004
Description	<p>In the network policies screen, the system recognizes the network configuration and provides the right node names automatically so that a user can create a row that defines a filtering action selecting the node names from dropdown menus.</p> <p>New filtering actions can be created by setting the filters in the dropdown menus and pressing “Add”. A new row would be added.</p> <p>Can be deleted by pressing delete (last column) on the corresponding row.</p> <ul style="list-style-type: none"> <li>• <b>Source Network:</b> the source network name</li> <li>• <b>Source VM:</b> the name of the source node (Workstation, Server or Gateway) of the message</li> <li>• <b>Target Network:</b> the target network name</li> <li>• <b>Target VM:</b> the name of the target node (Workstation, Server or Gateway) of the message</li> <li>• <b>Action:</b> the action to perform (chosen between: None, Block Application, NAT out, Allow application, Allow all applications)</li> <li>• <b>Port:</b> the port number</li> </ul> <p>It is possible to close the Policies popup from the close button, every modification will be saved</p>
Purpose	The purpose of this screen is to express which network policies should be implemented in the selected virtual machine.
Navigation and user interaction	The user can close the popup.
Other comments	

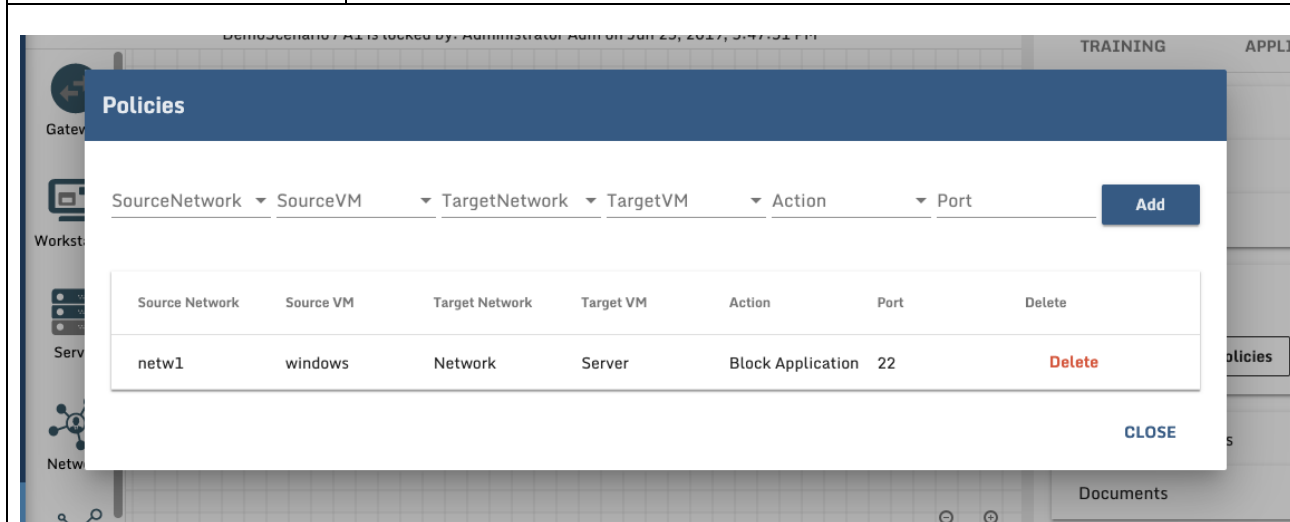


Table 17. UI.04.004 - CYBERWISER.eu – Network Policies

User Interface name	CYBERWISER.eu - VM network details
UI Id	UI.04.005
Description	<p>For each network to which node is linked, there is one panel with the Network name and a link to the network configuration. If clicked, the link opens a popup window where it is possible to see or set the node IP, subnet mask, Gateway and DNS. By default, the network settings of a Virtual Machine are by default set to generated settings based on the network configuration. Settings can be edited to be incompatible with said network, but in that case a warning is given in validation mode.</p> <ul style="list-style-type: none"> <li>• <b>IP address:</b> the IP of the node on that network.</li> <li>• <b>Subnet Mask:</b> the subnet mask in IP format (e.g. 255.255.255.0).</li> <li>• <b>Gateway:</b> the IP of the gateway, if present.</li> <li>• <b>DNS:</b> the IP of the Domain Name Server, if present.</li> </ul> <p>These field are automatically compiled as they are deduced by the diagram.</p>
Purpose	The purpose of the network details screen is to allow the users to control the IP network level settings of all of the VMs in the scenario, explicitly.
Navigation and user interaction	The user can close the popup
Other comments	
<div> <div>VM windows configuration for network netw1</div> <div> <div> <div>VM network configuration</div> <div> <div>IP address</div> <div>10.0.0.1</div> </div> <div> <div>Subnet mask</div> <div>255.255.255.0</div> </div> <div> <div>Gateway</div> <div>10.0.0.1</div> </div> <div> <div>DNS</div> <div>citef.local</div> </div> </div> <div> <div>Close</div> <div>Apply</div> </div> </div> </div>	

Table 18. UI.04.005 – CYBERWISER.eu – VM network details

User Interface name	CYBERWISER.eu – Scenario user permissions
UI Id	UI.04.006
Description	<p>Scenario Creator can grant other users a wide array of different permissions on the scenario including the ability to manage users and their permissions (see ASSIGN_USERS permission below). User management can be accessed by clicking “Manage users” menu item of corresponding Scenario Environment.</p> <p>To assign a user to the scenario, the user must be selected from the unassigned users list, at least one permission must be selected and then the up arrow button must be clicked. Permissions can be changed for an assigned user by selecting the user from assigned users list, selecting appropriate permissions and then clicking “Save” button. Permissions are given on the entire scenario, and are as follows:</p> <ul style="list-style-type: none"> <li>• ALTERNATIVE_EDITING: grants the ability to edit scenario diagram.</li> <li>• ALTERNATIVE_VALIDATING: grants the ability to validate scenario.</li> <li>• ALTERNATIVE_DELETING: grants the ability to delete a saved scenario alternative.</li> <li>• SCENARIO_DETAILS_UPDATE: grants the ability to update the details of a scenario (name, description, etc.).</li> <li>• NETWORK_ORDER: allows manual reordering of network interfaces (corresponding to connected networks) of a virtual machine.</li> <li>• VIEW: users with this permission will see a scenario in the scenario list and be able to see all nodes of the scenario diagram. VNC access to every VM can be enabled on a per-node's instance basis or via the FULL_VNC_ACCESS permission (see below) if desired.</li> <li>• VIEW_DOCUMENT: allows to see scenario's linked documents.</li> <li>• LINK_DOCUMENT: grants the ability to link a document to a node, requires VIEW_DOCUMENT permission.</li> <li>• VIEW_COMMENT: allows to see scenario's comments.</li> <li>• COMMENT: grants the ability to post a comment to the scenario, requires VIEW_COMMENT permission.</li> <li>• ASSIGN_USERS: grants the ability to assign users to the scenario and define their permissions.</li> <li>• FULL_VNC_ACCESS: grants access to all virtual machines of the instantiated scenario.</li> <li>• FINALIZED_SCENARIO_VIEW: this permission automatically excludes all other permissions and makes only validated scenarios visible to the user. Furthermore, visibility of every node of the diagram needs to be explicitly enabled for users with this permission. VNC access to every VM can be enabled on a per-node's instance basis if desired. This permission could be useful to assign to a trainee user in order to restrict access and visibility of the scenario.</li> </ul>
Purpose	The purpose of this screen is to allow users to manage which users have which authorizations with respect to the functionality they can use for a specific scenario.
Navigation and user interaction	Using the left menu, the user can switch to the scenario listing, the editor, or the version history.
Other comments	

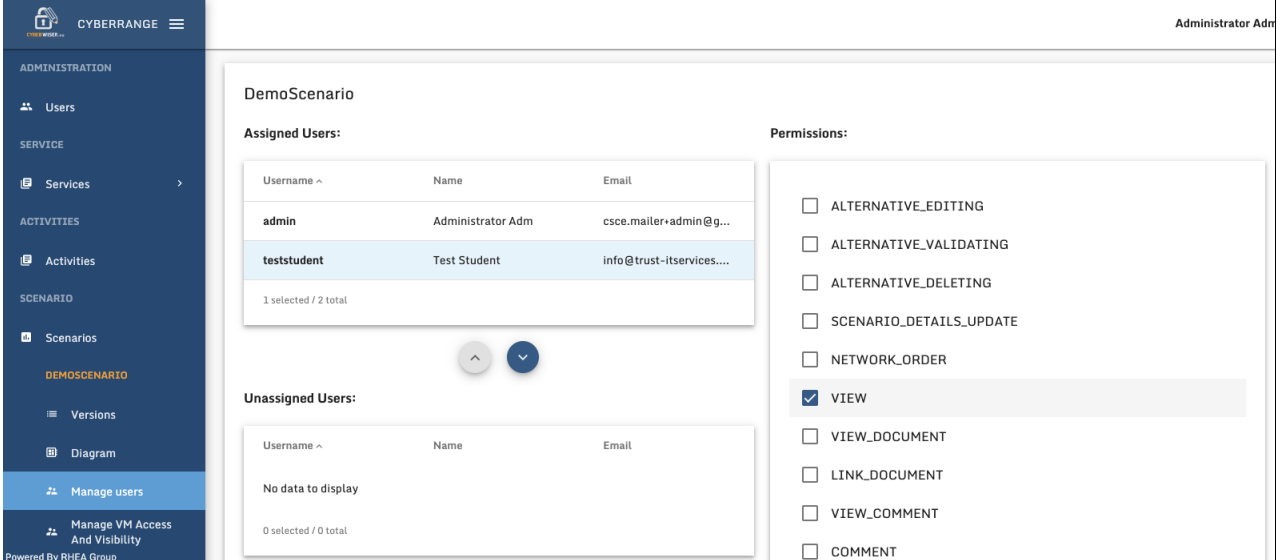
User Interface name	CYBERWISER.eu – Scenario user permissions
	

Table 19. UI.04.006 – CYBERWISER.eu – Scenario user permissions

User Interface name	CYBERWISER.eu – Add/edit asset
UI Id	UI.04.007
Description	<p>For an Asset, a name and description should be provided. An AssetType must be selected from the available options. Depending on the option selected in this select box, we will see different Asset Reference Types and Asset References to select in the next form fields.</p> <ul style="list-style-type: none"> <li>GATEWAY: to define of gateway (for example, a firewall or router)</li> <li>WORKSTATION: to define a desktop computer virtual machine</li> <li>SERVER: to define a server computer virtual machine</li> <li>GATEWAY_APPLIANCE: to define a physical gateway, that can be used in the scenarios.</li> <li>SERVER_APPLIANCE: to define a physical server, that can be used in the scenarios.</li> <li>NETWORK: to define a virtual network (this functionality is deprecated)</li> <li>EXTERNAL_NETWORK: to define a connected physical external to internal network, used for hybrid environments.</li> <li>CLIENT_APPLICATION: to define an application available for the workstations.</li> <li>SERVER_APPLICATION: to define an application available for the servers.</li> </ul> <p>This fields will be linked to the already classified Cloud assets in the IaaS provider, and it will make the system display the selected category.          In Asset Reference Type, depending on the selected Asset Type, different options will be presented in the select box:</p> <ul style="list-style-type: none"> <li>IMAGE: virtual machine image available in the CITEF Cloud (Open Nebula version).</li> <li>FILE: application available in the CITEF Cloud (Open Nebula version).</li> </ul>
Purpose	<p>The most important Assets in CITEF should be related to an AssetReference, pointing to a resource available in the IaaS. Assets like Workstation and Server for example, ideally already point to an available IMAGE, available in the IaaS, ready to</p>

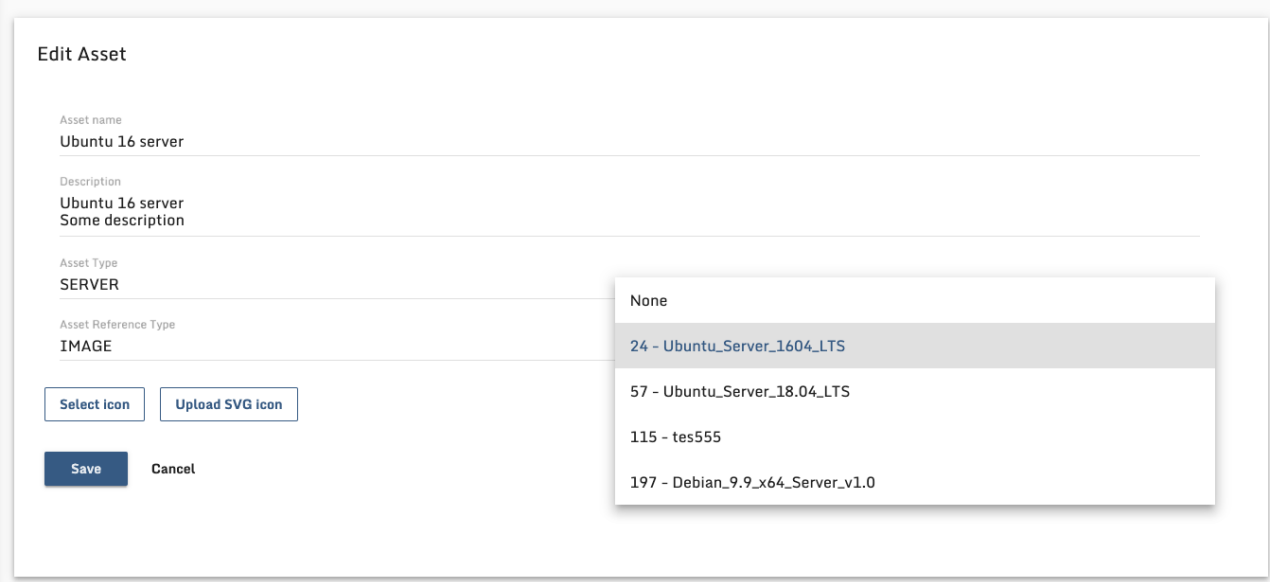
User Interface name	CYBERWISER.eu – Add/edit asset
	<p>be instantiated. AssetReferences are not strictly necessary though. It can be also useful to defined Assets for which you do not (yet) have an available IaaS AssetReferences. This way, users can already use these Assets in their Scenario Design, and you can indicate to those users, that Asset can actually be used in a scenario. Once added to a Scenario, to successfully instantiate the scenario would require the actual AssetReference to be filled in in the Digital Library.</p> <p>Before adding Assets to the Digital Library for which you want to specify an AssetReference, these AssetsReferences need to be already present in the CITEF Cloud, that can be managed by the Cloud Systems Engineers through the CITEF Infrastructure Service Portal, a cloud management interface, linked to the specific cloud provider.</p>
Navigation and user interaction	The user can navigate back to the asset listing.
Other comments	
	

Table 20. UI.04.007 – Add/edit asset

User Interface name	CYBERWISER.eu – VM access and visibility
UI Id	UI.04.008
Description	<p>VM access can be configured by selecting “VM Access” tab on “Manage VM Access and Visibility” page. If scenario template was not created a message “Scenario Environment Template is not created” will be.</p> <p>If scenario template is present, a list of available VMs instances will appear and access can be enabled for any VM instance from the list “Save” button at the bottom of the screen needs to be clicked for changes to be applied.</p> <p>If a user can see a node on a diagram but does not have an access enabled for any of node’s VMs, the VNC access icon will not be displayed and connection to node’s VMs will not be possible when scenario is instantiated. Note that a user can alternatively have FULL_VNC_ACCESS scenario permission assigned in order to get access to all scenario’s VMs.</p> <p>Node visibility can be granted before scenario template creation. Users who were assigned VIEW permission to the scenario gain visibility to all nodes of the scenario diagram automatically and thus do not require explicit visibility configuration to be set. On the contrary, users with FINALIZED_SCENARIO_VIEW permission need visibility to be enabled on a per-node basis. Node Visibility can be configured by selecting “Node Visibility” tab on “Manage VM Access and Visibility” page.</p>
Purpose	Each assigned to scenario user can be granted visibility to specific nodes of scenario diagram and given access to specific VMs instances of the selected scenario once scenario environment template is created. This is done by clicking “Manage VM Access And Visibility” menu item of corresponding Scenario Environment.
Navigation and user interaction	The user can switch back to the editor, version history or scenario listing using the left menu.
Other comments	

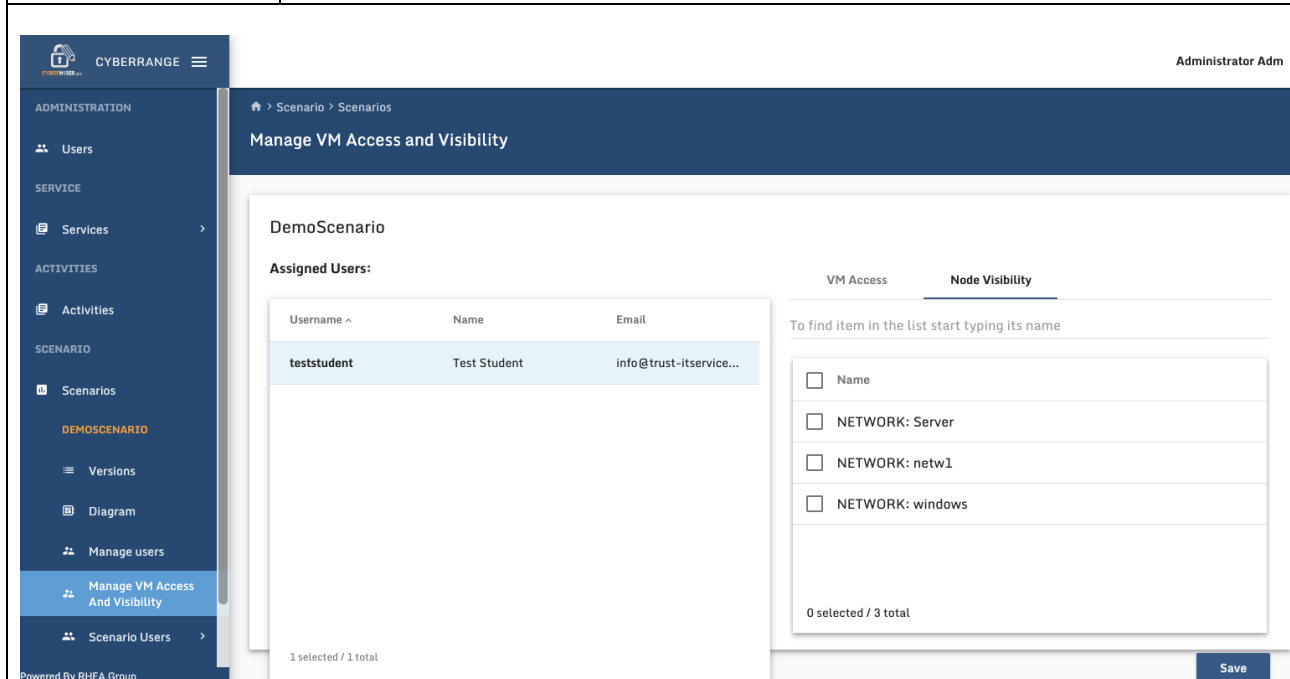


Table 21. UI.04.008 – CYBERWISER.eu – VM access and visibility

User Interface name	CYBERWISER.eu – VNC in modal
UI Id	UI.04.009
Description	<p>This screen shows the desktop/command line of the selected VM through a VNC connection. At the right top corner of the frame the following buttons are present:</p> <div> <div>Send CtrlAltDel</div> sends Ctrl+Alt+Del combination to VM </div> <div> <div>↓</div> copies text between user machine and VM (not available on all VMs) </div> <div> <div>⌂</div> opens VNC console in a full screen mode </div> <div> <div>↗</div> opens VNC console in a new browser tab </div> <p>When the VNC console is opened in a new browser tab all buttons described above will be present at the right top corner except the <div>↗</div> button.</p>
Purpose	
Navigation and user interaction	The user can close to pop up and go back to the editor view.
Other comments	

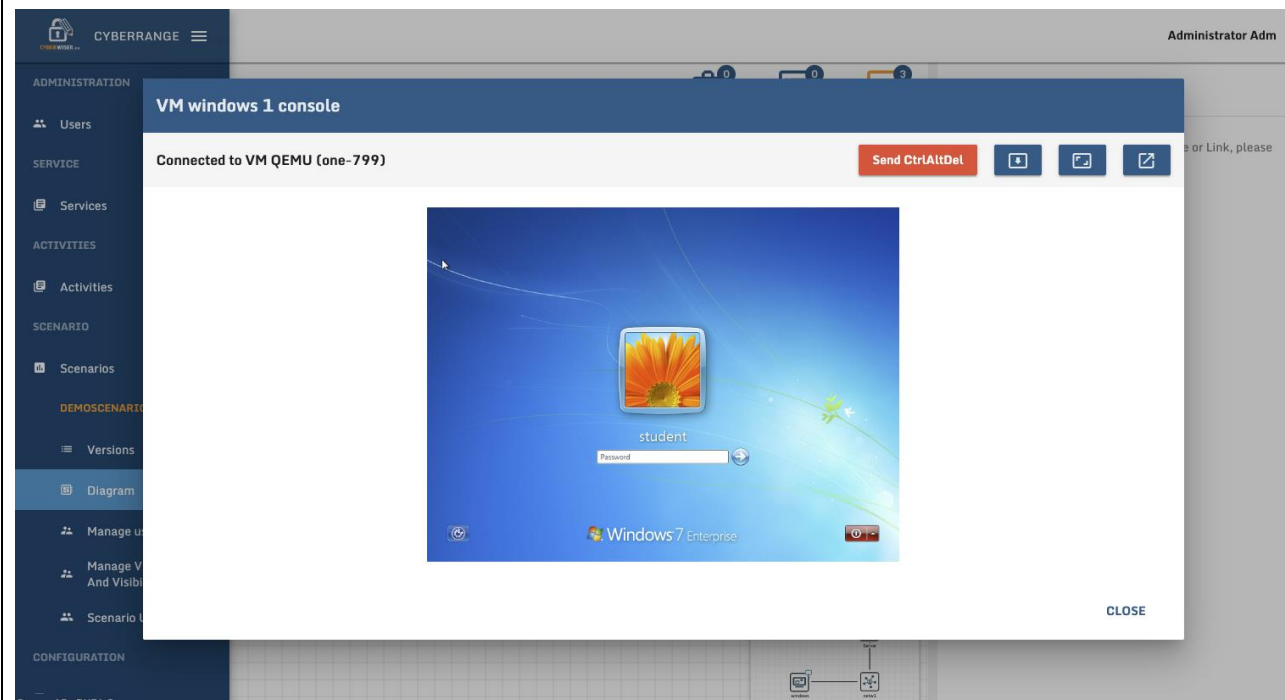


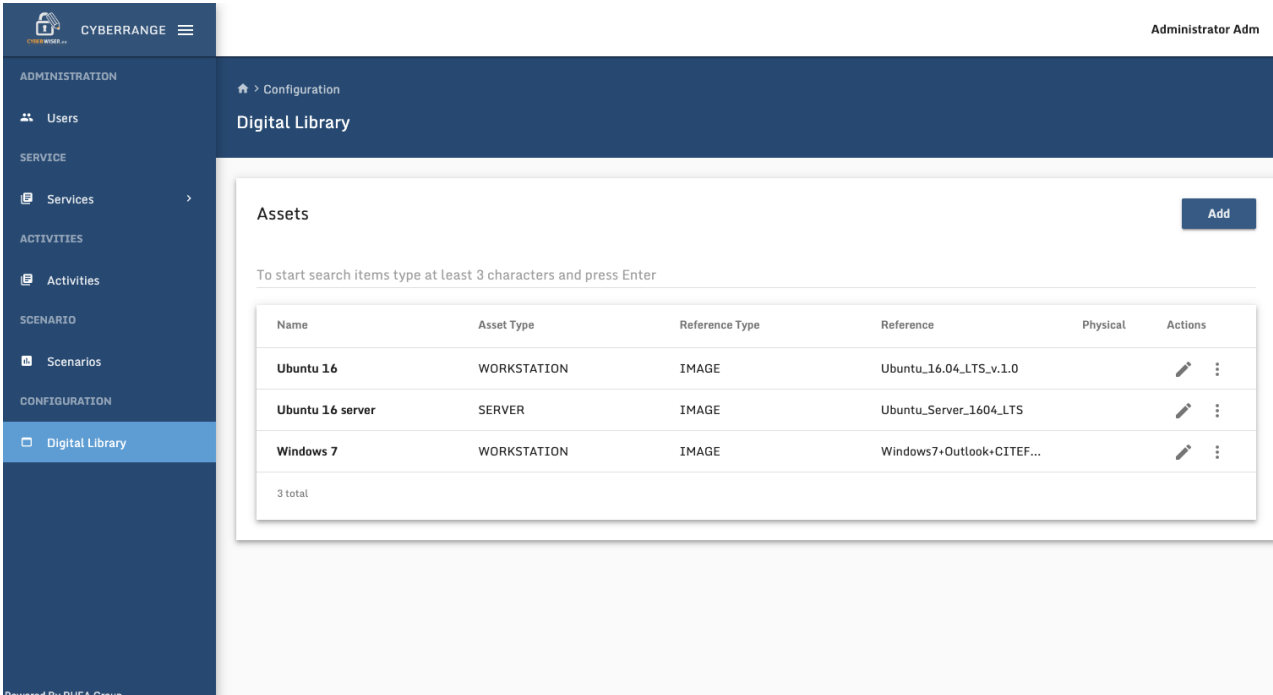
Table 22. UI.04.005 – CYBERWISER.eu – VNC in modal



User Interface name	Performance Evaluator
UI Id	UI.05.000
Description	
Purpose	
Navigation and user interaction	
Other comments	The PE will be a pure backend component with no user interface foreseen.

Table 23. UI.05.000 – Performance Evaluator

User Interface name	CYBERWISER.eu Digital Library
UI Id	UI.06.001
Description	The DL contains all the assets that we can link to nodes in the Scenario Designer. These can be either assets that will exist in the instantiated virtual environment (for example, virtual machines and software packages) but they can also be assets that are part of the diagram on the training or business layer, helping explain the context of a scenario. Examples of these are certain types of actors and events that may or should occur in a scenario.
Purpose	In order for these to be used in a Scenario Design, assets first need to be defined in the DL. The DL can be accessed from the menu, selecting "Digital Library" in the Configuration area.
Navigation and user interaction	From this page users can navigate to the asset details page to view and edit assets or create new ones.
Other comments	



ADMINISTRATION

Users

SERVICE

Services

ACTIVITIES

Activities

SCENARIO

Scenarios

CONFIGURATION

Digital Library

Configuration

Digital Library

Assets

Add

To start search items type at least 3 characters and press Enter

Name	Asset Type	Reference Type	Reference	Physical	Actions
Ubuntu 16	WORKSTATION	IMAGE	Ubuntu_16.04_LTS_v.1.0		
Ubuntu 16 server	SERVER	IMAGE	Ubuntu_Server_1604_LTS		
Windows 7	WORKSTATION	IMAGE	Windows7+Outlook+CITEF...		

3 total

Table 24. UI.06.001 – CYBERWISER.eu Digital Library

User Interface name	CYBERWISER.eu – Document Library
UI Id	UI.06.002
Description	The document library screen shows a listing of documents that the current user can access. It has a drag-and-drop area at the bottom of the screen to allow the user to upload new documents. Document can be (de)-selected from the list, allowing the user to manage which document are linked to a data item.
Purpose	The document library is a popup screen that is used to select or upload documents in order to attach them to data item in the cyber range. This can for example be a specific node in a scenario design.
Navigation and user interaction	The user can close the window, updating the current document selection.
Other comments	

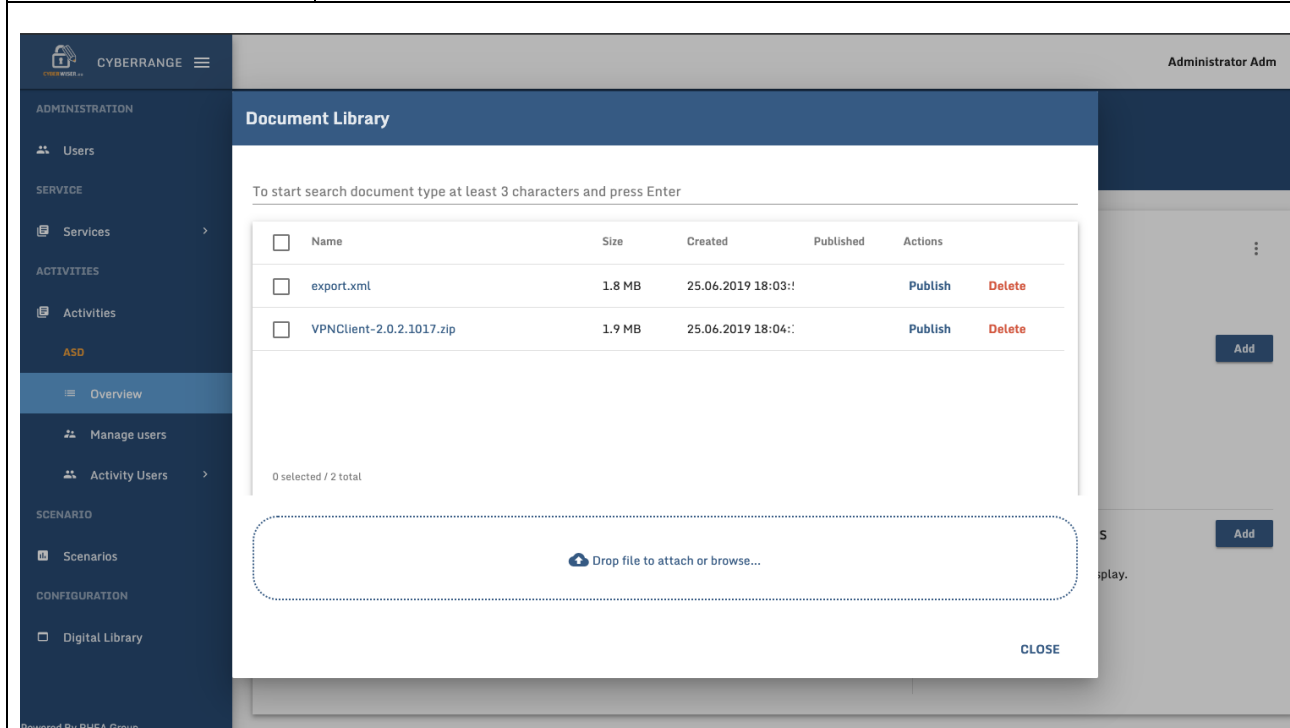


Table 25. UI.06.002 – CYBERWISER.eu Document Library

User Interface name	Economic Risk Evaluator – Quantitative risk report
UI Id	UI.07.001
Description	This interface shows the output of the risk assessment performed by the ERE. The results are expressed in a quantitative way, in terms of money. There are three key parameters to express the monetary exposure of the company. The typical loss to which they are exposed, the loss in a worst-case scenario, and an estimation of the number of events that may suffer the company during a year. The overall cyber risk is broken down into individual risks, and in turn the risks are detailed per target. The same three parameters are reported.
Purpose	The purpose of this interface is to inform about the economic exposure of the company to cyber risks, considering the existing cyber climate in which it operates.
Navigation and user interaction	For each section, individual risk or individual target, the parameters presented above are informed. There is a combo box to select which one to visualize. There are three tabs entitled <i>Quantitative</i> , <i>Mitigations</i> and <i>Info</i> that show different aspects of the risk report. The user can move from one to another freely. Similarly, by using the top menus some other parts of the ERE can be visited.
Other comments	

Risk Status Quantitative

Overall risk status:

Typical Loss: 32,959.49 EUR
Worst Case: 94,609.36 EUR
Events: 1.568

<b>Section:</b>	WRP1: Denial of Service Attack	Typical Loss: 157.33 EUR ▾
WRP1-R1	Hacker causes Service/s not available harms asset Availability of service	Typical Loss: 157.33 EUR ▾
<b>Section:</b>	WRP2: Invalidated Redirects and Forwards	Typical Loss: 32,802.16 EUR ▾
WRP2-R1	Hacker redirects victim to malicious site harms asset Integrity of system	Typical Loss: 6,560.06 EUR ▾
WRP2-R2	Hacker redirects victim to malicious site harms asset Confidentiality of user data	Typical Loss: 26,242.10 EUR ▾
<b>Section:</b>	WRP3: Bypass Login	Typical Loss: ▾

Table 26. UI.07.001 – Economic Risk Evaluator – Quantitative risk report

User Interface name	Economic Risk Evaluator – Mitigation measures
UI Id	UI.07.002
Description	This interface shows the mitigation measures proposed by the ERE in accordance to the cyber risk status computed following the cyber climate in which the organization operates. For each mitigation, some details are provided.
Purpose	The purpose of this interface is to show a list of proposals for countermeasures with the purpose of diminishing the cyber risk exposure of the organization.
Navigation and user interaction	By hovering over each mitigation measure, some additional information is displayed to provide the user with more details about such measure, so as to eventually implement it. There are three tabs entitled <i>Quantitative</i> , <i>Mitigations</i> and <i>Info</i> that show different aspects of the risk report. The user can move from one to another freely. Similarly, by using the top menus some other parts of the ERE can be visited.
Other comments	

Mitigation Measures

**Section:** WRP1: Denial of Service Attack

**Section:** WRP2: Invalidated Redirects and Forwards

Monitoring Engine 212.34.151.210

M8 Validate input

M9 Use an intermediate disclaimer page

M10 Map input values to actual filenames/URLs etc. and reject all other input

M11 Require unique nonce for all redirect request

M12 Identify all the potential areas where untrusted inputs can enter your software

M13 Use an application firewall that can detect attacks against URL redirection

Test Machine 192.168.3.65


**Section:** WRP3: Bypass Login


Table 27. UI.07.002 – Economic Risk Evaluator – Mitigation measures

User Interface name	Economic Risk Evaluator – Risk reports info						
UI Id	UI.07.003						
Description	This interface shows the history of issued cyber risk reports. For each report at first sight we see the date of creation, the quantitative result of the evaluation in the shape of a combo in which the user can see the exposure in typical and worst case, and the estimation of incidents per year						
Purpose	The purpose is to show the evolution of the cyber risk exposure over the time and to give some hints about trends that may trigger strategic actions at corporate level to diminish the risk.						
Navigation and user interaction	There are three tabs entitled <i>Quantitative</i> , <i>Mitigations</i> and <i>Info</i> that show different aspects of the risk report. The user can move from one to another freely. Similarly, by using the top menus some other parts of the ERE can be visited.						
Other comments							
<table border="1"> <thead> <tr> <th>Date</th><th>Quantitative</th></tr> </thead> <tbody> <tr> <td>Aug. 5, 2019, 1:13 p.m.</td><td>Typical Loss: 15,100.54 EUR ▼</td></tr> <tr> <td>Aug. 5, 2019, 1:06 p.m.</td><td>Typical Loss: 15,040.14 EUR ▼</td></tr> </tbody> </table>		Date	Quantitative	Aug. 5, 2019, 1:13 p.m.	Typical Loss: 15,100.54 EUR ▼	Aug. 5, 2019, 1:06 p.m.	Typical Loss: 15,040.14 EUR ▼
Date	Quantitative						
Aug. 5, 2019, 1:13 p.m.	Typical Loss: 15,100.54 EUR ▼						
Aug. 5, 2019, 1:06 p.m.	Typical Loss: 15,040.14 EUR ▼						

Table 28. UI.07.003 – Economic Risk Evaluator – Risk reports info

User Interface name	Economic Risk Evaluator – Targets configuration
UI Id	UI.07.004
Description	<p>This interface is used to check the training scenario information about all the monitored assets, called <i>targets</i> in this context that are considered in the calculation of the cyber risk exposure. The ERE does not calculate the cyber risk exposure directly for the whole infrastructure. Instead, it calculates it for each target individually and then aggregates the results. For each target introduced in the analysis the following information is available:</p> <ul style="list-style-type: none"> <li>• Target ID: This is an internal number for the ERE database</li> <li>• IP addresses of the machine</li> <li>• When the target is not a machine as a whole, but a specific application, the port is specified as well</li> <li>• Textual short description of the target</li> <li>• Textual complete description of the target.</li> <li>• Importance of the availability<sup>61</sup> of the information contained in the target, measured in a scale from 0 to 10.</li> <li>• Importance of the confidentiality<sup>62</sup> of the information contained in the target, measured in a scale from 0 to 10.</li> <li>• Importance of the integrity<sup>63</sup> of the information contained in the target measured in a scale from 0 to 10.</li> </ul>
Purpose	The purpose of the interface is to show the current target configuration, in case some need for changes is detected by the user.
Navigation and user interaction	<p>For each target there two icons, one linking to the interface to edit the information about the target, and a trash icon to delete the target. There is also a disabled link for the creation of a new target.</p> <p>The user can move to other parts of the ERE by means of the top menus</p>
Other comments	

Targets 

 Add Target







ID	IP Address	Port	Short Description	Complete Description	Availability	Confidentiality	Integrity		
71	212.34.151.210		Monitoring Engine	Monitoring Engine VM	10	10	8		
74	0.0.0.0		Atos	Infrastructure as a whole	6	9	7		
82	192.168.3.65		Test Machine	Test Machine	10	10	10		

Table 29. UI.07.004 – Economic Risk Evaluator – Targets configuration

<sup>61</sup> Availability means that, when required, the data can be accessed

<sup>62</sup> Confidentiality is about preventing sensitive information from reaching the wrong people, while ensuring that the right people can in fact get it.

<sup>63</sup> Data integrity assures that data are changed only in a specified and authorized manner

User Interface name	Economic Risk Evaluator – Target detail
UI Id	UI.07.005
Description	This interface is used to view a detailed information associated to a specific target.
Purpose	Check the economic values of the fields describing a target of cyber risk analysis
Navigation and user interaction	The information is provided by means of a form with the different fields. The Submit button is disabled since the target information is provided during the definition of the training scenario. The user can move to other parts of the ERE by means of the top menus.
Other comments	

Targets: Edition

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.

IP address:

Port:

Short description:

Complete description:

Availability:

Confidentiality:

Integrity:

It is strongly recommended that you specify the potential loss that could result from breaches of confidentiality, integrity or availability linked to this machine / application. If you do not provide such values, default estimated values will be applied.

Please indicate the loss values for a typical loss scenario and for the worst scenario in euros per incident

Loss typical availability:

Loss typical confidentiality:

Loss typical integrity:

Loss worst availability:


Loss worst confidentiality:


Loss worst integrity:


Table 30. UI.07.005 – Economic Risk Evaluator – Target detail



User Interface name	Economic Risk Evaluator – Models selected
UI Id	UI.07.006
Description	It is a simple interface showing the model/s that are active now. This is, the models whose algorithms will be executed to calculate cyber risk exposure. To make possible an assessment at least a model must be selected. The scenario designer can select as many models as needed. The more models selected, the more risk scenarios considered in the analysis.
Purpose	Show the models that are currently selected and therefore active for the calculation of the cyber risk exposure.
Navigation and user interaction	There is a button with a trash icon to deselect risk models and another that links to the interface of risk model selection which are disabled. The user can move to other parts of the ERE by means of the top menus.
Other comments	

Risk Model Selection 

 Select risk models

 Clear risk models

ID	Name	Risk Model name
1	Atos	WRP2

Table 31. UI.07.006 – Economic Risk Evaluator – Models selected

User Interface name	CYBERWISER.eu – Vulnerability Assessment Tools, Login page
UI Id	UI.08.001
Description	The page presents a form for submission of user credentials (username, password fields and a login button).
Purpose	Authentication of users to the Vulnerability Assessment Tools web application.
Navigation and user interaction	After successful login, the user is redirected to the main page showing the listing of vulnerability scanning tasks.
Other comments	The UI is accessible via Web browser running on the user's workstation VM, but only when the VAT is a part of the training scenario.

Table 32. UI.08.001 – CYBERWISER.eu – Vulnerability Assessment Tools, Login page

User Interface name	CYBERWISER.eu – Vulnerability Assessment Tools, Main page
UI Id	UI.08.002
Description	<p>This page shows the listing of vulnerability scans based on the role of the authenticated user in the platform (trainer/trainee). If the user is a trainee, only the data related to vulnerability scans requested by him is displayed (initially an empty set). If the user is a trainer, all the vulnerability scanning tasks are shown: those requested by the CYBERWISER.eu platform at the time of the scenario design, those requested by him or any other trainer, and those requested by any trainee participating in the exercise.</p> <p>The following data is shown for every vulnerability scanning task: name, type (generic scan / custom vulnerability detection), IP address of the target of the scan, status of the scanning task (created / pending / in progress / done / cancelled / aborted), identifier of the entity (user / platform) who requested the scan, timestamps associated with submission and possibly modification of the scan, and actions for manipulation of the scan (if any).</p>
Purpose	Providing a general overview of vulnerability scanning tasks in the training scenario, allowing monitoring and management of vulnerability scans.
Navigation and user interaction	<p>For every scanning task displayed in the listing, there is:</p> <ol style="list-style-type: none"> <li>1.) a set of buttons/links to pages for triggering actions associated with the scanning task, depending on the task's current status. If the task was created by the user but not submitted, there is a <i>submit</i> button that starts the scan. If the task has not concluded yet, it is possible to cancel the scan by clicking a button. Clicking a button for modification redirects the user to the vulnerability scan modification page.</li> <li>2.) A link to the details page.</li> </ol> <p>On the main page there is also a link to the page for creation of a new vulnerability scanning task.</p>
Other comments	The UI is accessible to authenticated and authorized users of the platform via the Web browser running on the user's workstation VM, but only when the VAT is a part of the training scenario.

Table 33. UI.08.002 – CYBERWISER.eu – Vulnerability Assessment Tools, Main page

User Interface name	CYBERWISER.eu – Vulnerability Assessment Tools, Vulnerability scan details page
UI Id	UI.08.003
Description	<p>This page shows the details of the selected vulnerability scanning task. Beside all the data already shown in the main page, all the configuration pertaining the vulnerability scan is also shown. Specifically, the values of vulnerability scan timeout and its schedule are shown, and an indication of the scanner that was used, along with the values of all relevant properties. In case the scan is associated with a vulnerability detection script (i.e. not a generic scan), the associated detection script is also available on this page.</p> <p>If the scanning task was executed at least once, the listing of executions is also displayed. For every execution of the scan, the timestamps indicating when the execution was started and finished are shown, along with the result (vulnerability reports for generic scans and an indication of whether a specific vulnerability is present or not for vulnerability detection scripts), the output files generated by the vulnerability scanner/script, and the standard output of the process in which the scan/script was run.</p>
Purpose	Providing a detailed view of the selected vulnerability scanning task and corresponding executions, grouping all the data and actions associated with the particular scan in one place.
Navigation and user interaction	<p>In case a vulnerability scanning task is associated with a vulnerability detection script (i.e. not a generic scan), the page contains download link for the script used to perform the detection.</p> <p>Just like in the main page with the vulnerability scan listing, a set of actions for manipulation of the scanning task (depending on its current status) is available via buttons (submission, cancellation, modification). Modification button redirects to the page for modification of an existing vulnerability scan.</p> <p>If the scanning task was executed at least once, download links for the output file generated by the scanner/detection script (if any), as well as standard output of the process in which the scanner/script was run, are available for every execution.</p>
Other comments	The UI is accessible to authenticated and authorized users of the platform via the Web browser running on the user's workstation VM, but only when the VAT is a part of the training scenario.

Table 34. UI.08.003 – CYBERWISER.eu – Vulnerability Assessment Tools, Vulnerability Scan details page

User Interface name	CYBERWISER.eu – Vulnerability Assessment Tools, Creating a new vulnerability scan
UI Id	UI.08.004
Description	This page allows the user to configure all aspects of a vulnerability scanning task.
Purpose	Configuration and subsequent creation of a new vulnerability scanning task.
Navigation and user interaction	<p>The user is guided through the process of configuring a vulnerability scanning task by means of a wizard comprising the following steps:</p> <ol style="list-style-type: none"> <li>1.) <i>Choice of the type of vulnerability scan.</i> Two options are available: generic scan and detection with a script. Note that this step is applicable for trainers only, as the trainees will only be offered generic scanners (i.e. they will skip directly to step 2a, see below).</li> <li>2.) <i>Configuration of the vulnerability scanner.</i> This step depends on the type of the vulnerability scan from the previous step:               <ol style="list-style-type: none"> <li>a. <u>For the generic scanners</u>, the user selects one or more scanning modules (for instance OpenVAS, w3af, ...) and configures properties common to all the scanning modules (for instance IP address of the target to scan) as well as any supported module-specific properties;</li> <li>b. <u>For detection of vulnerabilities with scripts</u>, the process is exactly the same as for creation of new attack tasks (see steps 1 and 2 in UI.11.004). The only difference is that here we are referring to vulnerability detection scripts instead of attack scripts.</li> </ol> </li> <li>3.) <i>(optional) Configuration of custom scan timeout and schedule</i> via numeric input fields and checkboxes. It will be possible to customize total number of executions, the interval between consecutive executions, and the delay before initial execution of a scan.</li> </ol> <p>Once the above sequence of steps is complete, a new scanning task is created (but not yet submitted), and the user is redirected to the details page of the newly created task, from where he can submit it.</p>
Other comments	The UI is accessible to authenticated and authorized users of the platform via the Web browser running on the user's workstation VM, but only when the VAT is a part of the training scenario.

Table 35. UI.08.004 – CYBERWISER.eu – Vulnerability Assessment Tools, creating a new vulnerability scan

User Interface name	CYBERWISER.eu – Vulnerability Assessment Tools, Modifying a vulnerability scan
UI Id	UI.08.005
Description	This page allows the user to modify an existing scanning task. Specifically, it is possible to modify the timeout value of a scan, total number of scan executions and the interval between consecutive executions.
Purpose	Support for modification (reconfiguration) of an existing vulnerability scanning task.
Navigation and user interaction	This page will be presented either as a subpage on the vulnerability scan details page or as a modal. The user will be able to modify configurable aspects of the scanning task via numeric input fields and checkboxes. Clicking a button for modification will result in immediate updating of the scanning task, after which the user will be redirected to the page with task details.
Other comments	The UI is accessible to authenticated and authorized users of the platform via the Web browser running on the user's workstation VM, but only when the VAT is a part of the training scenario.

Table 36. UI.08.005 – CYBERWISER.eu – Vulnerability Assessment Tools, Modifying a vulnerability scan

User Interface name	Monitoring Sensors
UI Id	UI.09.000
Description	
Purpose	
Navigation and user interaction	
Other comments	The Monitoring Sensors do not offer specific user interface

Table 37. UI.09.000 – Monitoring Sensors

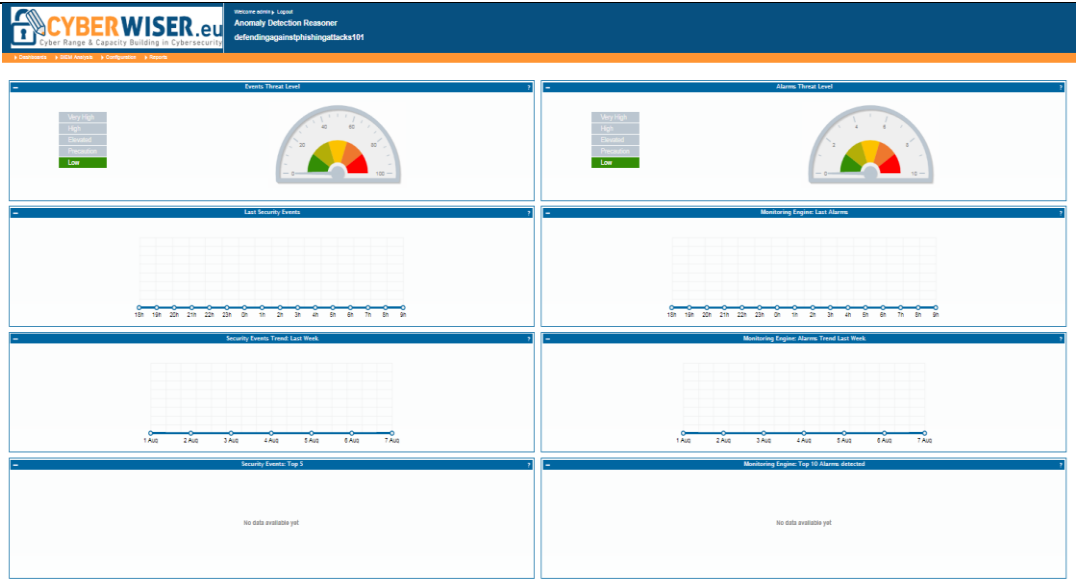
User Interface name	Anomaly Detection Reasoner landing page
UI Id	UI.10.001
Description	<p>This interface is a dashboard that shows an overview of the status of the monitored infrastructure aiming at providing a high-level vision, understandable both by technical and managerial profiles.</p> <p>There is a widget for the event threat level is represented by means of a speedometer whose needle marks such level.</p> <p>Another widget uses the same approach to inform about the alarms threat level.</p> <p>There is a widget showing a timeline with a curve with periodic (hourly) measures of the latest security events. Each measure indicates how many events took place since the previous measurement. This way the user can see the ups and downs in different moments of the day and try to identify trends.</p> <p>Another widget does the same to show the number of alarms</p> <p>Aggregating the hourly samples we can obtain the daily measurements for the security events and represent the last week. This is done in a different widget.</p> <p>The same is done with monitoring engine alarms, showing the evolution in the number of alarms.</p> <p>There is a widget showing the top 5 security events measured during the last week, indicating the number of occurrences of each event.</p> <p>Finally, another widget shows the top 10 alarms detected during the last week.</p> <p>On the top part, some links allow to move to other views of the ADR.</p>
Purpose	This interface offers a summary of the most relevant information that permits to obtain a quick picture of the status of the monitored infrastructure, offering hints on where the problems might be, triggering the consultation of more detailed views. This interface is understandable by a wide range of profiles within the company.
Navigation and user interaction	Most widgets show evolution curves that allow to identify trends. By hovering the mouse pointer over the points of such curves, a pop-up appears showing further information. From this interface the user can go to the specific interfaces in which the information about events and alarms respectively is elaborated with a higher level of detail
Other comments	
	

Table 38. UI.10.001 – Anomaly Detection Reasoner landing page

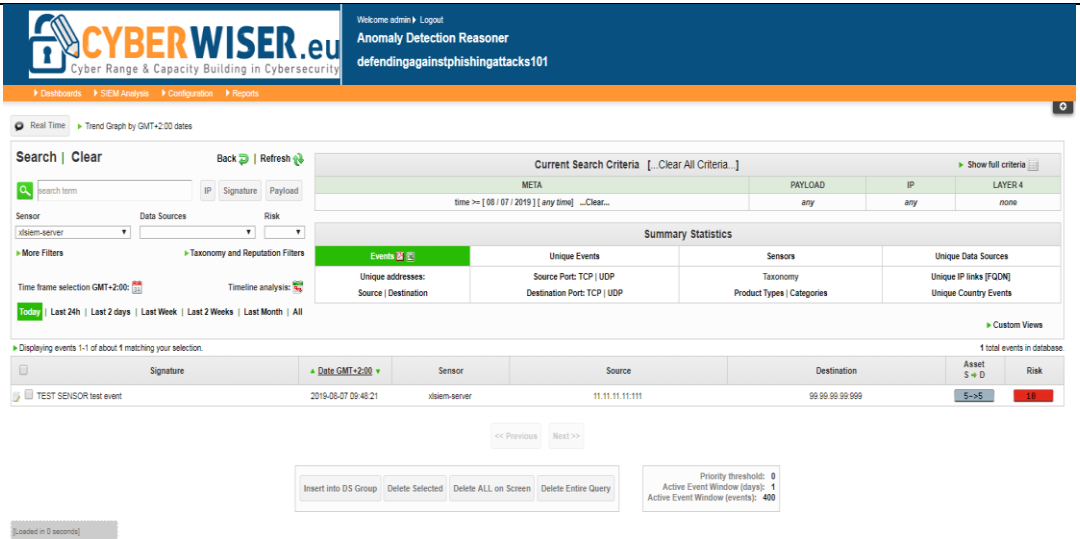
User Interface name	Events overview
UI Id	UI.10.002
Description	This interface shows the latest 50 events produced by the ADR, offering the chance to see older ones thanks to its paging feature. It has two parts: one is a widget in which different filtering parameters and search criteria can be introduced. The other part is the actual list of events raised. For each row the main information related to a certain event is shown: the event name, date and time, the name of the sensor that is behind such event, the source IP and the destination IP
Purpose	The purpose of this interface is to provide a quick list of the most recent events raised by the sensors deployed in the infrastructure, with the chance to look into older ones thanks to the paging feature. Also, to focus on determined types of events thanks to the filtering features.
Navigation and user interaction	By clicking on a specific event, the user can go to another interface in which the details are shown. By filling up the different filtering and search criteria the user can obtain the information that is really relevant for him. The introduced criteria can be cleaned at any moment. The user can go to other interfaces of the ADR going to the top menus and choosing the corresponding links.
Other comments	
	 <p>The screenshot displays the 'Events overview' interface of the CYBERWISER.eu platform. At the top, there is a navigation bar with the logo and menu items like Dashboards, SIEM Analysis, Configuration, and Reports. Below this, a search bar and various filters (Sensor, Data Sources, Risk) are visible. A 'Current Search Criteria' section shows filters for META, PAYLOAD, IP, and LAYER 4. A 'Summary Statistics' table provides an overview of unique events, sensors, and data sources. The main part of the interface is a table of events, with one event highlighted: 'TEST SENSOR test event' from 'xsiem-server' on '2019-08-07 09:48:21'. At the bottom, there are controls for pagination and a status bar showing 'Loaded in 0 seconds'.</p>

Table 39. UI.10.002 – Event overview



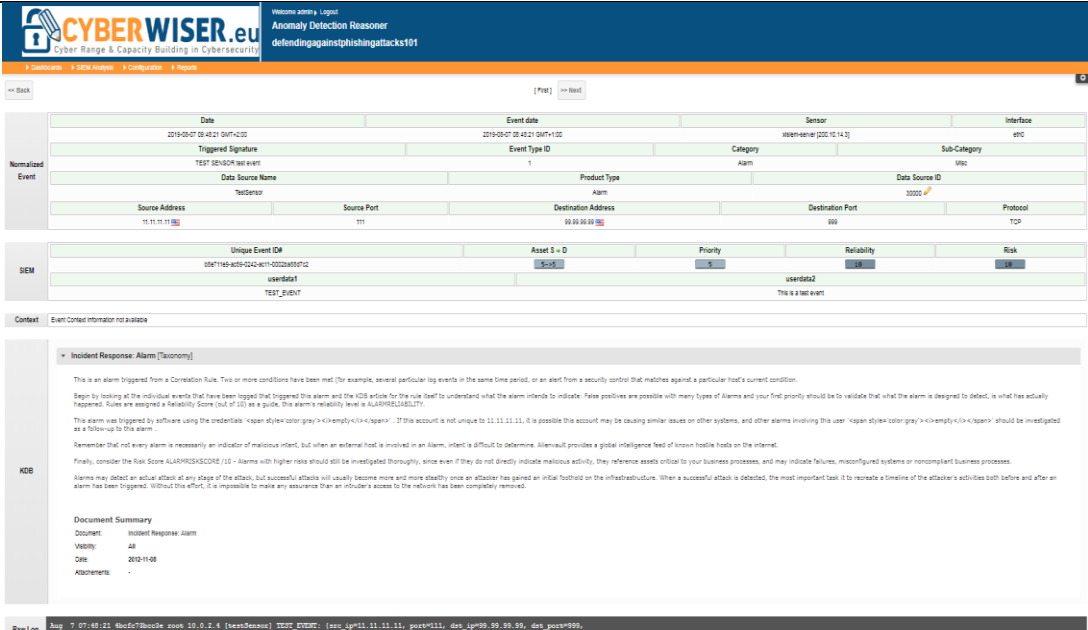
User Interface name	Event detail
UI Id	UI.10.003
Description	<p>This interface offers detailed information about a specific event selected by the user in the interface <i>Events Overview</i>.</p> <p>There is a set of normalized data shown about the event. The most relevant information is the following:</p> <ul style="list-style-type: none"> <li>• Event date</li> <li>• Sensor and corresponding IP that has sent logs to produce the event</li> <li>• Associated network interface (for instance, eth0)</li> <li>• Triggered signature: this is a short descriptive name of the event</li> <li>• Event category (Network, Host...)</li> <li>• Data Source Name: it is the type of sensor that has sent logs</li> <li>• Source IP and port</li> <li>• Destination IP and port</li> <li>• Protocol</li> </ul> <p>Depending on the event and the information available, supplementary information can be provided to the user to know more about what is being reported</p>
Purpose	This interface offers as much detail as possible about a specific event raised in the ADR. The information is structured in a schematic way that allows the user to directly look for what is relevant to the organization with respect to this event.
Navigation and user interaction	<p>The interface offers buttons entitled <i>previous</i> and <i>next</i> to move forward and backwards along the different events without needing to go back to the <i>Event Overview</i> interface.</p> <p>For specific data of the list shown above, there is the chance to click o obtain further details, for example those related to the involved IPs.</p>
Other comments	
	 <p>The screenshot displays the 'Event detail' interface within the CYBERWISER.eu platform. At the top, there's a navigation bar with tabs for 'Dashboard', 'Event Analysis', 'Configuration', and 'Reports'. The main content area is divided into several sections:</p> <ul style="list-style-type: none"> <li><b>Normalized Event:</b> A table showing event details such as Date (2019-03-07 03:42:21 GMT+020), Event date (2019-03-07 03:42:21 GMT+020), Sensor (sensorsense [202.10.14.1]), Interface (eth0), Triggered Signature (TEST_SENSOR test event), Event Type ID (1), Category (Alarm), Sub-Category (WNC), Data Source Name (testSensor), Product Type (Alarm), Data Source ID (30000 #), Source Address (15.15.15.15), Source Port (80), Destination Address (99.99.99.99), Destination Port (80), and Protocol (TCP).</li> <li><b>SEM (Security Event Manager):</b> A section showing Unique Event ID (108116f-act0-0242-4011-002b0501c), Asset S = D (S-C-S), Priority (S), Reliability (S), and Risk (S).</li> <li><b>Context:</b> A section titled 'Incident Response: Alarm [Taxonomy]' providing detailed information about the alarm, including a document summary, document name (Incident Response: Alarm), validity (All), date (2019-10-08), and attachments.</li> <li><b>Raw Log:</b> A section at the bottom showing the raw log data for the event.</li> </ul>

Table 40. UI.10.003 – Event detail

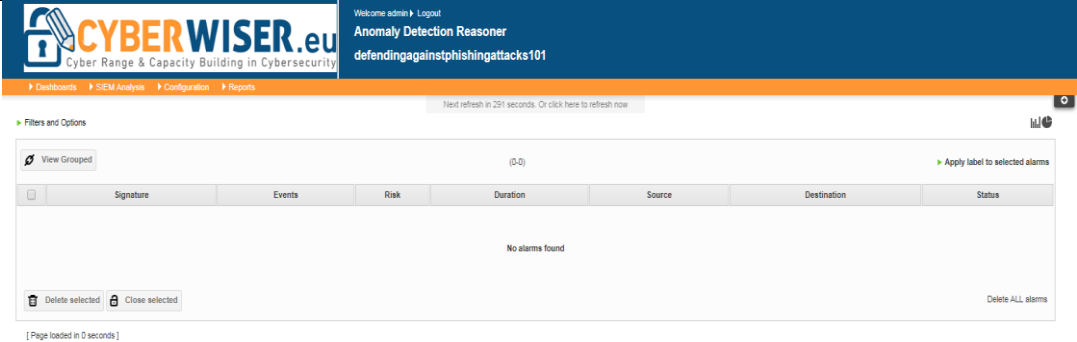
User Interface name	Alarms overview
UI Id	UI.10.004
Description	<p>This interface offers an overview of the alarms launched by the ADR after applying correlation rules to the different events received.</p> <p>The alarms are shown in a table, starting from the latest ones and including a paging feature to see older ones. Each row offers information about an alarm, namely a descriptive name of such alarm, the number of events that were correlated to obtain such alarm, the risk associated in a scale from 1 to 10, the duration of the alarm, the source IP, the destination IP and the status of the alarm.</p>
Purpose	This interface provides a quick overview of the alarms raised by the ADR, just showing the very basic information.
Navigation and user interaction	<p>The alarms can be shown ungrouped and grouped. The latter is very useful in order not to overload the table with repetitions of the same alarm in different moments. This way the user can see more types of alarms and get a better picture of the variety of incidents taking place. When the alarms are shown grouped, for each group the user has the chance to take responsibility for their treatment clicking on the field <i>Owner</i>, provided that he has the needed permissions. Similarly, if the status of an alarm is <i>Open</i>, if the user has the corresponding permissions can eventually evolve the status to <i>Closed</i>.</p> <p>The interface has a paging feature to see older bunches of alarms.</p> <p>By clicking on a specific row, the user can be taken to the detail of the alarm, which will be explained in the following. By clicking on the source or destination IP the user can see filtered the alarms involving the IP in question.</p> <p>Each row has a tick box that may be checked by the user, below there are two buttons for actions to be applied to the ticked rows: one is for deleting the chosen rows and the other one is for closing the selected alarms, this way, when a bunch of alarms is to be closed, the closing can be done in a more agile way.</p>
Other comments	
	

Table 41. UI.10.004 – Alarms overview

User Interface name	Alarm detail
UI Id	UI.10.005
Description	<p>This interface offers in an intuitive way detailed information about a chosen alarm. It indicates:</p> <ul style="list-style-type: none"> <li>the name of the alarm (this name will be descriptive enough),</li> <li>the status of the alarm (open / closed),</li> <li>how many events were correlated to obtain the alarm,</li> <li>the associated risk level derived from such correlation,</li> <li>how long went by between the first and the last event associated to the alarm,</li> <li>how long it went by since the alarm was created,</li> <li>specific information about the source IP address such as the IP itself and related port (if any and if this information is available), the location of the IP<sup>64</sup> (if known) and whether there is OTX<sup>65</sup> information available.,</li> <li>the same information about the destination IP.</li> <li>An enlightening text from the knowledge providing useful details about the alarm and linking to external sources of information, if they were available.</li> <li>Finally, there is an <i>Event Detail</i> tab which is used to see the list of events that were correlated. There might be several levels of correlation, that can lead to more precise and relevant alarms, this is clearly expressed in this tab.</li> </ul>
Purpose	Provide the user with as much information as possible about a specific alarm
Navigation and user interaction	The main action that can be taken is to go to the detail about the events that were correlated to generate the alarm. There two buttons at the bottom, one to open a ticket related to an alarm and the other one to close and archive the alarm. Hovering the mouse over relevant fields some pop-up appears offering additional information. Finally, there is a button to go back to the list of alarms that is shown in the interface named <i>Alarms Overview</i> .
Other comments	

Table 42. UI.10.005 – Alarm detail

User Interface name	CYBERWISER.eu – Attack Simulator, Login page
UI Id	UI.11.001
Description	The page will present a form for submission of user credentials (username, password fields and a login button).
Purpose	Authentication of users to the Attack Simulator web application.
Navigation and user interaction	After successful login, the user is redirected to the main page showing the listing of attacks.
Other comments	The UI is accessible via Web browser running on the user's workstation VM, but only when the AS is a part of the training scenario.

Table 43. UI.11.001 – CYBERWISER.eu – Attack Simulator, Login page

<sup>64</sup> If the location is available, it can be seen in a embedded map to get a better idea of the geographic position of that machine

<sup>65</sup> OTX stands for Open Threat Exchange

User Interface name	CYBERWISER.eu – Attack Simulator, Main page
UI Id	UI.11.002
Description	<p>This page shows the listing of attacks based on the role of the authenticated user in the platform (trainer/trainee). If the user is a trainee, only the attacks requested by him are shown (initially an empty set). If the user is a trainer, all the attacks are shown: those requested by the CYBERWISER.eu platform at the time of the scenario design, those requested by him or any other trainer, and those requested by any trainee participating in the exercise.</p> <p>The following data is shown for every attack task: name, status (created / pending / in progress / done / cancelled / aborted), identifier of the entity (user / platform) who requested the task, timestamps associated with submission and possibly modification of the attack, and actions for manipulation of the attack (if any).</p>
Purpose	Providing a general overview of attack tasks in the platform, allowing monitoring and management of attacks.
Navigation and user interaction	<p>For every attack task displayed in the listing, there is:</p> <ul style="list-style-type: none"> <li>• a set of buttons/links to pages for triggering actions associated with the attack task, depending on the task's current status. If the task was created by the user but not submitted, there is a <i>submit</i> button that activates the attack. If the task has not concluded yet, it is possible to cancel the attack by clicking a button. Clicking a button for modification redirects the user to the attack modification page.</li> <li>• A link to the details page.</li> </ul> <p>On the main page there is also a link to the page for creation of a new attack task.</p>
Other comments	The UI is accessible to authenticated and authorized users of the platform via the Web browser running on the user's workstation VM, but only when the AS is a part of the training scenario.

Table 44. UI.11.002 – Attack Simulator, Main page

User Interface name	CYBERWISER.eu – Attack Simulator, Attack details
UI Id	UI.11.003
Description	<p>This page shows the details of the selected attack task. Beside all the data already shown for the attack task in the main page, all the configuration pertaining the attack is also shown. Specifically, the values of attack timeout and its schedule are shown, and an indication of the attack tool used to execute the attack. The associated attack script will also be available via this page.</p> <p>If the attack task was executed at least once, the listing of executions is also displayed. For every execution of the attack, the timestamps indicating when the execution was started and finished are shown, along with an interpretation of the result (for instance success / failure), the output files generated by the attack script, and the standard output of the process in which the attack script was run.</p>
Purpose	Providing a detailed view of the selected attack task and corresponding attack executions, grouping all the data and actions associated with the attack in one place.
Navigation and user interaction	<p>The page contains download links for the attack script and potentially other files referenced in the attack script.</p> <p>Just like in the main page with the attack listing, a set of actions for manipulation of the attack task (depending on its current status) is available via buttons (submission, cancellation, modification). Modification button redirects to the page for modification of an existing attack.</p> <p>If the attack was executed at least once, download links for the output file generated by the attack script, as well as standard output of the process in which the attack script was run, are available for every execution.</p>
Other comments	The UI is accessible to authenticated and authorized users of the platform via the Web browser running on the user's workstation VM, but only when the AS is a part of the training scenario.

Table 45. UI.11.003 – Attack Simulator, Attack details

User Interface name	CYBERWISER.eu – Attack Simulator, Creating a new attack
UI Id	UI.11.004
Description	This page allows the user to configure all aspects of an attack task.
Purpose	Configuration and subsequent creation of a new attack task.
Navigation and user interaction	<p>The user is guided through the process of attack task configuration by means of a wizard comprising several steps:</p> <ol style="list-style-type: none"> <li>1.) <i>choice/configuration of an attack tool and attack script.</i> As a starting point, the user can choose an attack script from the set of attack scripts available for the scenario, download it, modify it on his workstation, and upload it. Alternatively, he can start from scratch and upload his own attack script and select the appropriate attack tool to run his script;</li> <li>2.) <i>(optional) Uploading of additional files referenced in the attack script;</i></li> <li>3.) <i>(optional) Configuration of custom attack timeout and schedule</i> via numeric input fields and checkboxes. It will be possible to customize total number of executions, the interval between consecutive attack executions, and the delay before initial execution.</li> </ol> <p>Once the above sequence of steps is complete, a new task is created (but not yet submitted), and the user is redirected to the details page of the newly created attack task, from where he can submit it.</p>
Other comments	The UI is accessible to authenticated and authorized users of the platform via the Web browser running on the user's workstation VM, but only when the AS is a part of the training scenario.

Table 46. UI.11.004 – Attack Simulator, Creating a new attack

User Interface name	CYBERWISER.eu – Attack Simulator, Modifying an attack
UI Id	UI.11.005
Description	This page allows the user to modify an existing attack task. Specifically, it is possible to modify the timeout value of an attack, total number of executions and the interval between consecutive executions.
Purpose	Support for modification (reconfiguration) of an existing attack task.
Navigation and user interaction	This page will be presented either as a subpage on the attack details page or as a modal. The user will be able to modify configurable aspects of the attack task via numeric input fields and checkboxes. Clicking a button for modification will result in immediate updating of the attack task, after which the user will be redirected to the page with attack task details.
Other comments	The UI is accessible to authenticated and authorized users of the platform via the Web browser running on the user's workstation VM, but only when the AS is a part of the training scenario.

Table 47. UI.11.005 – Attack Simulator, Modifying an attack

User Interface name	Countermeasures Simulator – Proposed countermeasures
UI Id	UI.12.001
Description	This interface shows a list of proposed mitigation measures that the user may apply in the light of the current scenario. If there is no risk at the moment, the list of countermeasures will appear empty. If the risk goes over a certain threshold, this will trigger the mitigation of certain measures. The offered information will depend on the difficulty level at which the CS is working, which is established by means of contextualization. These difficulty levels are explained in Section 5.13.
Purpose	The purpose of this interface is to propose different alternatives in terms of mitigation measures that may be applied according to the cyber risk to which the infrastructure is exposed.
Navigation and user interaction	By clicking on a specific user interface, a link will redirect to another interface entitled Countermeasures Simulator – Mitigation details and execution (explained below). In a separate tab, the user can find some information about the history of applied mitigations and how it has affected the available budget
Other comments	

Table 48. UI.12.001 – Countermeasures Simulator – Proposed countermeasures

User Interface name	Countermeasures Simulator – Mitigation details and execution
UI Id	UI.12.002
Description	<p>This interface provides some text giving details about a specific mitigation.</p> <p>Under this text, the syntax for mitigation execution is explained, emphasizing the parameters that have to be passed for correct execution.</p> <p>There is a text field in which the user can write the command to execute the mitigations, following the explanations given in this interface. There is a button to confirm the command, call the script in the backend and execute the mitigation. The effect will be similar to executing the mitigation from a console. When executing the countermeasure, if something went wrong a pop-up will appear showing an error message.</p> <p>Finally, if the user wants to download the script stored in the backend to freely edit it and try it in a separate console in an autonomous way, there is a button to do that.</p>
Purpose	The purpose is to provide details about different proposed mitigation measures to help the user with the decision on the mitigation strategy to apply.
Navigation and user interaction	<p>The user will write a command in the text field and will click on the existing button to confirm and execute the measure.</p> <p>There is also a download button to get the mitigation script.</p> <p>If the user decides not to apply the measure, can go back to the interface entitled <i>Countermeasures Simulator- Proposed countermeasures</i>, explained above.</p>
Other comments	

Table 49. UI.12.002 – Countermeasures Simulator – Mitigations details and execution



User Interface name	Countermeasures Simulator – Mitigations history and remaining budget
UI Id	UI.12.003
Description	<p>This interface shows the history of mitigation measures that have been applied during the exercises.</p> <p>A table will list the measures including the following fields</p> <ul style="list-style-type: none"> <li>• Sequence number</li> <li>• Name of the mitigation</li> <li>• Explanation of the mitigation (short sentence and full explanation pops up when hovering the mouse over the mitigation)</li> <li>• Timestamp to mark the moment when the mitigation was applied.</li> <li>• Price of the mitigation in the simulated budget</li> <li>• Rating of the mitigation</li> <li>• Remaining budget when the mitigation was applied.</li> </ul> <p>The user can sort the mitigations in ascending and descending way choosing one of these fields: sequence number, timestamp, price, rating and remaining budget.</p>
Purpose	The purpose of this interface is to provide with a recap of all mitigations applied.
Navigation and user interaction	The user can use some fields to sort the mitigations. There is a tab to go to the interface entitled Countermeasures Simulator – Proposed Countermeasures.
Other comments	

Table 50. UI.12.003 - Countermeasures Simulator – Mitigations history and remaining budget

User Interface name	Economic Risk Models
UI Id	UI.13.000
Description	
Purpose	
Navigation and user interaction	
Other comments	The Economic Risk Models are developed using the CORAS graphical tool and any kind of editor to implement the models in the R languages. These interfaces are not part of the CYBERWISER.eu platform. Nevertheless, how to model and how to use these tools are part of the envisioned learning path.

Table 51. UI.13.000 – Economic Risk Models

User Interface name	Message Broker
UI Id	UI.14.000
Description	
Purpose	
Navigation and user interaction	
Other comments	The user will not be offered specific interface of the Message Broker

Table 52. UI.14.000 – Message Broker

User Interface name	CYBERWISER.eu – Centralized Logging Component Dashboard
UI Id	UI.15.001
Description	Dashboard for viewing the data stored in the CLC.
Purpose	Display of various kinds of data stored by CYBERWISER.eu components in the CLC, including logging data, exercise events emitted by AS and VAT, vulnerability reports, performance evaluation reports and possibly others. This dashboard is available to platform operators and the white team (trainers).
Navigation and user interaction	The user interacts with the dashboard via Kibana application deployed as a part of the CLC, specifically via the <i>Dashboard</i> tab in Kibana's main menu. From the dashboard page, he can click the <i>refresh</i> button to obtain the latest data and observe how it changes in real time. He can also enter a query to display only specific data, sort the displayed data on different fields and manipulate other aspects of how the data is displayed.
Other comments	The dashboard is configured in a way that fits the needs of CYBERWISER.eu. The screenshot below depicts the dashboard showing logging data and exercise events, however the dashboard may be refined with data types not displayed here. Further, presentation of the data in the CLC may be complemented with different visualisations later on.

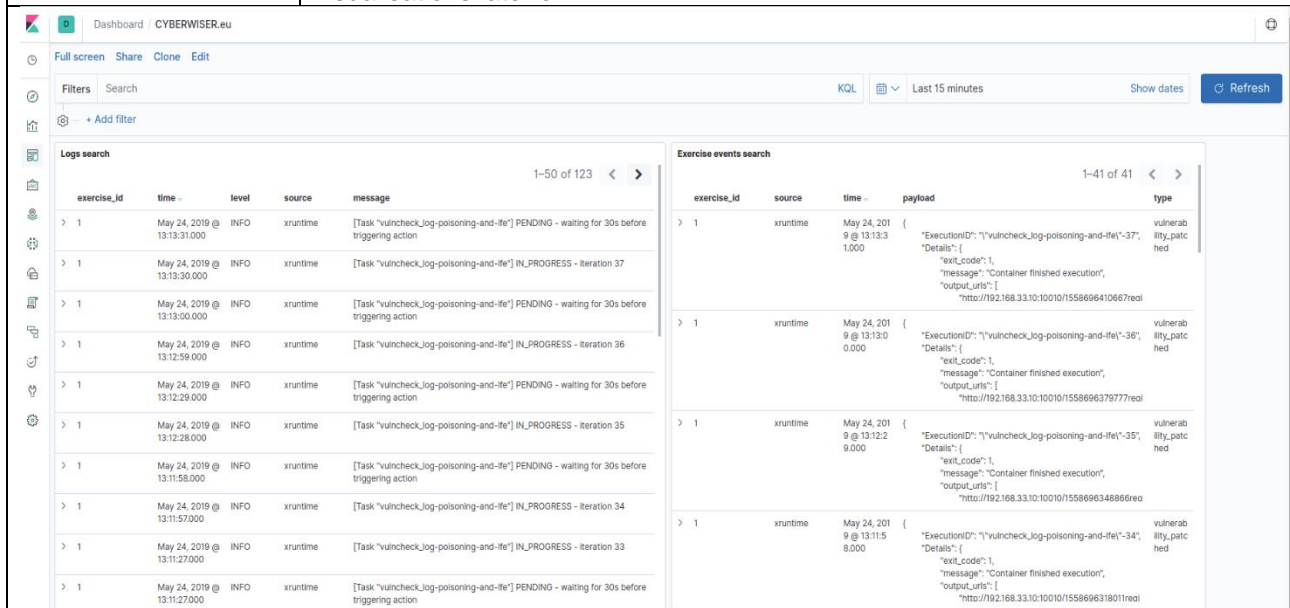


Table 53. UI.15.001 – Centralized Logging Component Dashboard

User Interface name	Infrastructure as a Service
UI Id	UI.16.000
Description	
Purpose	
Navigation and user interaction	
Other comments	Users of the platform will not be accessing this interface

Table 54. UI.16.000 – Infrastructure as a Service

User Interface name	Event-Based Service Orchestrator
UI Id	UI.17.000
Description	
Purpose	
Navigation and user interaction	
Other comments	No specific interface of the Event-Based Service Orchestrator will be offered to users. EBSO is indirectly accessed through both VAT and AS user interfaces.

Table 55. UI.17.000 – Event-Based Service Orchestrator

The figure below summarizes the relations among the different interfaces, offering a high-level view of the user flows that may take place when interacting with CYBERWISER.eu.

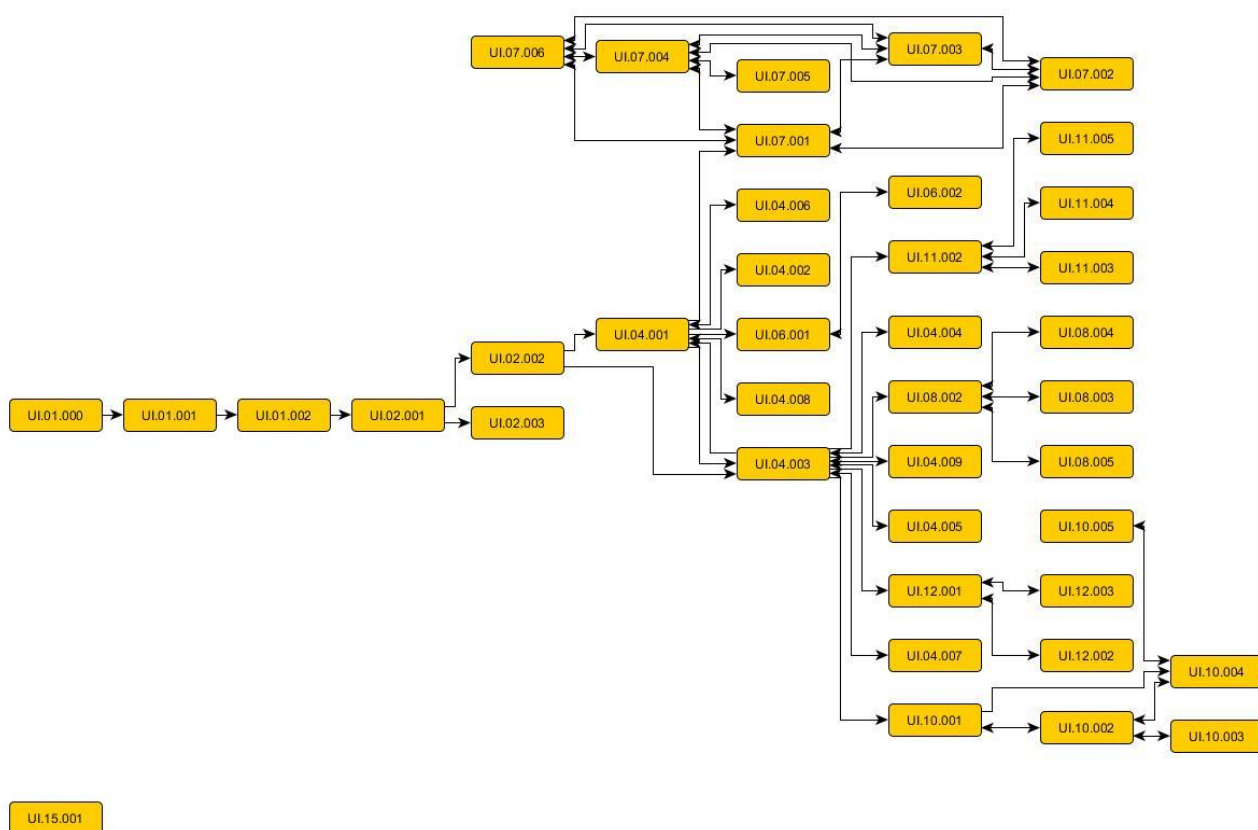


Figure 37. Relations among the user interfaces

## 6.2 Use cases

### 6.2.1 Website

Use Case Name	Website
UC id	UC.01.000
Description	The website offers to all users the single access point to the CYBERWISER.eu Platform through its Register/Login page.
Actors	Users, both trainers and trainees Login Page
Objects	Users Website Login page
Basic flow	Users land on the Website and can click on "Register"/"Login" to access the CYBERWISER.eu Login page which is required to access the CYBERWISER.eu platform.
Preconditions	
Postconditions	The Website must correctly link and timely load the Login page.
Dependencies	

Table 56. UC.01.000 – Website

#### Website sequence diagram

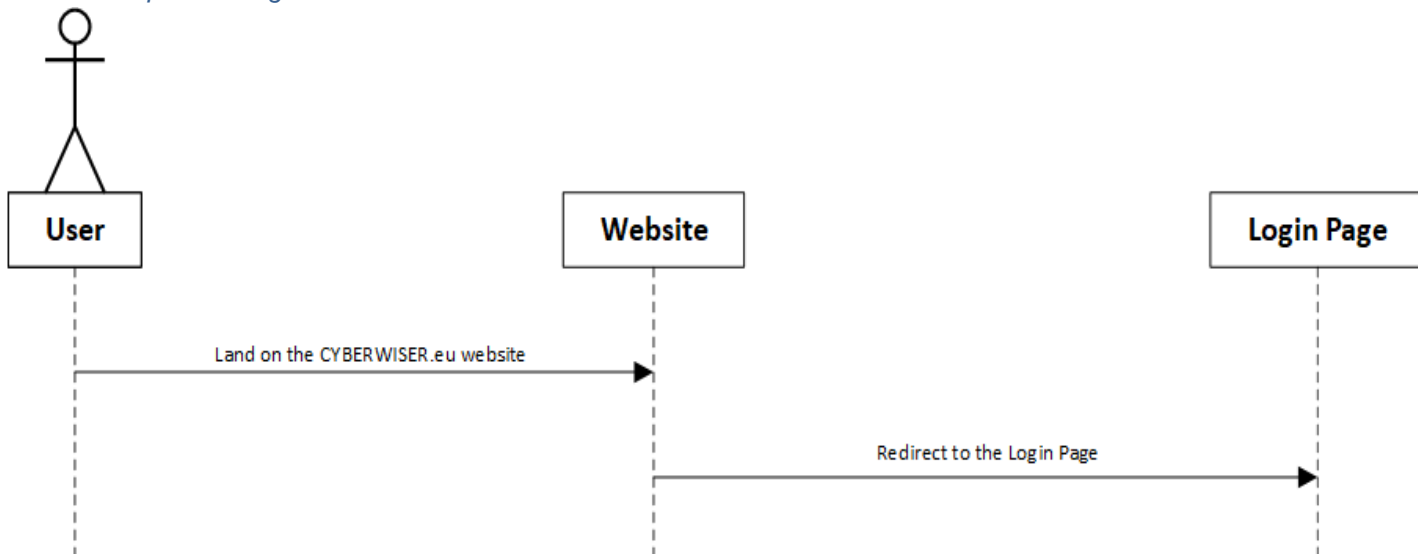


Figure 38. Website sequence diagram

## 6.2.2 Login page

Use Case Name	Login page user interaction
UC id	UC.01.001
Description	This interface has the goal of being the point to introduce the credentials to access the platform and its services.
Actors	Trainers Trainees
Objects	Users (both trainers and trainees) Website Login page
Basic flow	Users land on the Login page. They can register inserting basic data which will be used through the CYBERWISER.eu Platform, login if they previously register or request a new password. In case a user register for the first time the system automatically sends an email with further instructions to the email address indicated by the user. In case a user successfully login the user is redirect to the CYBERWISER.eu private area. In case a user requests a new password the system automatically send an email with further instructions to the email address indicated by the user.
Preconditions	The Website must correctly link and timely load the Login page.
Postconditions	The system automatically sends instruction to the user for registration and new password request and timely load the CYBERWISER.eu private area in case of successful login.
Dependencies	Proper functioning of UC.01.000

Table 57. UC.01.001 – Login page user interaction

### Login page sequence diagram

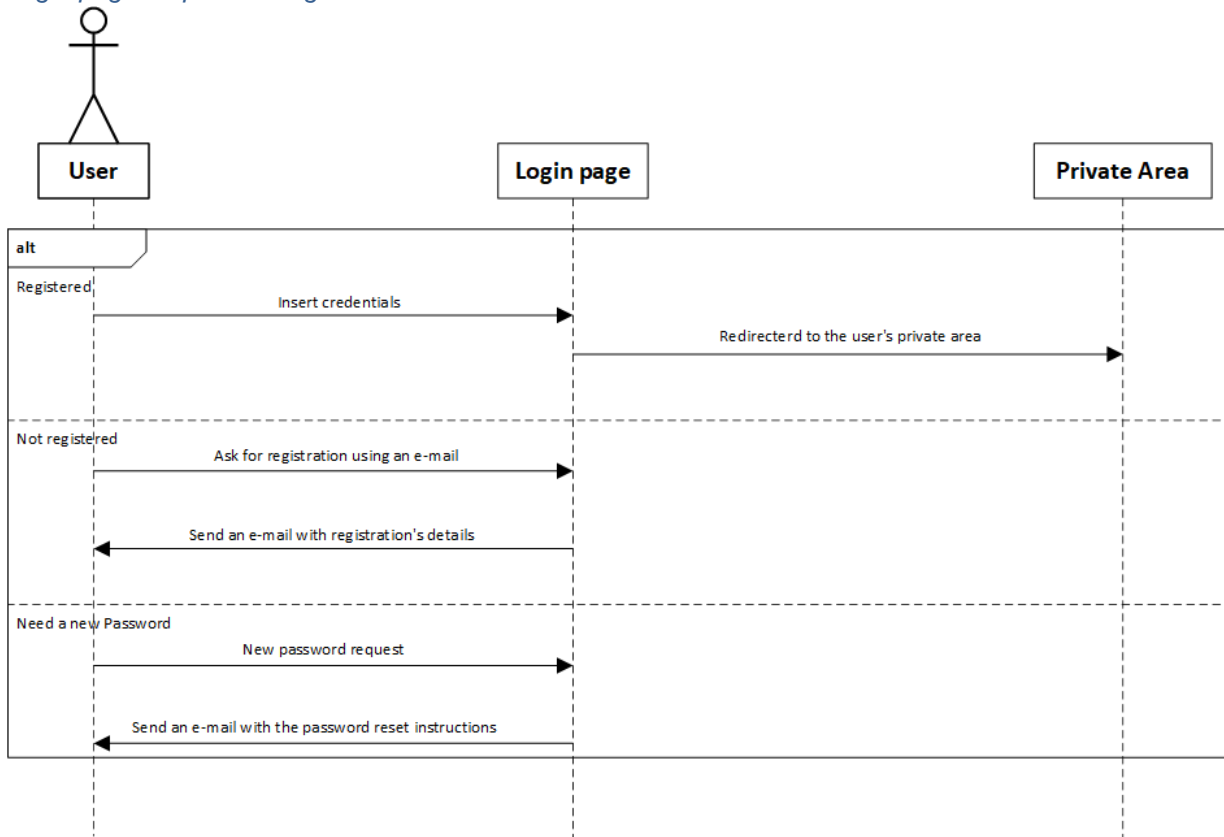


Figure 39. Login page sequence diagram

### 6.2.3 Private area

Use Case Name	CYBERWISER.eu private area user interaction
UC id	UC.01.002
Description	The purpose of this interface is to make available resources addressed to the members of the CYBERWISER.eu Community upon authentication of the user.
Actors	Trainers Trainees
Objects	Users (both trainers and trainees) Website Login page CYBERWISER.eu Workspace area
Basic flow	In case a user successfully login in the Login page the CYBERWISER.eu private area on the website is automatically shown by means of a simple banner temporarily labelled "Cross Learning Facilities".
Preconditions	The user must be successfully logged in to see the private area banner.
Postconditions	The system must show the private area banner only in case a user successfully login.
Dependencies	Successful user login in UC.01.001

Table 58. UC.01.002 – CYBERWISER.eu private area user interaction

#### Private area sequence diagram

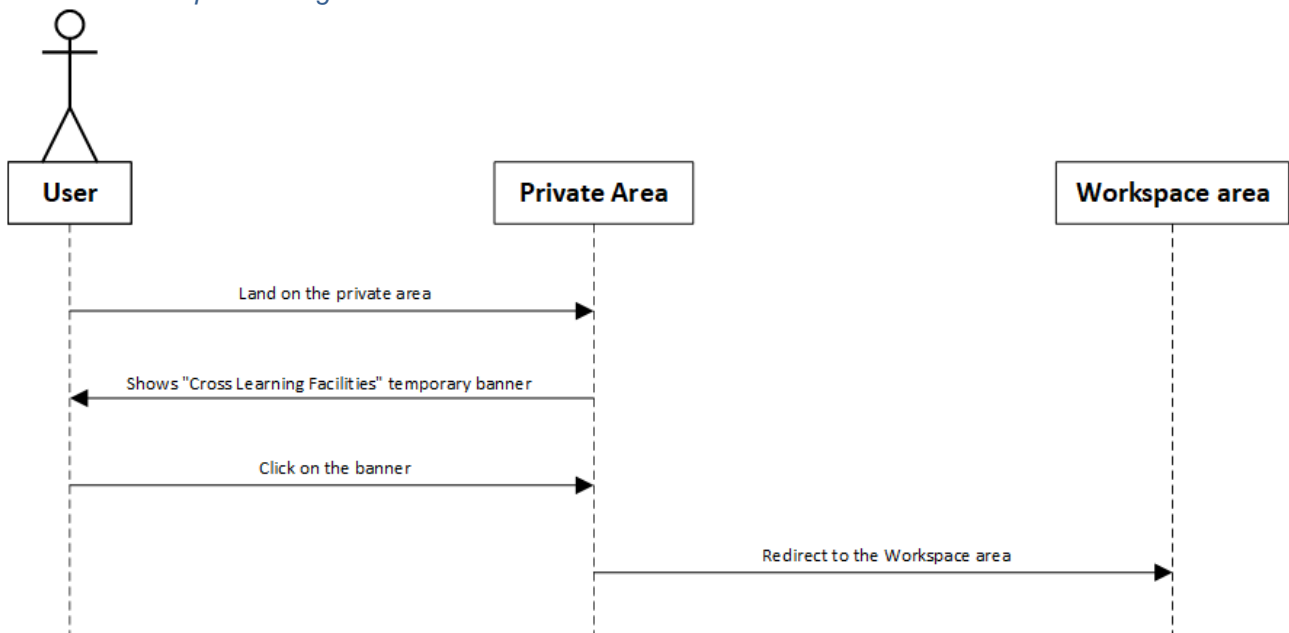


Figure 40. Private area sequence diagram

### 6.2.4 Workspace area

Use Case Name	CYBERWISER.eu Workspace area user interaction
UC id	UC.02.000
Description	The workspace area is a summary of the different groups in which a user is involved, and links to the different resources that the user is made available as a consequence of belonging to a specific group. There is a dependency between the resources that are made available and the group in question.
Actors	Trainers Trainees

Objects	Users (both trainers and trainees) CYBERWISER.eu Workspace area Moodle Dashboard
Basic flow	Once a user clicks on the "Cross Learning Facilities" banner he is redirected to the Workspace area which allow the user to access the actual courses he can take as well as the different resources namely the e-learning platform, the cyber range platform and a file repository that can be used to store documents or files.
Preconditions	The user must be successfully logged in to see the private area banner and therefore access the Workspace area.
Postconditions	The system must correctly link and timely load the e-learning platform, the cyber range platform and the file repository.
Dependencies	Successful user login in UC.01.001

Table 59. UC.02.000 – CYBERWISER.eu workspace area user interaction

*Workspace area sequence diagram*

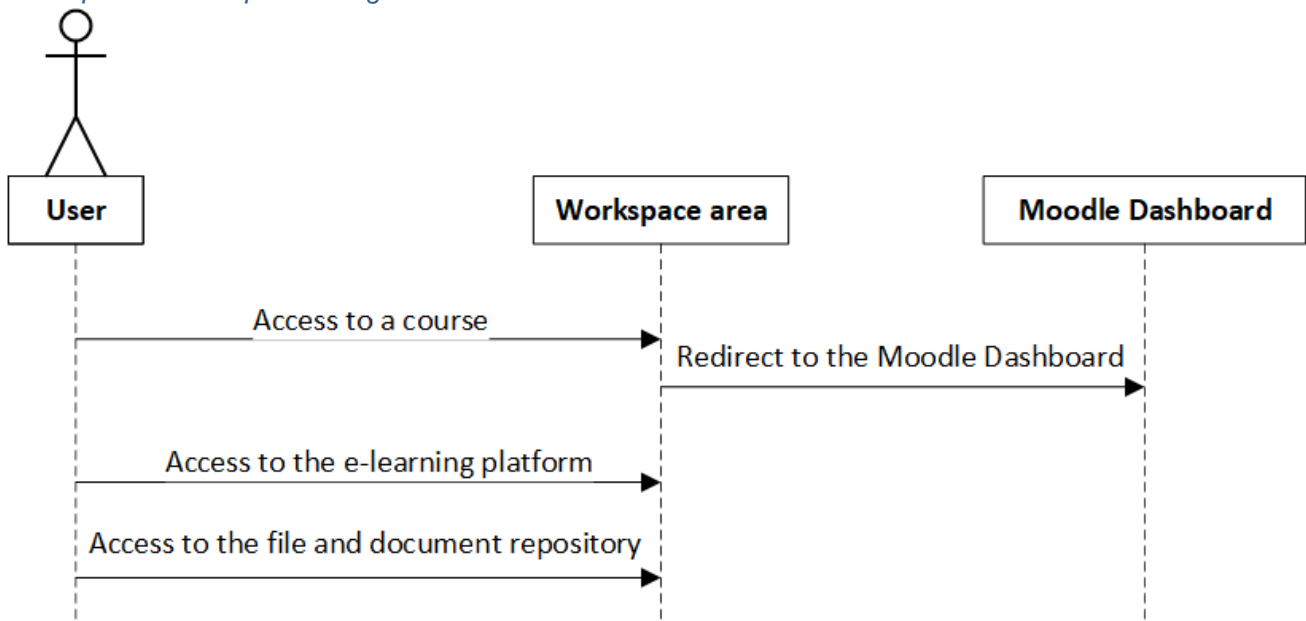


Figure 41. Workspace area sequence diagram



### 6.2.5 Moodle Dashboard – Trainee interaction

Use Case Name	CYBERWISER.eu Moodle dashboard – Trainee interaction
UC id	UC.02.001
Description	This functionality links the trainee who take a specific course to specific exercises held in the Training Manager.
Actors	Trainees
Objects	Users (trainers) Moodle Dashboard Courses Training Manager
Basic flow	Once a user clicks on a course this will be showed in the Cross-Learning Facilities through Moodle. As part of the course the user could be then directed to the cyber range platform through the training Manager and redirected to the Moodle in the Cross-Learning Facilities once the exercise in the cyber range is completed.
Preconditions	The system must correctly link courses to specific exercises in the Training Manager.
Postconditions	The Training Manager must correctly link the user back to the Moodle.
Dependencies	Correct loading of Workspace resources in UC.02.000

Table 60. UC.02.001 CYBERWISER.eu Moodle dashboard – trainee interaction

#### Moodle dashboard – Trainee interaction sequence diagram

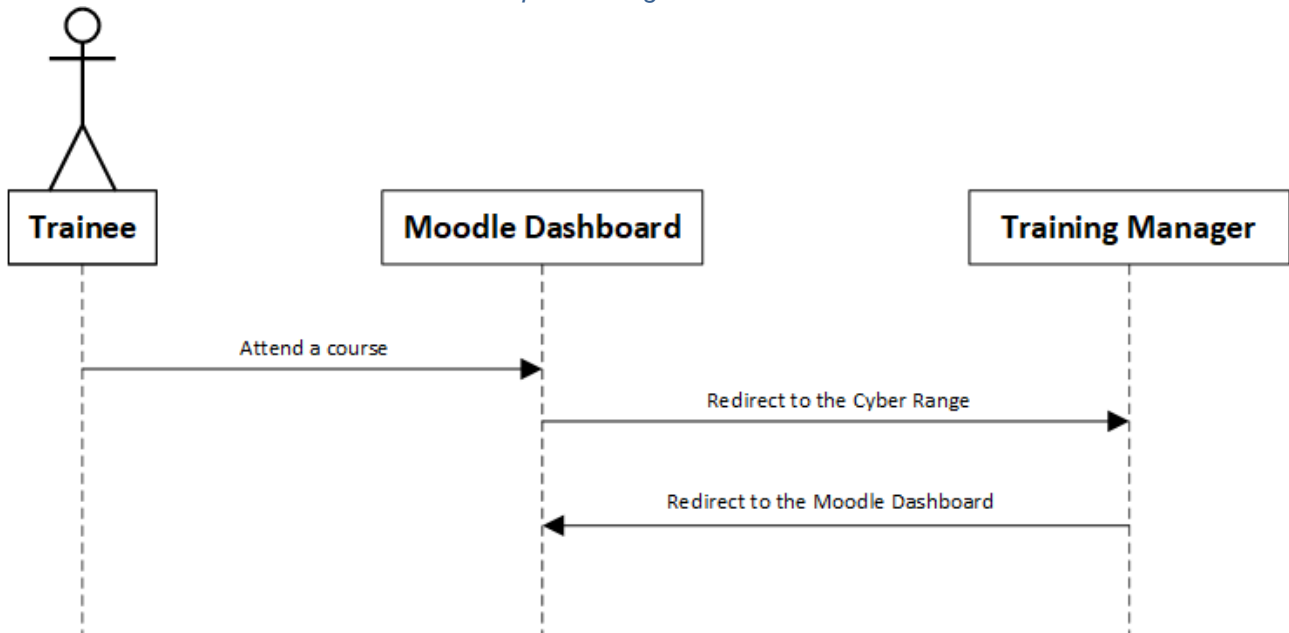


Figure 42. Moodle dashboard – trainee interaction sequence diagram

### 6.2.6 Moodle dashboard – Trainer interaction

Use Case Name	CYBERWISER.eu Moodle dashboard – Trainer interaction
UC id	UC.02.002
Description	This functionality allows the trainer to have administrative functionalities to shape the courses and monitor the trainee's activities.
Actors	Trainers
Objects	Users (trainers) Moodle Dashboard Courses
Basic flow	Once a trainer clicks on a course is presented with some administrative functionalities duly detailed in section 6.1 of this document.
Preconditions	The system must identify the trainer in order to grant him administrative functionalities.
Postconditions	The system must correctly save and present changes made by the trainer in the course.
Dependencies	Correct loading of Workspace resources in UC.02.000

Table 61. UC.02.002 – CYBERWISER.eu Moodle dashboard – Trainer interaction

#### *Moodle dashboard – Trainer interaction sequence diagram*

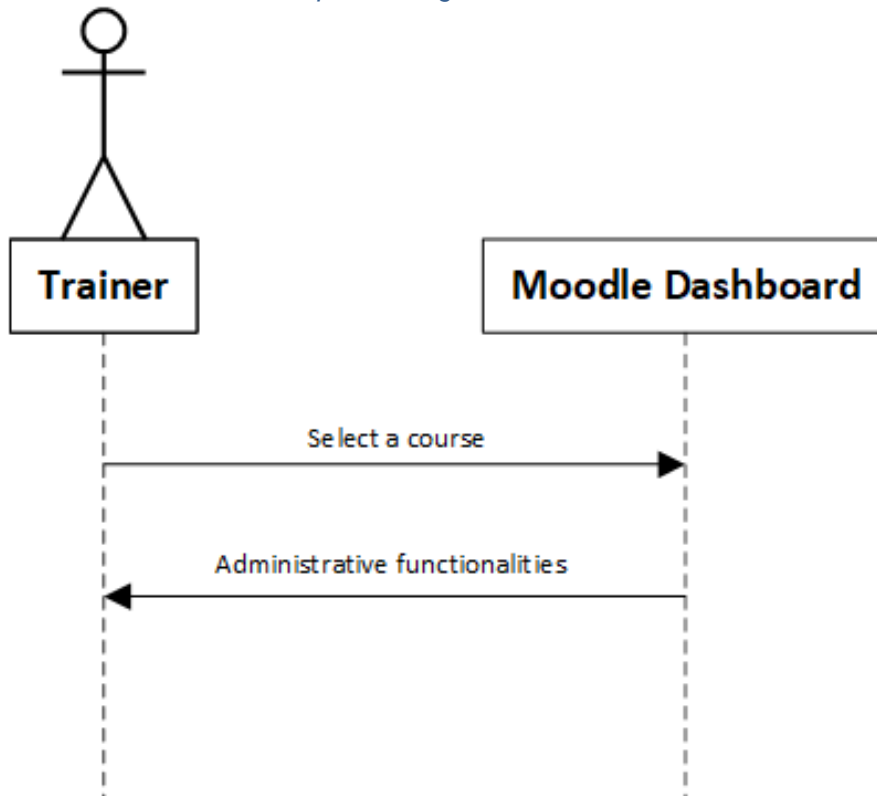


Figure 43. Moodle dashboard – Trainer interaction sequence diagram

### 6.2.7 Creating and using new Assets

Use Case Name	Creating and assigning a new asset
UC id	UC.06.001
Description	In order to use existing assets in the design of a scenario, they need to be configured in the Digital Library by a user after which they can be used in the design of a new scenario
Actors	Operator, Trainer
Objects	Digital Library, Training Manager
Basic flow	To define a new Asset, the Operator navigates to the Digital Library and fills out the details of the new Asset and saves it. After that a Trainer, when accessing a new or pre-existing scenario will be able to assign the new Asset to a node in the Scenario from the list of available Assets that the Training Manager has retrieved from the Digital Library.
Preconditions	Users are authenticated, Scenario has been created
Postconditions	One or more Assets are added and assigned as part of a scenario
Dependencies	

Table 62. UC.06.001 – Creating and assigning a new asset

#### Creating and assigning a new asset sequence diagram

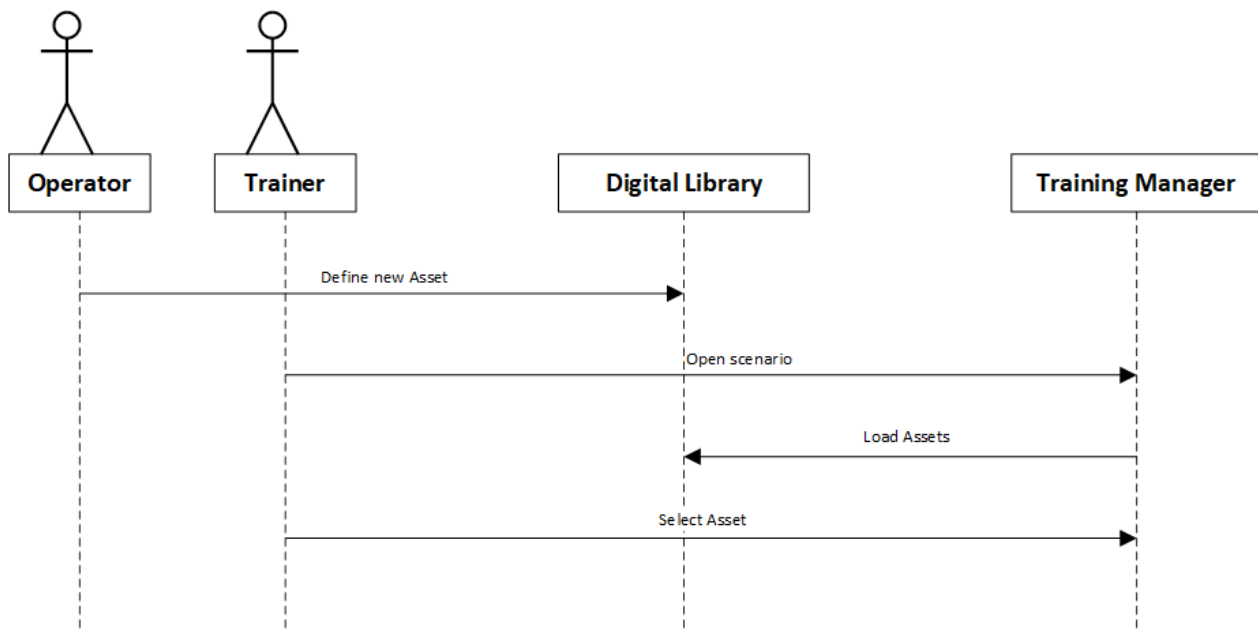


Figure 44. Creating and assigning an Asset

### 6.2.8 Designing a new Scenario

Use Case Name	Designing a new scenario
UC id	UC.04.001
Description	This use case describes the overall flow used to create a new scenario from scratch.
Actors	Operator, Trainer
Objects	Training Manager, Digital Library
Basic flow	When a Trainer has logged in, he or she can create a new blank scenario, by providing its basic details like a name and a description. After that the Scenario Designer can be used to define the content of the different layers that make up a Scenario. The training itself can be specified on the Training layer, the application landscape can be defined using the Application layer and the virtual machines and network can be defined on the Network layer. After these definitions are done, the user can save the scenario at which point the Training Manager will perform the basic validation of the Scenario. If it is valid, the scenario is stored. The Trainer can then decide to submit the scenario for validation, after which an operator user will perform a manual validation. This manual validation includes the creation of possible missing assets. After the assets have been created and assigned, the scenario is validated and ready for instantiation.
Preconditions	Users are authenticated
Postconditions	A new scenario is validated, ready for instantiation
Dependencies	

Table 63. UC.04.001 – Designing a new scenario

#### *Designing a new scenario sequence diagram*

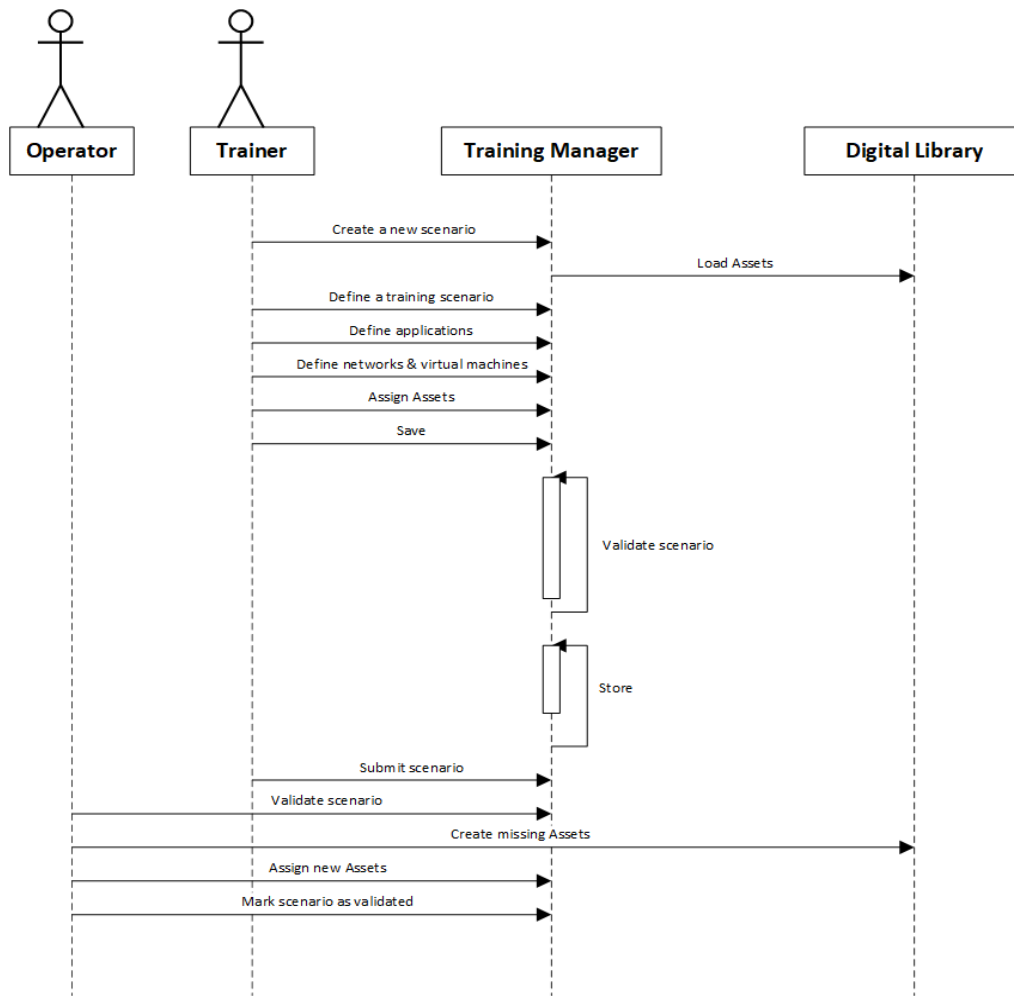


Figure 45. Designing a new scenario

### 6.2.9 Using a designed Scenario

Use Case Name	Using a designed scenario
UC id	UC.04.002
Description	The use case describes the flow related to ensuring a Trainee can use a validated, scenario.
Actors	Operator, Trainer, Trainee
Objects	Training Manager, Digital Library, Simulated Infrastructure Manager, IaaS, Anomaly Detection Reasoner
Basic flow	Starting from a validated scenario, an Operator will create a scenario template for the scenario. This will result in the creation of the relevant networks and virtual machine templates at the IaaS provider through the Training Manager and Simulated Infrastructure Manager. After that the Trainer will be able to provide access to the Trainee. This involves assigning the right permissions, virtual machine access and node visibility. The latter defines what the Trainee can see when looking at the scenario design. The scenario can now be instantiated by the Trainer. After the instantiation is triggered, the ADR is notified of the fact that a new scenario is running. When the scenario is running the Trainee can access the scenario, inspect the scenario design based on the assigned node visibility, and access VMs to which he or she has been granted access, provided the VM has booted up. The VM status is polled through the SIM and shown to the user.
Preconditions	A validated scenario, authenticated users
Postconditions	Trainee is using the validated scenario
Dependencies	

Table 64. UC.04.002 - Using a designed scenario

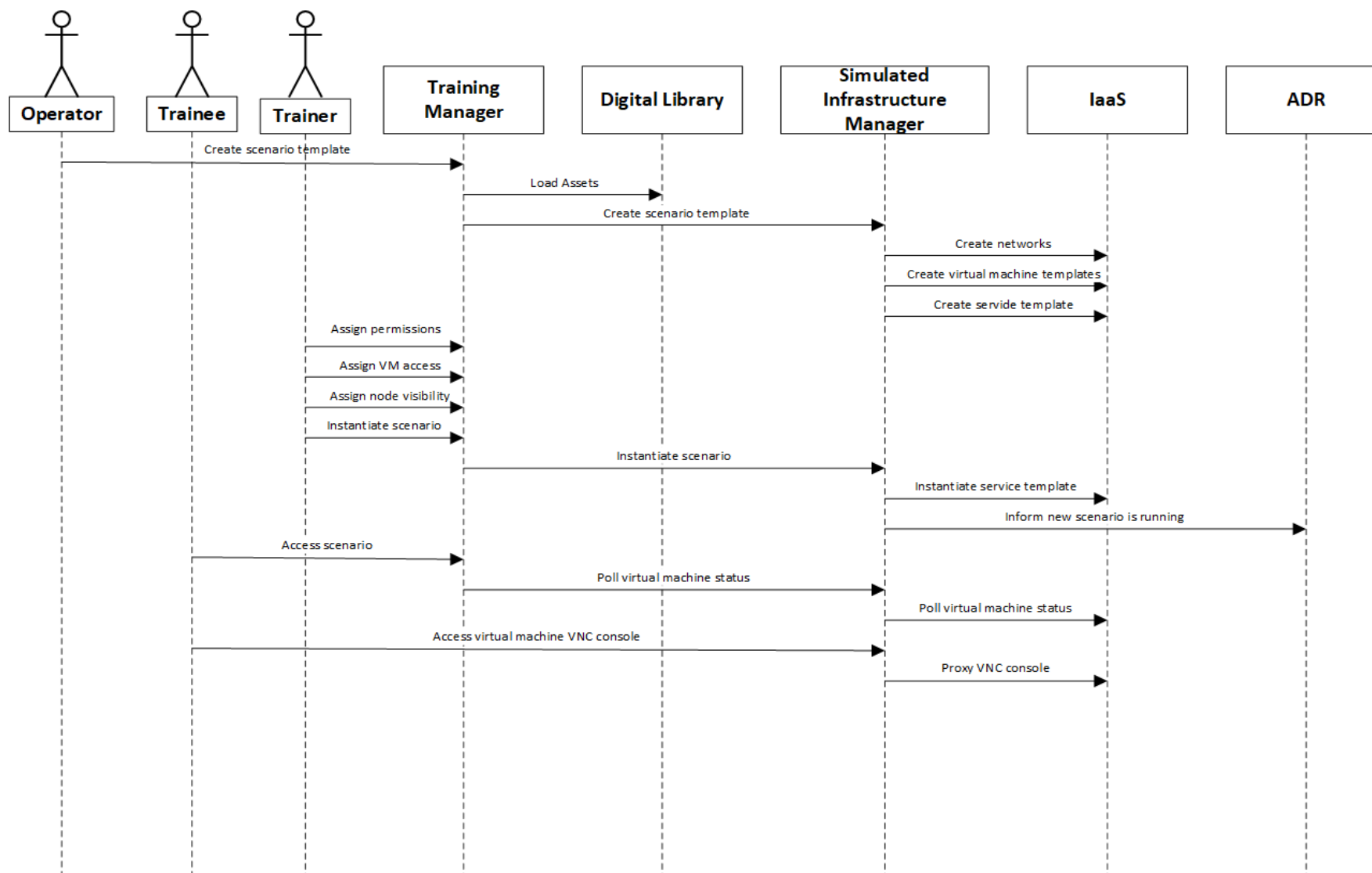


Figure 46. Using a designed scenario



### 6.2.10 Vulnerability Assessment Tools – Login Page

Use Case Name	Vulnerability Assessment Tools – Login Page
UC id	UC.08.001
Description	Offers the user a point to authenticate and enter the VAT Scan Configurator.
Actors	Trainer Trainee
Objects	User Web server EBSO back-end
Basic flow	When accessing the login page, the user is given a login form to introduce their username and password. When presenting the credentials, web server confirms the user's authorization with the back-end service and, if the authorization is successful, returns a session token and redirects user to the VAT main page. In case of an invalid login, an error notice is displayed on the login page and the user can try to enter credentials again. The sequence diagram shows only the positive flow.
Preconditions	EBSO back-end must have received the user credentials through contextualization.
Postconditions	In case of a correctly authenticated and authorized user, the user must have a valid session token and be able to use Scan Configurator. In case of an invalid login, the user must be denied access and offered the login page again.
Dependencies	

Table 65. UC.08.001 – Vulnerability Assessment Tools – Login Page

#### Vulnerability Assessment Tools – Login Page Sequence Diagram

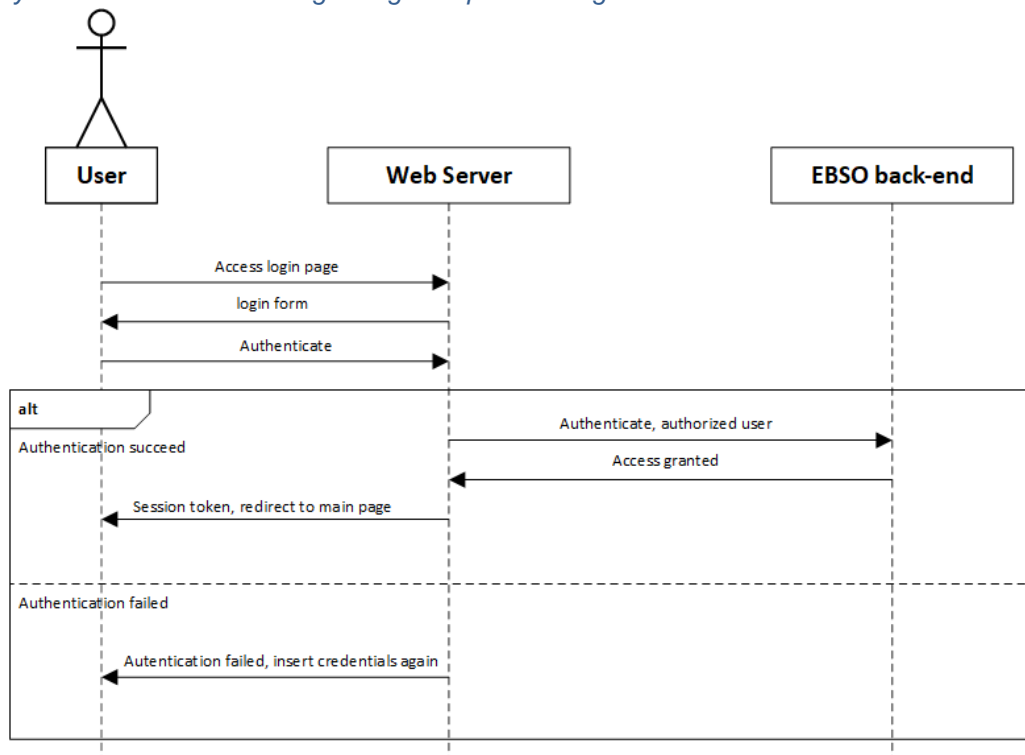


Figure 47. VAT – Login Page Sequence Diagram

### 6.2.11 Vulnerability Assessment Tools – Main Page

Use Case Name	Vulnerability Assessment Tools – Main Page
UC id	UC.08.002
Description	Provides an overview of past and pending vulnerability scans to the user.
Actors	Trainer Trainee
Objects	User Web server EBSO back-end
Basic flow	When a user requests the VAT main page (e.g. after being redirected from a successful login), the web server requests the appropriate data from the back-end service, according to the specific logged-in user connected. If the user is a trainer, they can see data about all scans requested by any user as well as the scans executed by the platform (described in the scenario template). In case the user is a trainee, they are only shown data about scans requested by themselves.
Preconditions	User must be logged in and have a valid session token identifying them with the VAT web server.
Postconditions	User is shown the appropriate scans information.
Dependencies	UC.08.001

Table 66. UC.08.002 – Vulnerability Assessment Tools – Main Page

#### *Vulnerability Assessment Tools – Main Page Sequence Diagram*

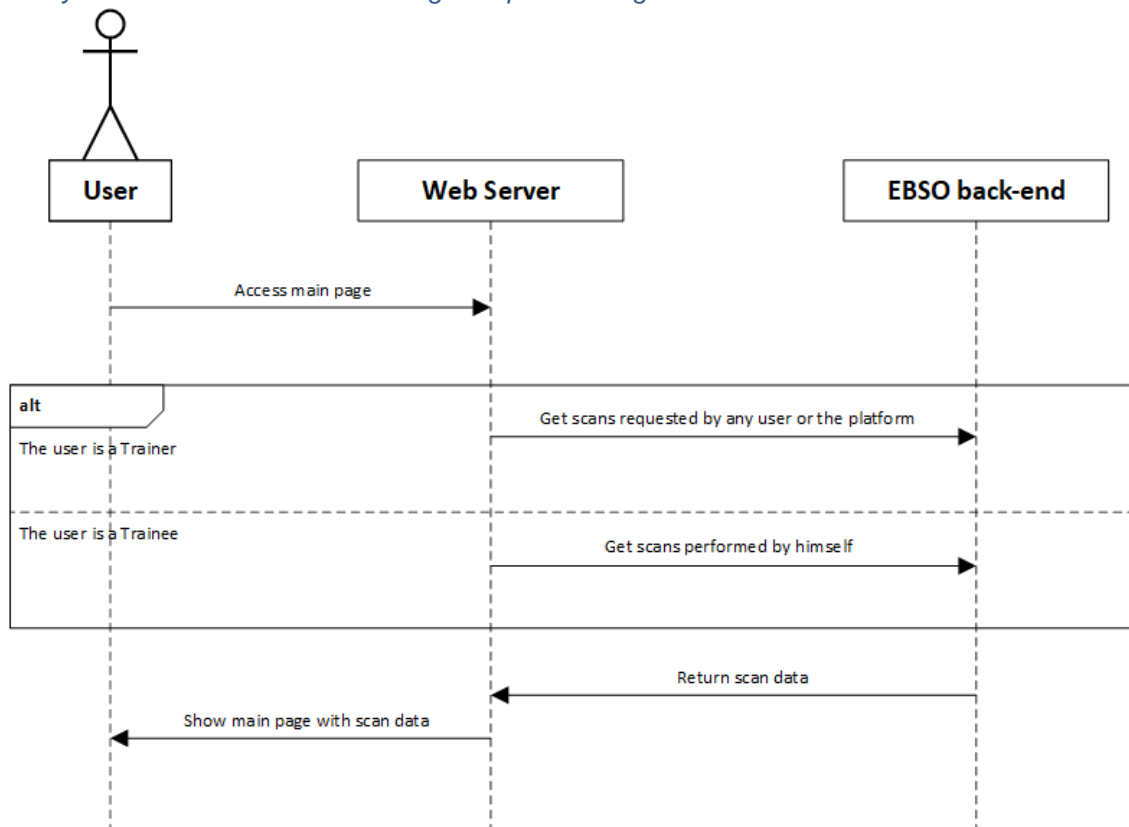


Figure 48. VAT – Main Page

### 6.2.12 Vulnerability Assessment Tools – Scan Details Page

Use Case Name	Vulnerability Assessment Tools – Scan Details Page
UC id	UC.08.003
Description	Provides detailed information about a vulnerability scan task.
Actors	Trainer Trainee
Objects	User Web Server EBSO back-end
Basic flow	When a user requests details of an individual scan task (from the VAT Main Page), the web server requests and gathers data from the EBSO back-end. In this process, the back-end component checks whether the user has access to the specific scan and the positive case returns the data, including scan task schedule, settings, list of past executions with timestamps and results, etc. This data is displayed to the user.
Preconditions	User must be logged in and have a valid session token identifying them with the VAT web server. The requested task details have to be visible to the user (trainees can only access their own tasks, while trainers can access all tasks).
Postconditions	User is shown the requested scan task details.
Dependencies	UC.08.002

Table 67. UC.08.003 – Vulnerability Assessment Tools – Scan Details Page

#### Vulnerability Assessment Tools – Scan Details Sequence Diagram

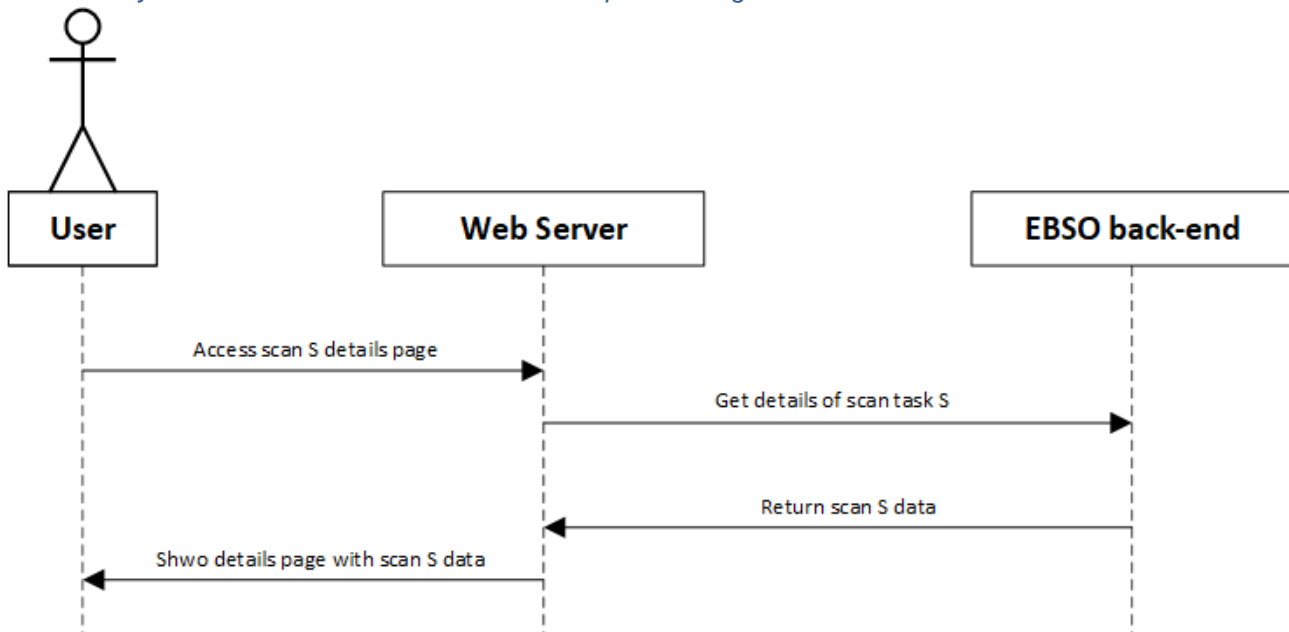


Figure 49. VAT – Scan Details Sequence Diagram

### 6.2.13 Vulnerability Assessment Tools – Vulnerability Scan Creation

Use Case Name	Vulnerability Assessment Tools – Vulnerability Scan Creation
UC id	UC.08.004
Description	Provides the ability to create and configure a new vulnerability scanning task.
Actors	Trainer Trainee
Objects	User Web Server EBSO back-end
Basic flow	<p>Using a link on the VAT Main Page, the user can access a wizard for creating a new vulnerability scan. When requesting the wizard, the user is shown a web page where they are able to choose the type of the scan. Depending on the user's role, a trainer can choose both generic scans and scans with custom scanning scripts. A trainee can only choose the generic vulnerability scans. Sequence diagram on Figure 50 shows both options: generic scans and custom scripts. Note that only one of these two paths are followed, depending on the user's choice.</p> <p>In case of a generic scan, the user is presented a web page for the generic scan configuration. They input the settings (scanner module selection, configuration of the modules, scanning target) and continue the wizard with schedule configuration. If the custom scanning script option is selected, the user must first choose an attack tool and a scanning script to be run, then configure the script's options and (optionally) upload additional files needed by the script.</p> <p>After the scan configuration is sent, a schedule configuration page is presented. Settable schedule parameters are: delay before first execution, interval between executions, number of executions, and a timeout (when to stop the scan if it is running for an unexpectedly long time). The schedule setting can be omitted, in which case the task is run immediately and only once. After the user confirms the configuration, the web server sends it to the EBSO back-end where it is scheduled for execution. A confirmation message is displayed to the user and they are redirected to the VAT Main Page.</p>
Preconditions	User must be logged in and have a valid session token identifying them with the VAT web server.
Postconditions	A new scanning task is created with the requested configuration.
Dependencies	UC.08.002

Table 68. UC.08.004 – Vulnerability Assessment Tools – Vulnerability Scan Creation

*Vulnerability Assessment Tools – Vulnerability Scan Creation Sequence Diagram*

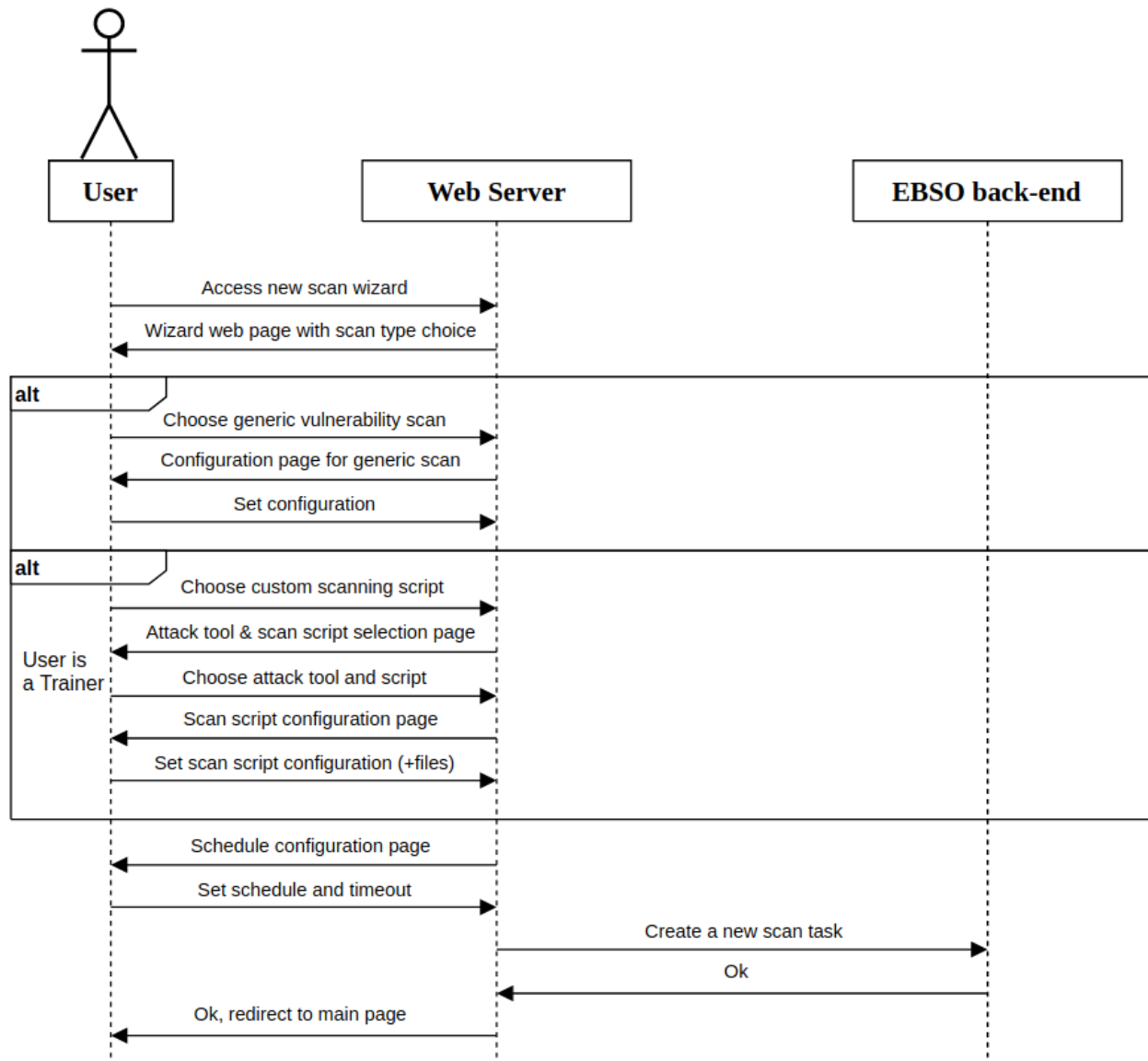


Figure 50. VAT – Vulnerability Scan Creation Sequence Diagram

#### 6.2.14 Vulnerability Assessment Tools – Vulnerability Scan Modification

Use Case Name	Vulnerability Assessment Tools – Vulnerability Scan Modification
UC id	UC.08.005
Description	Provides the ability to modify the settings of an existing vulnerability scanning task or cancel its scheduled executions.
Actors	Trainer Trainee
Objects	User Web Server EBSO back-end
Basic flow	Using a link on the scan details page, a user can request to modify the schedule parameters of an existing scanning task, scheduled for future execution. The user is shown a schedule configuration page, where they can modify the number of future scan executions, the interval between executions, and the timeout value. The task can also be disabled, preventing future executions. After the user confirms the settings, the web server sends the requested values to the EBSO back-end, which updates the task schedule. Finally, a confirmation message is displayed to the user and they are redirected back to the scan details page.
Preconditions	User must be logged in and have a valid session token identifying them with the VAT web server. There needs to be at least one task scheduled for execution, available to the current user.
Postconditions	The scanning task is modified according to the configuration requested.
Dependencies	UC.08.003

Table 69. UC.08.005 – Vulnerability Assessment Tool – Vulnerability Scan Modification

#### Vulnerability Assessment Tools – Vulnerability Scan Modification Sequence Diagram

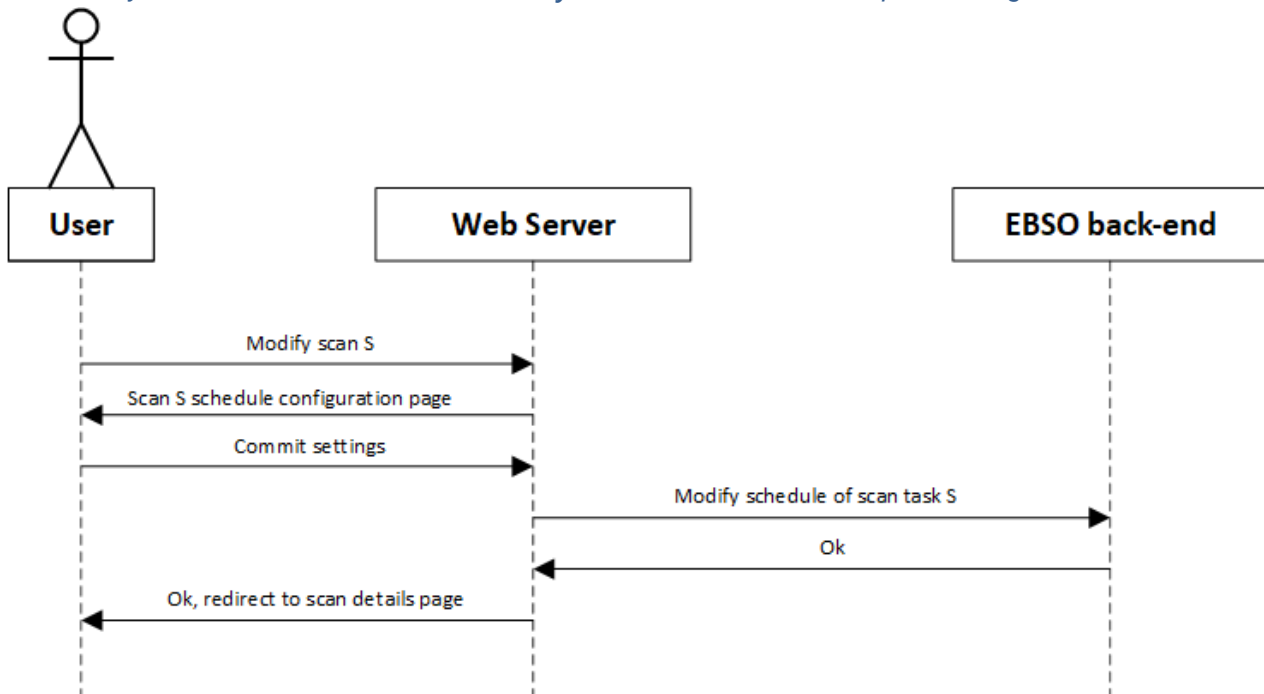


Figure 51. VAT – Vulnerability Scan Modification Sequence Diagram

### 6.2.15 Attack Simulator – Login Page

Use Case Name	Attack Simulator – Login Page
UC id	UC.11.001
Description	Offers the user a point to authenticate and enter the AS Attack Configurator.
Actors	Trainer Trainee
Objects	User Web server EBSO back-end
Basic flow	When accessing the login page, the user is given a login form to introduce their username and password. When presenting the credentials, web server confirms the user's authorization with the back-end service and, if the authorization is successful, returns a session token and redirects user to the AS main page. In case of an invalid login, an error notice is displayed on the login page and the user can try to enter credentials again. The sequence diagram shows only the positive flow.
Preconditions	EBSO back-end must have received the user credentials through contextualization.
Postconditions	In case of a correctly authenticated and authorized user, the user must have a valid session token and be able to use Attack Configurator. In case of an invalid login, the user must be denied access and offered the login page again.
Dependencies	

Table 70. UC.11.001 – Attack Simulator – Login Page



*Attack Simulator – Login Page Sequence Diagram*

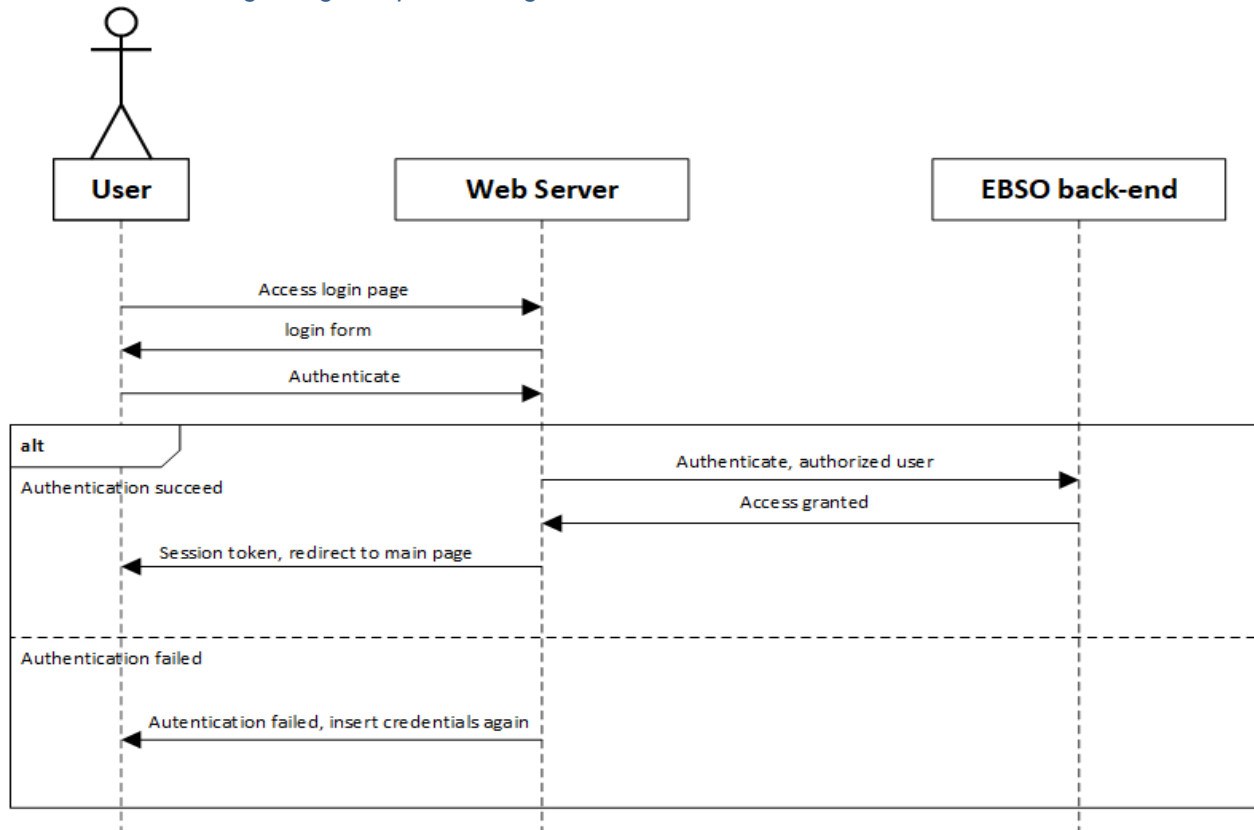


Figure 52. AS – Login Page Sequence Diagram

### 6.2.16 Attack Simulator – Main Page

Use Case Name	Attack Simulator – Main Page
UC id	UC.11.002
Description	Provides an overview of past and pending attacks to the user.
Actors	Trainer Trainee
Objects	User Web server EBSO back-end
Basic flow	When a user requests the AS main page (e.g. after being redirected from a successful login), the web server requests the appropriate data from the back-end service, according to the specific logged-in user connected. If the user is a trainer, they can see data about all attacks requested by any user as well as the attacks executed by the platform (described in the scenario template). In case the user is a trainee, they are only shown data about attacks requested by themselves.
Preconditions	User must be logged in and have a valid session token identifying them with the AS web server.
Postconditions	User is shown the appropriate attacks information.
Dependencies	UC.11.001

Table 71. UC.11.002 – Attack Simulator – Main Page

#### Attack Simulator – Main Page Sequence Diagram

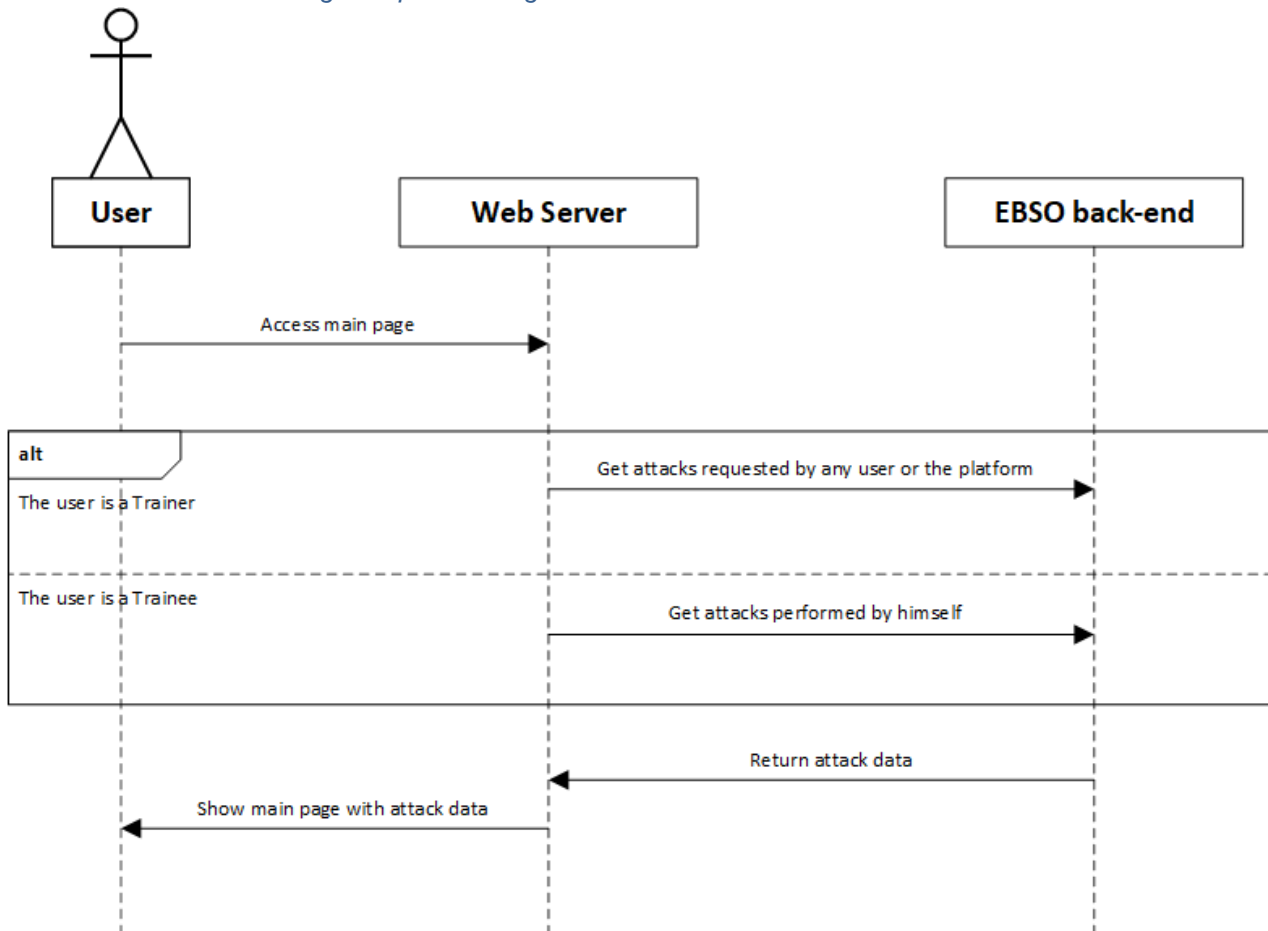


Figure 53. AS – Main Page Sequence Diagram

### 6.2.17 Attack Simulator – Attack Details Page

Use Case Name	Attack Simulator – Attack Details Page
UC id	UC.11.003
Description	Provides detailed information about an attack task.
Actors	Trainer Trainee
Objects	User Web Server EBSO back-end
Basic flow	When a user requests details of an individual scan task (from the AS Main Page), the web server requests and gathers data from the EBSO back-end. In this process, the back-end component checks whether the user has access to the specific attack task and in the positive case returns the required data, including attack task schedule, settings, list of past executions with timestamps and results, etc. This data is displayed to the user.
Preconditions	User must be logged in and have a valid session token identifying them with the AS web server. The requested task details have to be visible to the user (trainees can only access their own tasks, while trainers can access all tasks).
Postconditions	User is shown the requested attack task details.
Dependencies	UC.11.002

Table 72. UC.11.003 – Attack Simulator – Attack Details Page

#### Attack Simulator – Attack Details Page Sequence Diagram

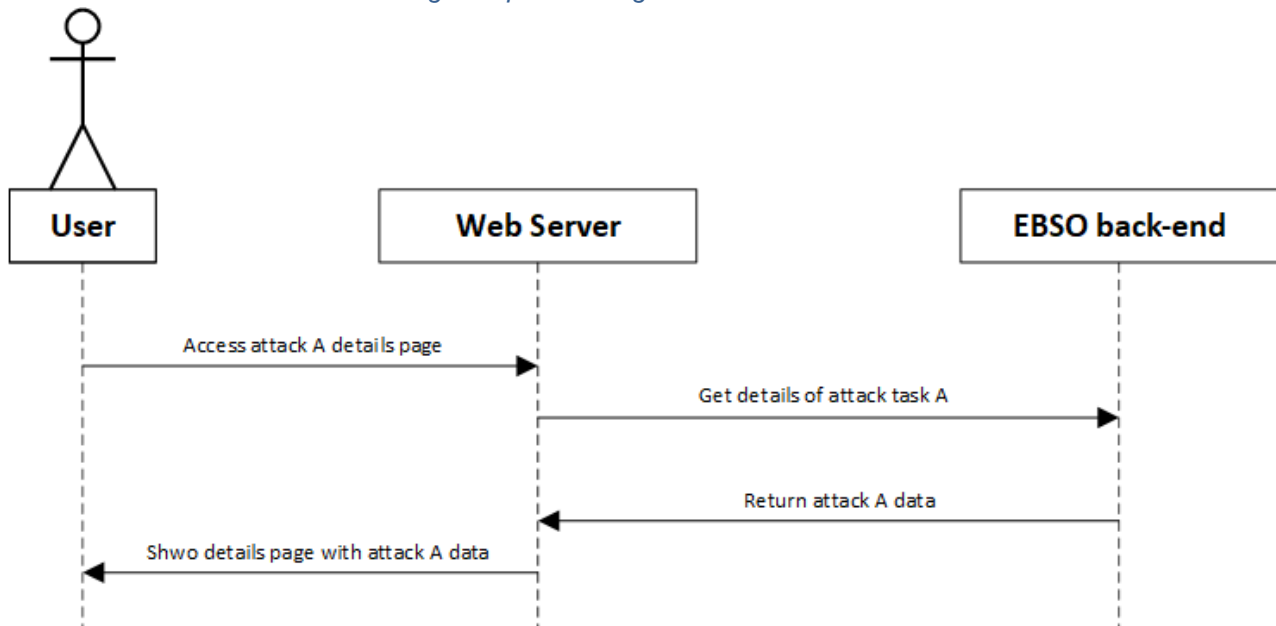


Figure 54. AS – Attack Details Page

### 6.2.18 Attack Simulator – Attack Creation

Use Case Name	Attack Simulator – Attack Creation
UC id	UC.11.004
Description	Provides the ability to create and configure a new attack task.
Actors	Trainer Trainee
Objects	User Web Server EBSO back-end
Basic flow	Using a link on the AS Main Page, the user can access a wizard for creating a new attack task. When requesting the wizard, the user is shown a web page where they are able to an attack tool and a scanning script to be run. After selecting the script, the server returns a new page for configuration of the script's options and (optionally) upload of additional files needed by the script. After the attack configuration is sent, a schedule configuration page is presented. Settable schedule parameters are: delay before first execution, interval between executions, number of executions, and a timeout (when to stop the attack if it is running for an unexpectedly long time). The schedule setting can be omitted, in which case the task is run immediately and only once. After the user confirms the configuration, the web server sends it to the EBSO back-end where it is scheduled for execution. A confirmation message is displayed to the user and they are redirected to the AS Main Page.
Preconditions	User must be logged in and have a valid session token identifying them with the AS web server.
Postconditions	A new attack task is created with the requested configuration.
Dependencies	UC.11.002

Table 73. UC.11.004 – Attack Simulator – Attack Creation

*Attack Simulator – Attack Creation Sequence Diagram*

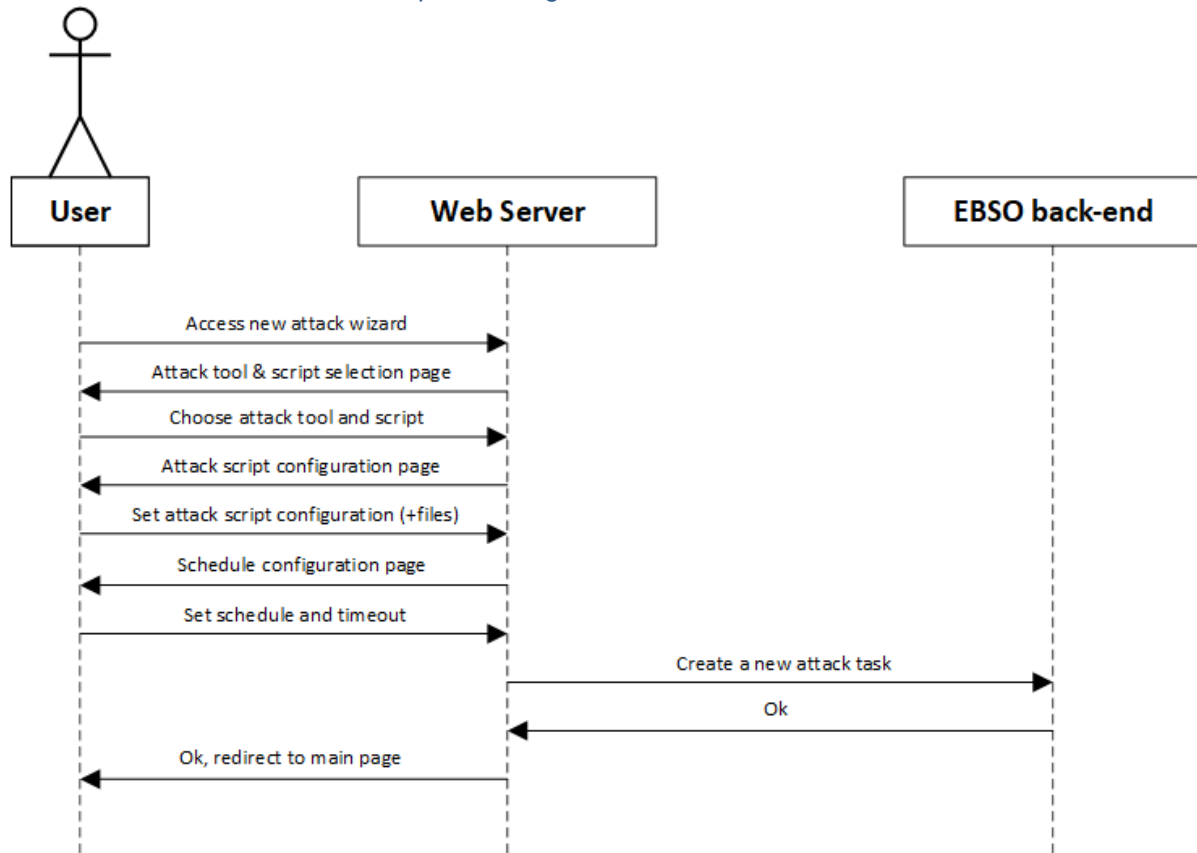


Figure 55. AS – Attack Creation Sequence Diagram

### 6.2.19 Attack Simulator – Attack Modification

Use Case Name	Attack Simulator – Attack Modification
UC id	UC.11.005
Description	Provides the ability to modify the settings of an existing attack task or cancel its scheduled executions.
Actors	Trainer Trainee
Objects	User Web Server EBSO back-end
Basic flow	Using a link on the attack details page, a user can request to modify the schedule parameters of an existing attack task, scheduled for future execution. The user is shown a schedule configuration page, where they can modify the number of future attack executions, the interval between executions, and the timeout value. The task can also be disabled, preventing future executions. After the user confirms the settings, the web server sends the requested values to the EBSO back-end, which updates the task schedule. Finally, a confirmation message is displayed to the user and they are redirected back to the attack details page.
Preconditions	User must be logged in and have a valid session token identifying them with the AS web server. There needs to be at least one task scheduled for execution, available to the current user.
Postconditions	The attack task is modified according to the configuration requested.
Dependencies	UC.11.003

Table 74. UC.11.005 – Attack Simulator – Attack Modification

#### Attack Simulator – Attack Modification Sequence Diagram

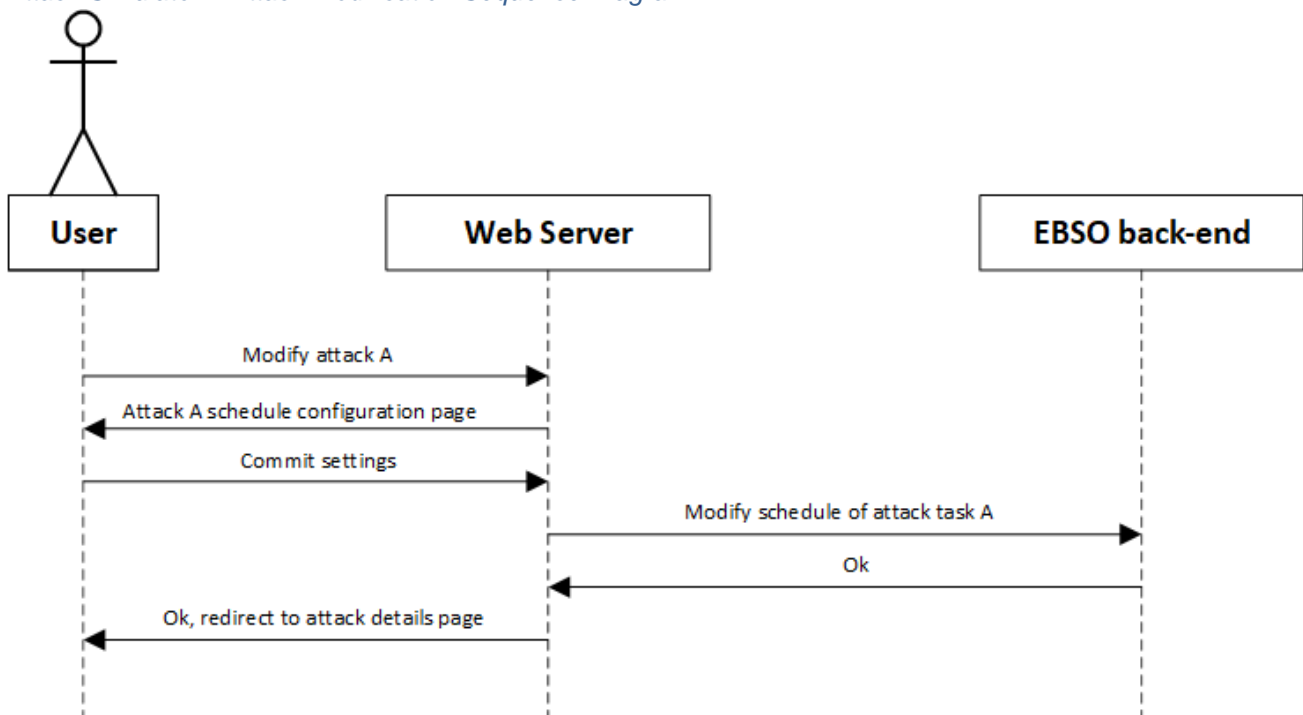


Figure 56. AS – Attack Modification Sequence Diagram

### 6.2.20 Centralized Logging Component Dashboard

Use Case Name	Centralized Logging Component Dashboard
UC id	UC.15.001
Description	Provides a user interface to view logs, saved in the CLC
Actors	Trainer Platform operator
Objects	User CLC Web UI Server CLC back-end
Basic flow	After a user accesses the web UI, the web server requests the log data from the CLC back-end service. It returns the logs and a web page with the data is shown to the user. The returned page enables an overview of the logs and their filtering. If a user chooses a filter or inputs a filter query, the query gets forwarded to the back-end, the results are returned to the server, and shown to the user in their web browser.
Preconditions	
Postconditions	User is shown the appropriate data
Dependencies	

Table 75. UC.15.001 – Centralized Logging Component Dashboard

#### Centralized Logging Component Dashboard Sequence Diagram

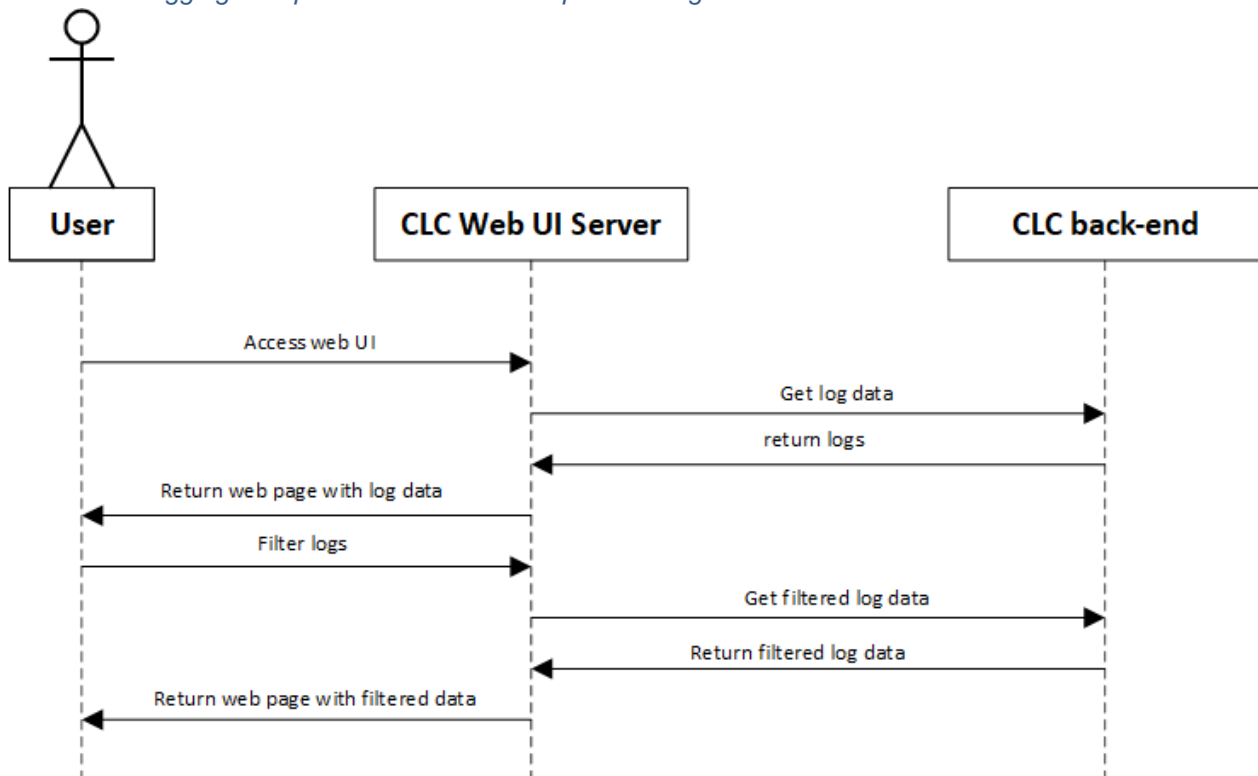


Figure 57. CLC Dashboard Sequence Diagram

### 6.2.21 Economic Risk Evaluator – getting risk information

Use Case Name	Economic Risk Evaluator – getting risk information
UC id	UC.07.001
Description	This use case covers the retrieval of information about the current cyber risk to which the infrastructure is exposed, the possible mitigations to be applied to diminish the risk, the history of the risk evolution over the time, and consulting a specific cyber risk report generated in the past
Actors	Trainer Trainee
Objects	User ERE UI ERE back-end
Basic flow	This use case is basically about requesting and obtaining information from the internal database of the ERE. This covers the current risk, the possible mitigations, the history and picking old reports to further check details.
Preconditions	The ERE must be part of the scenario. This can happen in the Medium and Advanced offering levels; for lower levels this asset is not offered as part of the training exercise. In addition, the ERE must be configured by means of the contextualization The user must initiate the NoVNC connection with the VM hosting the ERE
Postconditions	
Dependencies	

Table 76. UC.07.001 – Economic Risk Evaluator – getting risk information

#### *Economic Risk Evaluator – getting risk information sequence diagram*

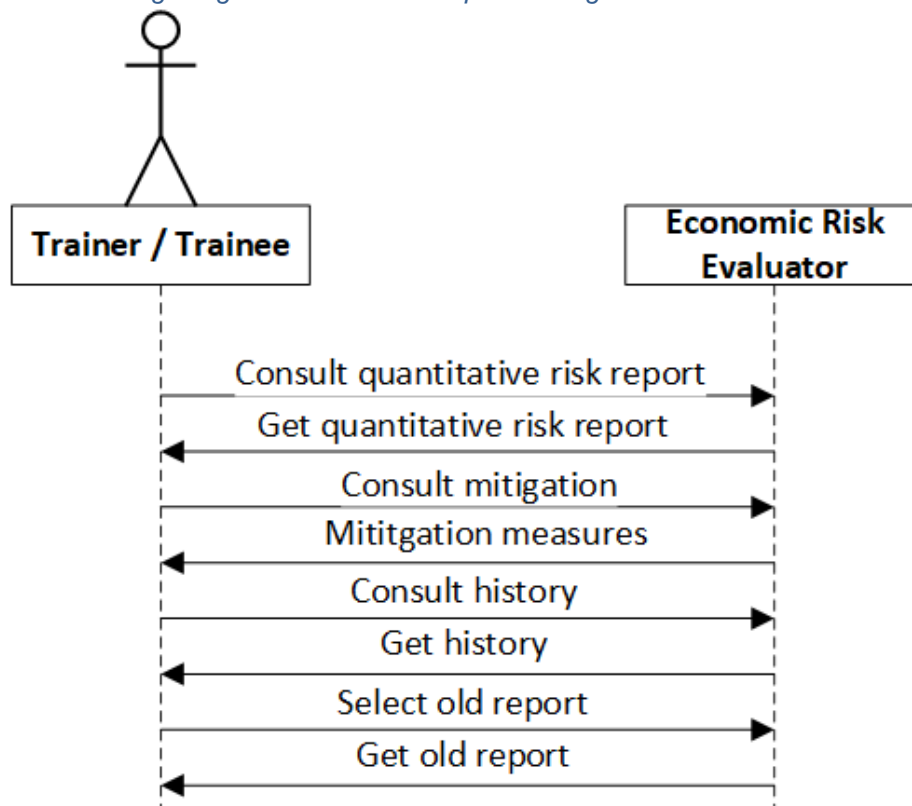


Figure 58. Economic Risk Evaluator – getting risk information sequence diagram



## 6.2.22 Economic Risk Evaluator – consulting target configuration

Use Case Name	Economic Risk Evaluator – consulting target configuration
UC id	UC.07.002
Description	This use case covers the retrieval of the information related to the targets <sup>66</sup> present in the simulated scenario. The user can check the list of targets with the basic information related to each of them, and can also go to the detail of each target by clicking on a specific one
Actors	Trainer Trainee
Objects	User ERE UI ERE back-end
Basic flow	This use case is basically about requesting and obtaining information from the internal database of the ERE. This covers the basic information for the list of existing targets and the detail per target.
Preconditions	The ERE must be part of the scenario. This can happen in the Medium and Advanced offering levels; for lower levels this asset is not offered as part of the training exercise. In addition, the ERE must be configured by means of the contextualization. The user must initiate the NoVNC connection with the VM hosting the ERE.
Postconditions	
Dependencies	

Table 77. UC.07.002 – Economic Risk Evaluator – consulting target configuration

### Economic Risk Evaluator – consulting target configuration sequence diagram

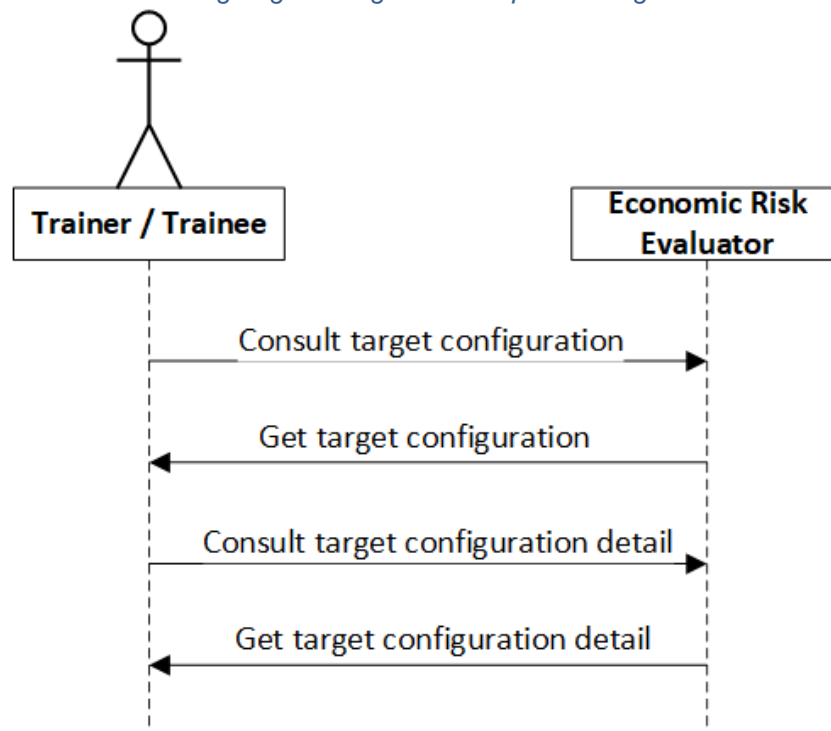


Figure 59. Economic Risk Evaluator – consulting target configuration sequence diagram

<sup>66</sup> Targets mean the different relevant individual elements of the simulated infrastructure, considering both machines and applications. Machines are identified by means of an IP address and applications are identified by IP and Port

### 6.2.23 Economic Risk Evaluator – consulting models selected

Use Case Name	Economic Risk Evaluator – consulting models selected
UC id	UC.07.003
Description	This use case is about consulting which ones among the available risk models are active for the calculation of the cyber risk exposure in the context of an ongoing exercise
Actors	Trainee Trainer
Objects	User ERE UI ERE Back-end
Basic flow	This use case is basically about requesting and obtaining information from the internal database of the ERE. This covers the list of cyber risk models that are active for the running exercise.
Preconditions	The ERE must be part of the scenario. This can happen in the Medium and Advanced offering levels; for lower levels this asset is not offered as part of the training exercise. In addition, the ERE must be configured by means of the contextualization. The user must initiate the NoVNC connection with the VM hosting the ERE.
Postconditions	
Dependencies	

Table 78. UC.07.003 – Economic Risk Evaluator – consulting models selected

#### *Economic Risk Evaluator – consulting models selected sequence diagram*

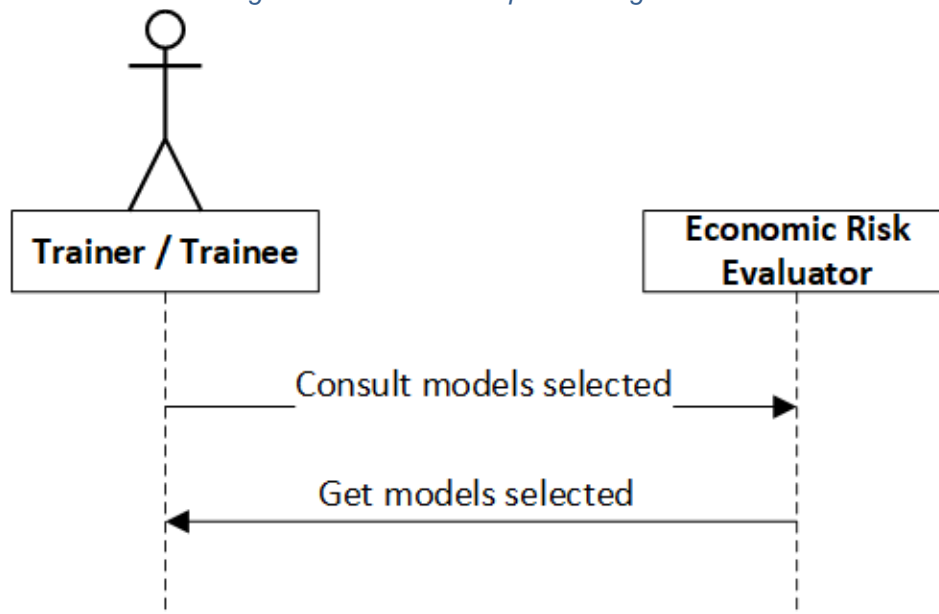


Figure 60. Economic Risk Evaluator – consulting models selected sequence diagram

#### 6.2.24 Anomaly Detection Reasoner – monitoring the simulated infrastructure

Use Case Name	Anomaly Detection Reasoner – monitoring the simulated infrastructure
UC id	UC.10.001
Description	This use case covers the monitoring of the simulated infrastructure and following up the events, alarms and some trend statistics that can be seen in the Anomaly Detection Reasoner landing page (UI.10.001). This is the basic flow the user can follow within the ADR
Actors	Trainees Trainers
Objects	User SIM UI SIM back-end ADR UI ADR back-end
Basic flow	The user establishes a NoVNC connection with the ADR from the SIM and this leads him to the landing page of the ADR. From this landing page, the different UIs documented in the section 6.1 (in UI.10) can be directly accessed, as it also shown in Figure 37. The information is obtained by interacting with the ADR back-end by means of the user interface.
Preconditions	The scenario must have been launched and the exercise must be running, with the ADR having been deployed and connected to the Monitoring Sensors and the ADR-Agent
Postconditions	
Dependencies	UC.04.002

Table 79. UC.10.001 – Anomaly Detection Reasoner – monitoring the simulated infrastructure

#### Anomaly Detection Reasoner – monitoring the simulated infrastructure sequence diagram

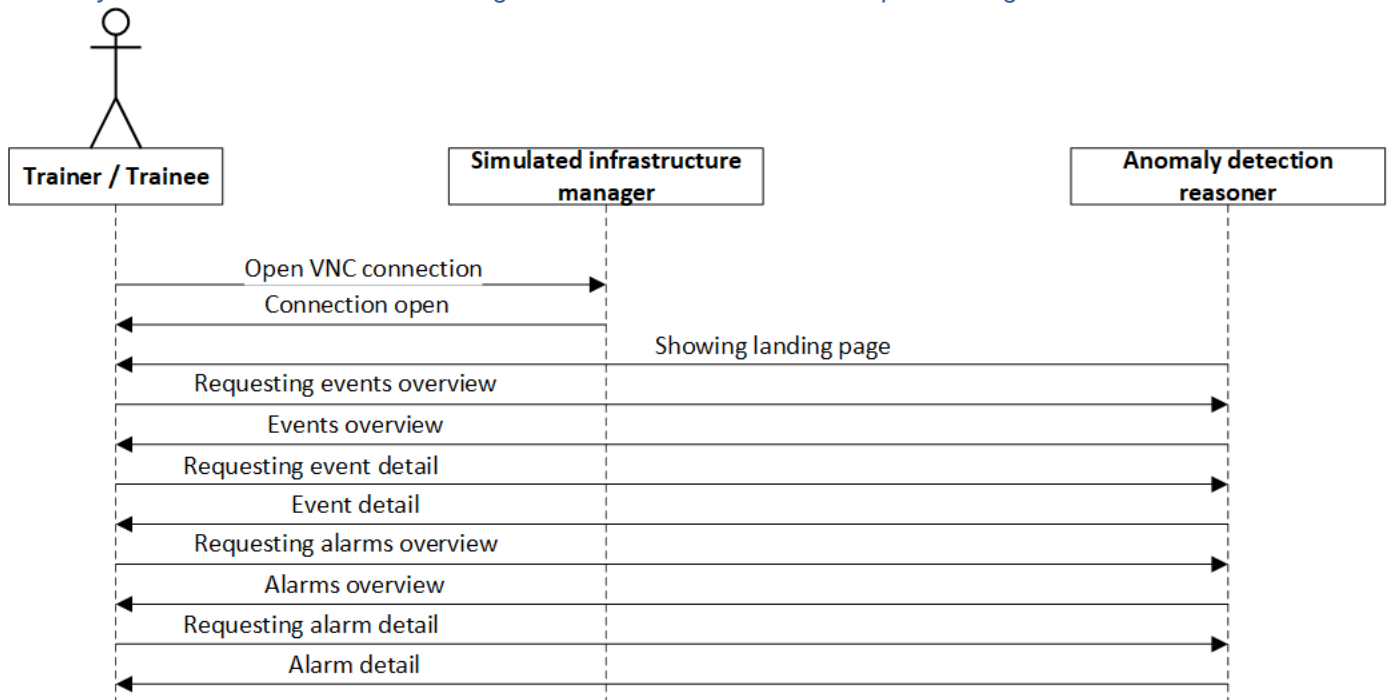


Figure 61. Anomaly Detection Reasoner – simulating the monitored infrastructure sequence diagram

### 6.2.25 Countermeasures Simulator – applying mitigation

Use Case Name	Countermeasures Simulator – applying mitigation
UC id	UC.12.001
Description	This use case covers the identification of a mitigation measure by the Countermeasures Simulator, following an incident in the simulated scenario, and the application of the measure, either directly or executing some existing script in the CS back-end.
Actors	Trainee
Objects	User Simulated Infrastructure Anomaly Detection Reasoner Message Broker Countermeasures Simulator UI Countermeasures Simulator back-end
Basic flow	When an anomaly takes place in the simulated infrastructure, the logs will contain relevant information that will be processed by the ADR-Agent and the ADR-itself to raise a meaningful alarm, containing (at minimum) basic information about the incident happening and the IP involved. This information is sent to the CS via the MB, and the CS internal intelligence will identify one or more candidate mitigations to be applied. Such mitigations are shown to the user by means of the CS GUI. Then the user selects a mitigation. If there are pre-defined scripts to run the mitigation, the user will take advantage of them and will execute the mitigation via the CS passing the needed parameters to the script. If there is not any available script, the user himself will run the mitigation in his own. This is conveniently reflected in the sequence diagram of this use case. More information about the Countermeasures Simulator can be found in sections 3.2.13 and 5.13.
Preconditions	The exercise must have started, this is the simulation of the target infrastructure must be running. As part of the scenario instantiation, the Monitoring Sensors, the ADR-Agent and the ADR must be up and running. The MB must be up and running Some incident must have taken place. The CS must have been deployed and passed all the contextualization information needed for its internal functioning
Postconditions	The correct application of the mitigation measure must affect the feedback sent by the ADR and the ERE, as main elements used to follow-up the execution of the exercise. Additionally, the application of the measure must impact the evaluation of the user performance calculated by the PE.
Dependencies	UC.04.002, UC.10.001

Table 80. UC.12.001 – Countermeasures Simulator – applying mitigation

*Countermeasures Simulator – applying mitigation sequence diagram*

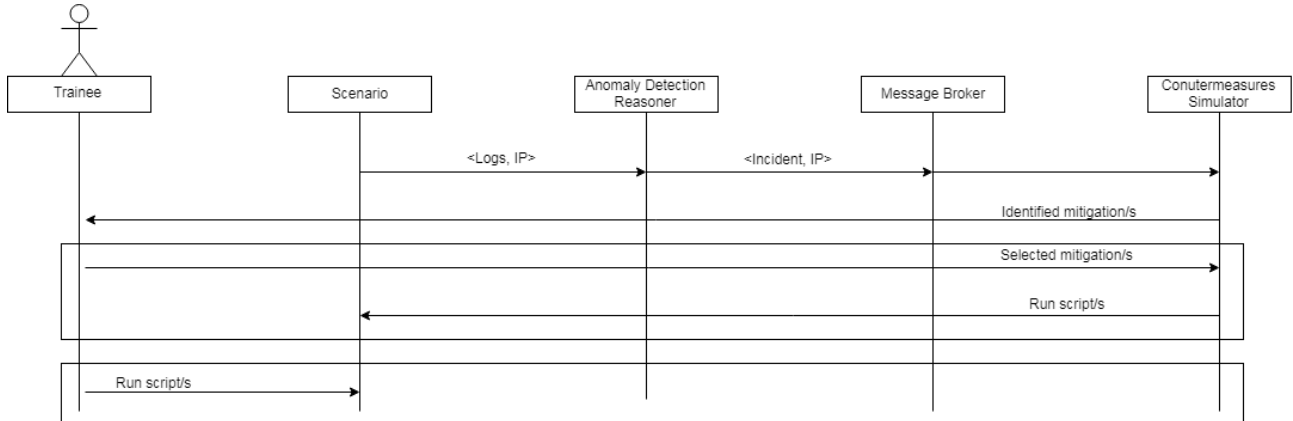


Figure 62. Countermeasures Simulator – applying mitigation sequence diagram

## 7 Impact of the business requirements on the design

### 7.1 CYBERWISER.eu versions and components

Within T2.1, CYBERWISER.eu has been presented within four Platform offers, for different customers and different markets:

- Primer,
- Basic,
- Intermediate,
- Advanced.

From the Primer to the Advanced version, the complexity of the platform increases: the Advanced version corresponds to the full CYBERWISER.eu features and components highlighted in the next section.

The other versions of the platform will be deployed with a limited number of components in order to support different kind of services. In Table 81 all main CYBERWISER.eu components have been listed in connection with the platform version where they will be deployed and integrated.

As can be seen, the Primer version does not encompass cyber-range capabilities. From the Basic version, the cyber-range features are added and become more and more complete up to the Advanced version.

It is important to highlight also that several components will offer different features depending on the platform version. This is summarized in Table 81.

ASSET PLATFORM version	PRIMER	BASIC	INTERMEDIATE	ADVANCED
Web portal	Available	Available	Available	Available
Cross-Learning facilities	<p>Introductory course about cybersecurity and risk analysis in general.</p> <p>Introductory course about risk assessment.</p> <p>Simple high score competitions in quiz mode at individual level.</p> <p>The CYBERWISER.eu platform will release a certificate to all successful participants.</p> <p>Teasers of the higher level from Basic up will be provided to all Primer users in the form of videos, brochures etc.</p>	<p>Training materials, basic level.</p> <p>Static exercises (Trainees' activity guided by simple training material -slide decks, guides, reports etc.- accessible through Cross-Learning Facilities).</p> <p>Monitoring of student's course progress and grading.</p>	<p>Training materials, intermediate level.</p> <p>Static exercises (Trainees' activity guided by simple training material -slide decks, guides, reports etc.- accessible through Cross-Learning Facilities).</p> <p>Monitoring of student's course progress and grading.</p>	<p>Training materials, advanced level.</p> <p>Dynamic exercises (Trainees' activity guided by training material enriched with interactive tools, accessible through Cross-Learning Facilities).</p> <p>Monitoring of student's course progress and grading.</p>
Training Manager	N/A	Prepared training scenario composed by up to 10 scenario elements (Virtual Machines / Virtual Networks)	<p>Prepared training scenario composed by up to 50 scenario elements (Virtual Machines/Virtual Networks).</p> <p>Possibility to create new training scenarios with up to 50 elements.</p> <p>Competitive training scenarios (blue vs. red team).</p>	<p>Prepared training scenario instances composed by more than 50 scenario elements (Virtual Machines/Virtual Networks).</p> <p>Possibility to create new training scenarios with up to 500 elements.</p> <p>Possibility to request new elements to be added to the Digital Library, if not already present.</p>
Performance evaluator	N/A	N/A	Evaluations based on a limited number of parameters.	Advanced evaluation capabilities

ASSET PLATFORM version /	PRIMER	BASIC	INTERMEDIATE	ADVANCED
Digital Library	N/A	Limited training scenario elements choice	Full training scenario elements choice	Full training scenario elements choice
Economic Risk Evaluator	N/A	N/A	Basic suite of economic risk evaluation models and algorithms.	Advanced suite of economic risk evaluation models and algorithms.
Simulated Infrastructure Manager	N/A	Simple training scenarios. Low cyber-range capacity requirements	Medium complexity training scenarios. Medium cyber-range capacity requirements	Complex training scenarios. High cyber-range capacity requirements
Monitoring sensors	N/A	Automatically deployed in the training scenarios instances	Automatically deployed in the training scenarios instances	Automatically deployed in the training scenarios instances
Attack Simulator	N/A	Basic suite of available attacks	Advanced suite of available attacks	Advanced suite of available attacks
Countermeasures Simulator	N/A	N/A	Intermediate suite of available mitigations	Advanced suite of available mitigations
Anomaly Detection Reasoner	N/A	Available	Available	Available
Vulnerability Assessment tools	N/A	N/A	Intermediate version of vulnerability assessment tools	Advanced version of vulnerability assessment tools.
Centralized logging component	N/A	Available	Available	Available

Table 81. CYBERWISER.eu versions and components

Each of the CYBERWISER.eu version will hence encompass a different deployment diagram with respect to Figure 4 (corresponding to the full CYBERWISER.eu, **Advanced** version)



## 7.2 Primer Version Deployment Diagram

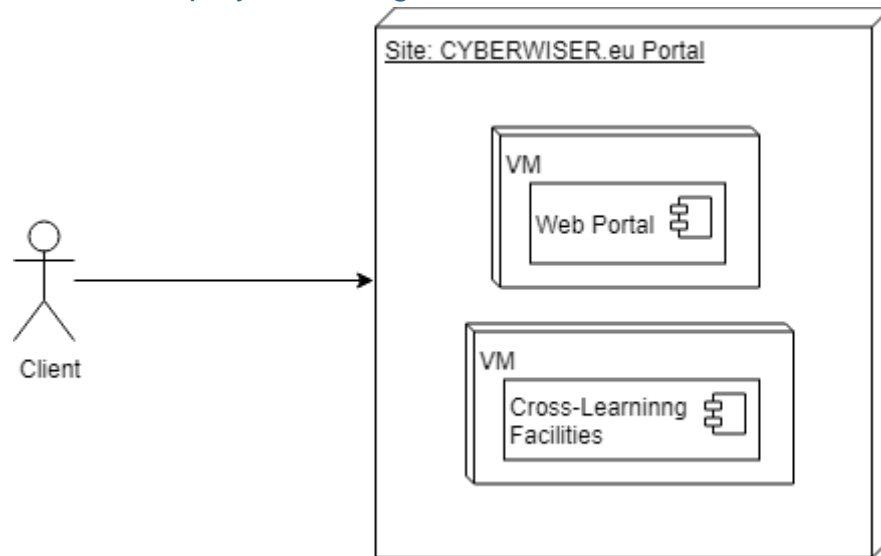


Figure 63. CYBERWISER.eu Primer version deployment diagram

The Primer version considers a very simple deployment (see Figure 63) with two VMs, one of them hosting the Web Portal and the other assigned to the Cross-Learning Facilities that become the only component available in this offering level.

### 7.3 Basic Version deployment diagram

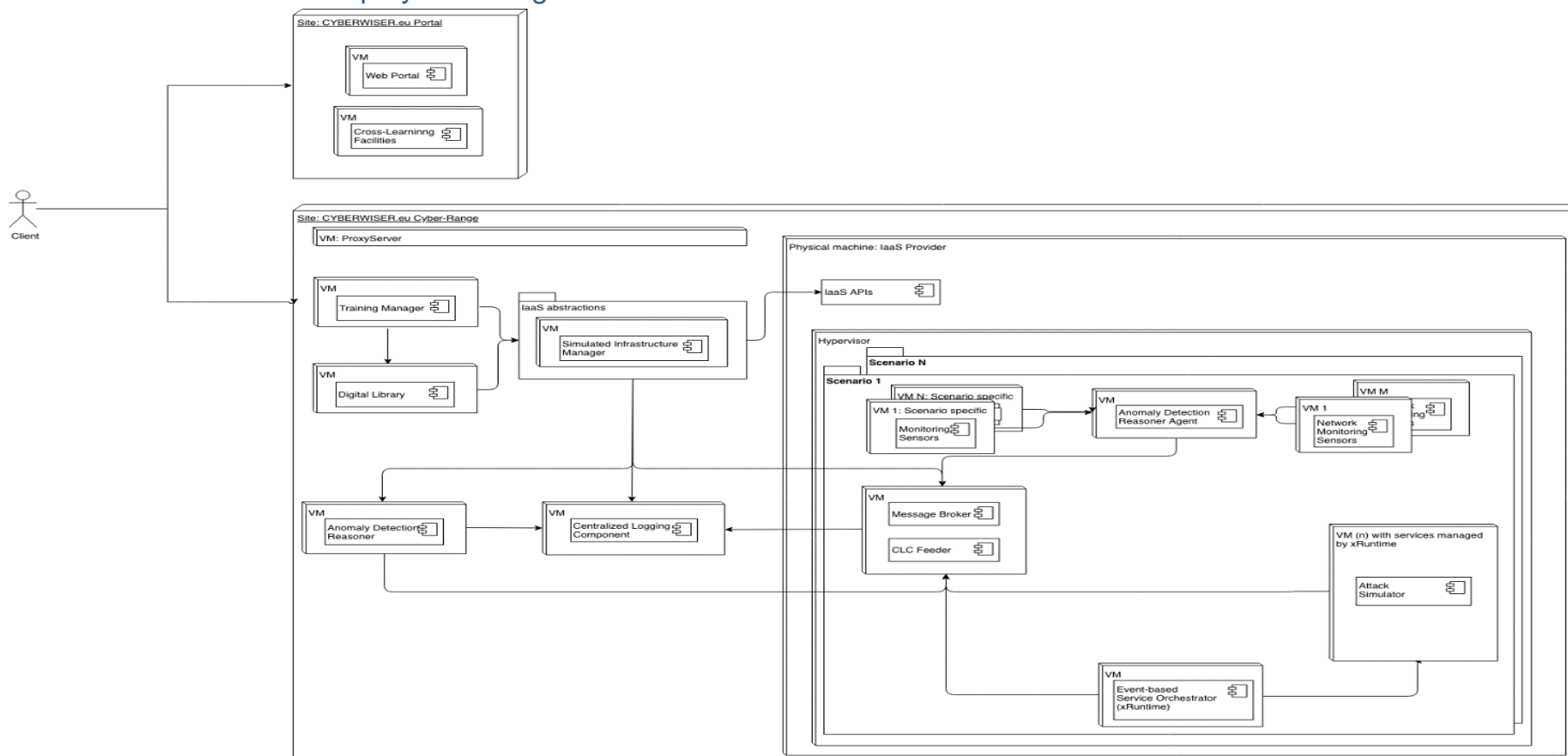


Figure 64. CYBERWISER.eu Basic version deployment diagram

At the Basic level the first cyber range capabilities are available. In particular, the IaaS infrastructure is available for the deployment of scenarios on which the training exercises are generated. This deployment is managed by the SIM using inputs coming from the TM and considering in such deployment the elements chosen from the DL. In addition, the ADR and the CLC, which operate outside the scenario, are deployed as well. All these components stay available from this level onwards, being strengthened and supplemented in subsequent levels.

## 7.4 Intermediate Version Deployment Diagram

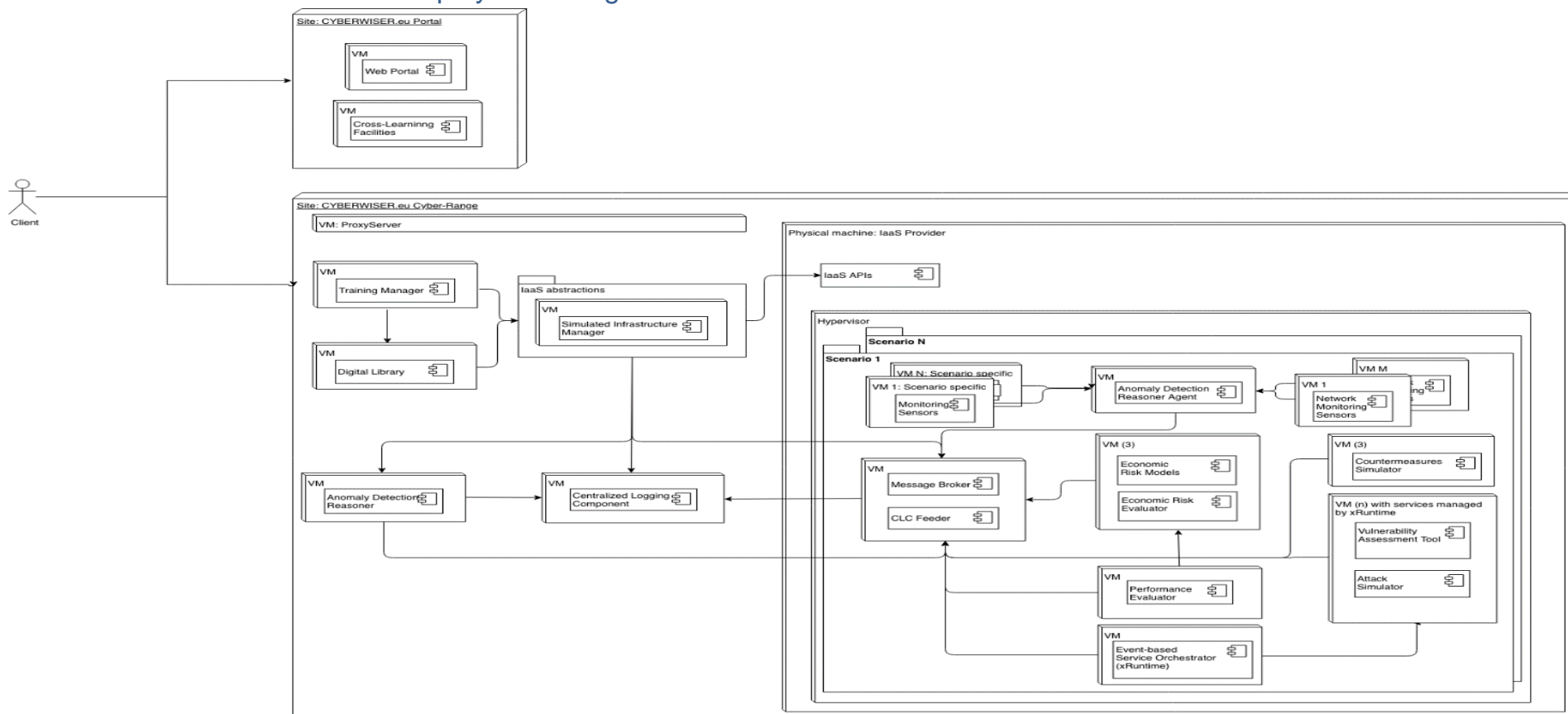


Figure 65. CYBERWISER.eu Intermediate Version Deployment Diagram

As can be seen in the previous figure, the Intermediate version encompasses all main CYBERWISER.eu components (except for the Pluggable Deployment Manager). Differences with the Advanced version are related to the offered features and capabilities of the deployed components (for example, simpler training scenarios will be supported, or limited number of attack scripts will be provided).

## 8 Final requirements traceability

Within Section 2.4, the need of full traceability between CYBERWISER.eu requirements and the design has been highlighted as a necessary step to ensure that all requirements will be verifiable after the development phase. In this Section, Requirements Realization Matrices are provided, tracing the requirements from D2.2 and the design of the Advanced version of the platform proposed in this deliverable. The purpose is to show that the current design satisfies the requirements from D2.2.

High-level requirements (for example, FUNC-1) are satisfied by the whole CYBERWISER.eu platform (Advanced version).

Requirements traceability will be proposed by dividing the requirements following their Type (Functional, Platform, Legal, Security, Usability and Performance), from Table 82 to Table 87.

ID	Type	Traced Asset	Description
FUNC-1	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST support training through online cybersecurity courses.
FUNC-2	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST provide simulated environments for cybersecurity exercises.
FUNC-3	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST allow trainees to perform the role of system defenders (blue team).
FUNC-4	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST allow trainees to perform the role of attackers (red team).
FUNC-5	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST support multiple simultaneous instantiations of training scenarios.
FUNC-6	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST support various training modes with respect to the level of collaboration (non-collaborative meaning individual trainees only, collaborative meaning also teams) and competition (non-competitive means against the CYBERWISER.eu Platform, competitive meaning competition between trainees). All combinations must be covered.
FUNC-7	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST support training of individual trainees.
FUNC-8	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST support collaborative exercises, e.g. exercises that involve more than one individual (a team), where a team strives towards a common objective in the training scenario.
FUNC-9	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST support non-competitive training scenarios. In non-competitive training, an individual/team trains against the CYBERWISER.eu Platform.

ID	Type	Traced Asset	Description
FUNC-10	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST support competitive training scenarios between teams and/or individuals competing on the opposing sides, e.g. attackers against defenders (blue team vs. red team).
FUNC-11	Functional	CYBERWISER.eu Platform	Elements of exercise configuration MUST be dynamically reconfigurable during the execution, in order to support on-the-fly adjustments of scope and difficulty.
FUNC-12	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform COULD allow trainees to suspend ongoing training scenarios and resume them later on.
FUNC-13	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform COULD allow trainees to record the GUI or CLI of certain virtual machines, assigned to them, and later replay the recording of an exercise.
FUNC-14	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST differentiate between various end-user types (for instance trainer, trainee), by assigning each user an appropriate role.
FUNC-15	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform COULD allow streaming of on-going training sessions (or selected fractions of them).
FUNC-16	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform SHOULD allow to create nicknames for the trainees.
FUNC-17	Functional	CYBERWISER.eu Platform	With the nickname, The CYBERWISER.eu Platform SHOULD show a performance rating (e.g. number of participated training sessions, etc..).
FUNC-18	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST integrate learning materials supporting entry-level training.
FUNC-19	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform SHOULD allow a customised access to the results of an organization's training exercises, providing for each exercise a summary report of the main simulation outcomes (for instance best defending strategy, estimated economic impact of a specific scenario etc.).
FUNC-20	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST provide a summary report for all conducted exercises of all trainees, consisting of the main simulation outcomes (best defending strategy, estimated economic impact of a specific scenario etc.). In this case the summary must be anonymised and filtered considering sensitive information.

ID	Type	Traced Asset	Description
FUNC-21	Functional	CYBERWISER.eu Platform	All VMs in the cyber-range exercise environment MUST be accessible through the CYBERWISER.eu Simulated Infrastructure Manager graphical user interface.
FUNC-22	Functional	CYBERWISER.eu Platform	During the cyber-range exercise, red team trainees MUST have access to offensive security & penetration testing software from the exercise VMs assigned to them.
FUNC-23	Functional	Cross-Learning Facilities	The Cross-Learning Facilities MUST include Workspace Areas to enable users to exchange messages, manage documents, directly access "external" services as training and eLearning sections.
FUNC-24	Functional	Cross-Learning Facilities	Each Workspace Area MUST have at least one administrator.
FUNC-25	Functional	Cross-Learning Facilities	It MUST be possible to associate at least one cyber-range training scenario with a particular online course.
FUNC-26	Functional	Cross-Learning Facilities	An online course COULD be composed by series of slides enriched by audio comment, possibly divided in modules accessible according to a pre-ordered precedence.
FUNC-27	Functional	Cross-Learning Facilities	An online course MUST allow the exchange of files between trainees and trainers.
FUNC-28	Functional	Cross-Learning Facilities	An online course MUST trace the point of the course actually reached by a student following the course.
FUNC-29	Functional	Cross-Learning Facilities	An online course MUST allow to the trainer, in whatever point of the course, to verify the state of learning of the trainee, finally automatically and/or manually grading the trainee, possibly reiterating the student until he/she reaches a required level.
FUNC-30	Functional	Cross-Learning Facilities	An online course MUST provide a final certificate to the students who completed the entire course.
FUNC-31	Functional	Cross-Learning Facilities	The Workspace Area MUST provide the capability to create/delete folders, upload/download files respecting the user profile.
FUNC-32	Functional	Scenario Designer	The Scenario Designer MUST support the design of the ICT topology of the simulated environments leveraging the training scenario requests.
FUNC-33	Functional	Scenario Designer	The Scenario Designer MUST have an ICT Topology Design Dashboard where the user will be able to draw the ICT topology of a training scenario



ID	Type	Traced Asset	Description
			request. The dashboard will present: <ul style="list-style-type: none"> <li>· Network diagram area,</li> <li>· Network taxonomy objects toolbar,</li> <li>· Wizard object characterization tabbed box.</li> </ul>
FUNC-34	Functional	Scenario Designer	The Scenario Designer MUST support the design of the application topology of the simulated environments leveraging the training scenario requests.
FUNC-35	Functional	Scenario Designer	The Scenario Designer MUST have an Application Design Dashboard where the user will be able to draw the application topology of a training scenario request. The dashboard will present: <ul style="list-style-type: none"> <li>· Application diagram area,</li> <li>· Application taxonomy objects toolbar,</li> <li>· Wizard object characterization tabbed box.</li> </ul>
FUNC-36	Functional	Scenario Designer	The Scenario Designer MUST support the definition of scenario events and events triggers for the simulated environments leveraging the training scenario requests.
FUNC-37	Functional	Scenario Designer	The Scenario Designer MUST support the definition of the monetary value of identified elements in the simulated environments leveraging the training scenario requests.
FUNC-38	Functional	Scenario Designer	The Scenario Designer MUST support the definition of the instructions for the trainees involved in the training scenario via a training design dashboard.
FUNC-39	Functional	Scenario Designer	The Scenario Designer MUST have a Training Dashboard where the user will be able to design and collect the Training details (instructions for the trainees, pedagogical objectives, evaluation criteria) of a training scenario request. The dashboard will present: <ul style="list-style-type: none"> <li>· Training diagram area,</li> <li>· Training taxonomy object toolbar,</li> <li>· Wizard object characterization tabbed box.</li> </ul>
FUNC-40	Functional	Scenario Designer	The Scenario Designer MUST provide a graphical user interface for definition, configuration and composition of training scenario requests as part of Activities defined in the Training Manager.
FUNC-41	Functional	Scenario Designer	The Scenario Designer MUST support training scenario requests versioning.

ID	Type	Traced Asset	Description
FUNC-42	Functional	Scenario Designer	The Scenario Designer MUST allow to define training scenarios with limited time duration.
FUNC-43	Functional	Scenario Designer	The Scenario Designer MUST be able to leverage on the catalogue of pre-defined set of virtual/physical elements provided by the Digital Library.
FUNC-44	Functional	Scenario Designer	The Scenario Designer MUST be able to leverage on the catalogue of pre-defined set of training scenarios provided by the Digital Library.
FUNC-45	Functional	Scenario Designer	The Scenario Designer SHOULD support the design of the business topology of the simulated environments leveraging the training scenario requests.
FUNC-46	Functional	Scenario Designer	<p>The Scenario Designer SHOULD have a Business Design Dashboard where the user can draw the underlying business processes of a training scenario request.</p> <p>This dashboard will present:</p> <ul style="list-style-type: none"> <li>· Business Processes diagram area,</li> <li>· Business Processes taxonomy objects toolbar,</li> <li>· Wizard object characterization tabbed box.</li> </ul>
FUNC-47	Functional	Scenario Designer	The Scenario Designer MUST be able to design and add training scenario templates in the pre-defined set of training scenario templates in the Digital Library.
FUNC-48	Functional	Scenario Designer	The Scenario Designer SHOULD support the definition of a timeline of events and events triggers for the simulated environments leveraging the training scenario requests.
FUNC-49	Functional	Scenario Designer	<p>The Scenario Designer MUST have a dashboard where relevant information related to the training scenario requests will be summarized:</p> <ul style="list-style-type: none"> <li>· Unique ID,</li> <li>· Creation date/time,</li> <li>· Status (Design in Progress, Waiting, Validated, Instantiated),</li> <li>· Description and/or motivation.</li> </ul>
FUNC-50	Functional	Scenario Designer	<p>Training scenario requests MUST be pre-validated before they are sent to the Simulated Infrastructure Manager. Pre-validation will consist of a consistency check on the minimal set of information that the request must encompass:</p> <ul style="list-style-type: none"> <li>· ICT Topology and its assets' attributes,</li> <li>· Application Topology and its assets' attributes,</li> </ul>

ID	Type	Traced Asset	Description
			<ul style="list-style-type: none"> <li>· Business Topology and its assets' attributes,</li> <li>· Description and/or motivation.</li> </ul>
FUNC-51	Functional	Scenario Designer	The Scenario Designer MUST allow the user to define a moment in the future at which the training scenario instantiation will automatically start as part of a defined Activity.
FUNC-52	Functional	Scenario Designer	The ICT Topology Design and the Application Topology Design dashboards SHOULD enable the User to associate the elements with hardware/software assets that the IaaS platform is able to provide in Hybrid Virtual Environment (e.g. physical Space Assets modems within a virtual environment at IP network level).
FUNC-53	Functional	Scenario Designer	The Scenario Designer MUST, after pre-validation, send the training scenario requests to the Simulated Infrastructure Manager.
FUNC-54	Functional	Scenario Designer	It MUST be possible to limit the number of training scenario elements in the ICT Topology Design Dashboard and in the Application Dashboard.
FUNC-55	Functional	Scenario Designer	It MUST be possible to keep track of the number of training scenario elements in the ICT Topology Design Dashboard and in the Application Dashboard.
FUNC-56	Functional	Scenario Designer	It MUST be possible to mark a training scenario as editable or non-editable (locked).
FUNC-57	Functional	Performance Evaluator	The input to the trainee's performance evaluation MUST be based on the cyber-range exercises carried out by the trainee.
FUNC-58	Functional	Performance Evaluator	The Performance Evaluator MUST be able to timely evaluate trainees' progress, based on the relevant indicators for performance evaluation criteria which are monitored and processed with a frequency that allows observability.
FUNC-59	Functional	Performance Evaluator	The Performance Evaluator MUST offer an interface to follow the evolution of the user performance over the time, with clear and concise information to be easily understood by both trainers and trainees.
FUNC-60	Functional	Performance Evaluator	At the end of the exercise, the Performance Evaluator MUST issue a final evaluation report assigned to the trainee with the most relevant information about his performance during the exercise.

ID	Type	Traced Asset	Description
FUNC-61	Functional	Performance Evaluator	Performance evaluation MUST be based on clear criteria for evaluating the trainee.
FUNC-62	Functional	Performance Evaluator	The Performance Evaluator MUST admit flexible configuration from the white team.
FUNC-63	Functional	Digital Library	The Digital Library MUST be able to provide to the Scenario Designer a catalogue of pre-defined set of virtual/physical elements available in the cyber-range infrastructure (to be potentially used during the definition of the ICT topology or the application topology).
FUNC-64	Functional	Digital Library	The Digital Library MUST be able to provide to the Scenario Designer a catalogue of pre-defined set of training scenarios.
FUNC-65	Functional	Digital Library	The catalogue of pre-defined training scenarios MUST include scenarios focused on early prevention of cyber-incidents, e.g. exercises for proactive defence against specific attacks.
FUNC-66	Functional	Digital Library	The Digital Library MUST be able to store in the catalogue of pre-defined training scenarios any training scenarios created by the Scenario Designer.
FUNC-67	Functional	Digital Library	The Digital Library MUST be able to store new metadata of physical elements in the catalogue of pre-defined set of physical elements available for the cyber-range infrastructure (to be potentially used during the definition of the ICT topology or the application topology).
FUNC-68	Functional	Digital Library	The Digital Library MUST be able to store new metadata of virtual elements in the catalogue of pre-defined set of virtual elements available for the cyber-range infrastructure (to be potentially used during the definition of the ICT topology or the application topology).
FUNC-69	Functional	Training Manager	When a trainee concludes the cyber-range exercise, the Training Manager MUST show evaluation criteria and achieved scores to both the trainee and the trainer.
FUNC-70	Functional	Training Manager	The Training Manager MUST encompass an activity area where the information regarding each activity will be aggregated.
FUNC-71	Functional	Training Manager	The Training Manager MUST offer a collaborative space where users may find and/or exchange information about training scenarios and activities.

ID	Type	Traced Asset	Description
FUNC-72	Functional	Training Manager	The Training Manager MUST allow the users to create/edit/delete activities.
FUNC-73	Functional	Training Manager	The Training Manager MUST allow the users to create/edit/delete training scenarios within an activity.
FUNC-74	Functional	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST provide the capability to simulate the ICT infrastructure (or identified sections of it) of an end-user leveraging on a set of virtual and/or physical resources (the cyber-range infrastructure).
FUNC-75	Functional	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST provide the capability to instantiate/deinstantiate/delete, in the cyber-range infrastructure, the virtual environment templates leveraging the training scenario requests.
FUNC-76	Functional	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST provide the capability to validate the simulation environment templates leveraging the training scenario requests.
FUNC-77	Functional	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST provide the capability to reject the virtual environment templates leveraging the training scenario requests. Rejected virtual environment templates will be notified to the requesters.
FUNC-78	Functional	Simulated Infrastructure Manager	Authorized users MUST be able to command the Simulated Infrastructure Manager to instantiate/deinstantiate/delete virtual environment templates leveraging the training scenario requests.
FUNC-79	Functional	Simulated Infrastructure Manager	Authorized users MUST be able to start a training scenario instance instantiated in the cyber-range infrastructure.
FUNC-80	Functional	Simulated Infrastructure Manager	Authorized users MUST be able to stop a training scenario instance instantiated in the cyber-range infrastructure.
FUNC-81	Functional	Simulated Infrastructure Manager	Authorized users MUST be able to reboot (start again the training scenario instance from its initial state) a training scenario instance instantiated in the cyber-range infrastructure.
FUNC-82	Functional	Simulated Infrastructure Manager	Authorized users MUST be able to open remote sessions (if available) with any involved Virtual Machine in the active (Started) instantiated training scenario instance.

ID	Type	Traced Asset	Description
FUNC-83	Functional	Simulated Infrastructure Manager	Authorized users MUST be able to manually command training scenario events and event triggers, during training scenario instance execution.
FUNC-84	Functional	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST be able to provide the status and availability of the resources of the cyber-range infrastructure.
FUNC-85	Functional	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST be able to provide status (e.g. Running, Stopped, Ready for instantiation, etc.) of the instantiated virtual environment templates and all involved virtual machines.
FUNC-86	Functional	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST be able to retrieve information about the training exercise execution, e.g. trainee's actions and events in the training exercises.
FUNC-87	Functional	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST provide a graphical user interface leveraging the validation of the virtual environment templates, the instantiation of the virtual environment templates and the monitoring and command of the execution of the training scenarios in the cyber-range infrastructure.
FUNC-88	Functional	Simulated Infrastructure Manager	The Simulated Infrastructure Manager graphical user interface MUST allow trainers to have insight into trainees' progresses on an active training scenario.
FUNC-89	Functional	Simulated Infrastructure Manager	Authorized users MUST be able to check the status and availability of the resources of the cyber-range infrastructure.
FUNC-90	Functional	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST be able to handle training scenarios with limited time duration.
FUNC-91	Functional	Simulated Infrastructure Manager	Authorized users MUST be able to check the status (e.g. Running, Stopped, Ready for instantiation, etc.) of the instantiated virtual environment templates and all the involved virtual machines.
FUNC-92	Functional	Pluggable Deployment Manager (xOpera)	Simulated Infrastructure Manager SHOULD demonstrate the capacity to instantiate a training scenario on at least two different IaaS providers.
FUNC-93	Functional	Economic Risk Models	The risk models MUST provide an estimate of risk level, with respect to the training scenario including the value of the simulated asset, in terms of the

ID	Type	Traced Asset		Description
				likelihood of the risk as well as the impact of the risk in terms of monetary loss.
FUNC-94	Functional	Economic Models	Risk	The risk models MUST reflect risks the individual client organisation and system are exposed to.
FUNC-95	Functional	Economic Models	Risk	The CYBERWISER.eu Platform MUST provide comprehensive training material for development and selection of risk models and corresponding real-time assessment and response proposal algorithms for a client organisation.
FUNC-96	Functional	Economic Models	Risk	The training material on development and selection of risk models provided by The CYBERWISER.eu Platform MUST be aimed at different levels of trainee skills.
FUNC-97	Functional	Economic Evaluator	Risk	The Economic Risk Evaluator MUST provide economic risk assessment reports based on the business topology information, detected vulnerabilities, events, and alarms.
FUNC-98	Functional	Economic Evaluator	Risk	Each automatic mitigation action suggested by the Economic Risk Evaluator to the blue team MUST be associated with a certain cost.
FUNC-99	Functional	Economic Evaluator	Risk	It MUST be possible to assign values to organization's virtual resources (simulated as elements within the training scenarios), reflecting their monetary values.
FUNC-100	Functional	Economic Evaluator	Risk	During the course of a training scenario in the cyber-range, it MUST be possible to continuously assess exposure of the organizations' simulated assets to cyber-risk.
FUNC-101	Functional	Economic Evaluator	Risk	The Economic Risk Evaluator MUST offer a flexible interface to execute different risk models.
FUNC-102	Functional	Economic Evaluator	Risk	The Economic Risk Evaluator MUST generate risk reports and store them.
FUNC-103	Functional	Economic Evaluator	Risk	The Economic Risk Evaluator MUST show both economic risk exposure and possible mitigation actions.
FUNC-104	Functional	Economic Evaluator	Risk	The Economic Risk Evaluator SHOULD offer a graphical user interface to the user.
FUNC-105	Functional	Anomaly Reasoner	Detection	Based on the recorded events from the training scenario, it MUST be possible to detect anomalous activity in the cyber-range.



ID	Type	Traced Asset	Description
FUNC-106	Functional	Anomaly Detection Reasoner	The Anomaly Detection Reasoner MUST support several types of sensors used in the training scenarios.
FUNC-107	Functional	Anomaly Detection Reasoner	The Anomaly Detection Reasoner MUST process the information collected by the sensors deployed in the monitored simulated infrastructure.
FUNC-108	Functional	Anomaly Detection Reasoner	The Anomaly Detection Reasoner MUST be able to record events and raise alarms with the relevant information related to the cyber-risk faced by the monitored simulated infrastructure.
FUNC-109	Functional	Anomaly Detection Reasoner	The Anomaly Detection Reasoner MUST be accessible by the defending team so that they can use it to supervise the exercise.
FUNC-110	Functional	Vulnerability Assessment Tools	It MUST be possible to detect whether vulnerabilities that were initially present as a part of the scenario definition are still present at a particular point in the cyber-range exercise.
FUNC-111	Functional	Vulnerability Assessment Tools	Vulnerability Assessment Tools COULD automatically execute network reconnaissance procedures towards the simulated environment.
FUNC-112	Functional	Vulnerability Assessment Tools	Vulnerability Assessment Tools MUST provide access to several vulnerability scanning tools.
FUNC-113	Functional	Attack Simulator	Attack Simulator MUST be able to execute attack scripts provided in the training scenario definition.
FUNC-114	Functional	Attack Simulator	It MUST be possible to modify the attack scripts during an ongoing cyber-range exercise.
FUNC-115	Functional	Attack Simulator	Attack Simulator MUST be able to execute an attack script according to a pre-defined timeline (specified in the training scenario definition).
FUNC-116	Functional	Attack Simulator	It MUST be possible to modify the schedules of attack scripts during an ongoing cyber-range exercise.
FUNC-117	Functional	Attack Simulator	Red team MUST be offered a series of pre-defined attack templates/scripts, which they can use as a starting ground for launching automated attacks.
FUNC-118	Functional	Attack Simulator	Red team MUST be able to customize attack templates/scripts offered by the CYBERWISER.eu Platform during an exercise, in order to allow generating new attacks or modifying the existing ones with the specifics pertaining a particular training scenario or exercise environment.



ID	Type	Traced Asset	Description
FUNC-119	Functional	Attack Simulator	Red team MUST be able to implement new attack scripts during the cyber-range exercise.
FUNC-120	Functional	Attack Simulator	Trainees SHOULD be able to share attack scripts with other users.
FUNC-121	Functional	Countermeasures Simulator	The Countermeasures Simulator MUST provide a catalogue of countermeasures that can be chosen by the trainees during the execution of the training exercise to be executed automatically.
FUNC-122	Functional	Countermeasures Simulator	Blue team MUST be allocated a limited (pre-defined) budget, which they can spend on automatically running specific mitigation actions. Suggestions for appropriate mitigation actions and associated costs are coming from the CYBERWISER.eu Platform.
FUNC-123	Functional	Countermeasures Simulator	When a specific mitigation action selected by the blue team would take too long to execute in real-time, the Countermeasures Simulator SHOULD be capable of simulating it.
FUNC-124	Functional	Countermeasures Simulator	In the training scenarios, blue team MUST have access to appropriate defence tools, for instance monitoring tools, IDS/IPS systems, and honeypots, in order to defend organization's simulated assets.
FUNC-125	Functional	Countermeasures Simulator	Blue team players MUST be offered a series of pre-defined countermeasures in the shape of templates/scripts or some other format, which they can use as tools to protect the infrastructure.
FUNC-126	Functional	Countermeasures Simulator	Blue team MUST be able to customize countermeasures means offered by the CYBERWISER.eu Platform in order to allow generating new security actions for a particular training scenario.
FUNC-127	Functional	Countermeasures Simulator	Blue team MUST be able to implement new countermeasure scripts during an exercise.
FUNC-128	Functional	Countermeasures Simulator	Trainees SHOULD be able to share countermeasures scripts with other users.
FUNC-129	Functional	Countermeasures Simulator	Blue team MUST be able to search for specific countermeasures within the catalogue basing on keywords.
FUNC-130	Functional	Countermeasures Simulator	Countermeasures Simulator MUST be able to execute countermeasures means following a specific schedule. This applies to the case when the CYBERWISER.eu Platform acts as blue team.

ID	Type	Traced Asset	Description
FUNC-131	Functional	Countermeasures Simulator	The countermeasures and their schedules MUST be dynamically adaptable by the white team during the scenario execution.
FUNC-132	Functional	Countermeasures Simulator	The white team MUST be able to add new countermeasures dynamically during the execution of the training.
FUNC-133	Functional	Centralized Logging Component	Centralized Logging Component MUST be capable of storing and searching JSON data.
FUNC-134	Functional	Centralized Logging Component	The Centralized Logging Component MUST expose a web user interface for access to logs by human operators.

Table 82. Functional requirements

ID	Type	Traced Asset	Description
T-PLAT-1	Platform	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST support a Single-Sign-On mechanism (SSO) for users to log in.
T-PLAT-2	Platform	CYBERWISER.eu Platform	All relevant managed events that occur in a training scenario instance (for instance attack triggered, attack detected, progress status changed) MUST be logged.
T-PLAT-3	Platform	CYBERWISER.eu Platform	A Message-oriented-Middleware (MoM) service implementing the AMQP protocol MUST be made available to every CYBERWISER.eu component that interfaces it.
T-PLAT-4	Platform	CYBERWISER.eu Platform	All CYBERWISER.eu components designed to run within training scenario instances MUST be packaged in a way that allows their storage and automated deployment by the Digital Library.
T-PLAT-5	Platform	Cross-Learning Facilities	The CYBERWISER.eu Platform MUST integrate, in a Workspace Area, various features supporting E-Learning, including document management, broadcasting web seminars, desktop sharing, virtual classrooms and online courses.
T-PLAT-6	Platform	Scenario Designer	The Scenario Designer SHOULD be a sub-component of the Training Manager.
T-PLAT-7	Platform	Scenario Designer	The Scenario Designer MUST be able to outline the training scenario requests in a human-readable format.
T-PLAT-8	Platform	Scenario Designer	The Scenario Designer MUST allow to export the outline of the training scenario requests in a file (e.g. JSON, or XML).
T-PLAT-9	Platform	Scenario Designer	The Scenario Designer MUST implement an interface with the Simulated Infrastructure Manager.
T-PLAT-10	Platform	Scenario Designer	The Scenario Designer MUST implement an interface with the Digital Library.
T-PLAT-11	Platform	Scenario Designer	The Scenario Designer's underlying Data Model MUST be in common with the Simulated Infrastructure Manager and the Digital Library.

ID	Type	Traced Asset	Description
T-PLAT-12	Platform	Training Manager	The Training Manager MUST restore its state with the last persisted data after crash or reboot of its server-side application.
T-PLAT-13	Platform	Training Manager	Data Model used by the Training Manager MUST be the same as the Data Model of the Simulated Infrastructure Manager and the Digital Library.
T-PLAT-14	Platform	Simulated Infrastructure Manager	Simulated Infrastructure Manager SHOULD demonstrate the capacity to translate a training scenario request to virtual environment template for at least two different IaaS providers.
T-PLAT-15	Platform	Simulated Infrastructure Manager	The training scenario requests generated by the Scenario Designer MUST be translatable into simulated environment templates for the underlying Simulated Infrastructure Manager.
T-PLAT-16	Platform	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST implement an interface with the Scenario Designer.
T-PLAT-17	Platform	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST implement an interface with the Digital Library.
T-PLAT-18	Platform	Simulated Infrastructure Manager	All training scenario instance VMs MUST allow VNC connections.
T-PLAT-19	Platform	Simulated Infrastructure Manager	Simulated Infrastructure Manager graphical user interface MUST integrate a web-based VNC client.
T-PLAT-20	Platform	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST provide the business topology information to the Economic Risk Evaluator through contextualization files at the instantiation of a training scenario.
T-PLAT-21	Platform	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST provide attack scripts and configuration to the Attack Simulator through contextualization files at the instantiation of a training scenario.
T-PLAT-22	Platform	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST provide vulnerability scanner configuration and custom scripts to the Vulnerability Assessment Tools through contextualization files at the instantiation of a training scenario.
T-PLAT-23	Platform	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST provide configuration to the Countermeasures Simulator through contextualization files at the instantiation of a training scenario.
T-PLAT-24	Platform	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST inform the Anomaly Detection Reasoner about the instantiation of a new training scenario using a REST API call so that it can isolate its monitoring events.
T-PLAT-25	Platform	Economic Risk Evaluator	The Economic Risk Evaluator SHOULD allow configuration by means of the interface or by executing a batch file.

ID	Type	Traced Asset	Description
T-PLAT-26	Platform	Economic Risk Evaluator	The Economic Risk Evaluator MUST expose an HTTPS interface for retrieval of risk evaluation reports by the Performance Evaluator.
T-PLAT-27	Platform	Anomaly Detection Reasoner	The Anomaly Detection Reasoner MUST be able to manage the multitenant concept to permit the parallel execution of exercises and to control different scenarios/users and RabbitMQ instances.
T-PLAT-28	Platform	Anomaly Detection Reasoner	The Anomaly Detection Reasoner MUST send the information about the activation/deactivation of every anomaly detected during the execution of a training scenario to a message queue, enabling subsequent storage of this data in the Centralized Logging Component.
T-PLAT-29	Platform	Anomaly Detection Reasoner	The Anomaly Detection Reasoner MUST send the information about alarms to a message queue, enabling the subsequent storage of this data in the Centralized Logging Component.
T-PLAT-30	Platform	Vulnerability Assessment Tools	Vulnerability Assessment Tools MUST produce results (in the form of a vulnerability report) that can be used as input for the Economic Risk Evaluator.
T-PLAT-31	Platform	Vulnerability Assessment Tools	Vulnerability Assessment Tools MUST propagate vulnerability reports to the Centralized Logging Component via a message queue.
T-PLAT-32	Platform	Vulnerability Assessment Tools	Vulnerability Assessment Tools MUST be accessible to both blue and red team trainees.
T-PLAT-33	Platform	Vulnerability Assessment Tools	Vulnerability Assessment Tools MUST support execution of custom scripts that scan for specific vulnerabilities.
T-PLAT-34	Platform	Vulnerability Assessment Tools	Vulnerability Assessment Tools MUST encapsulate at least OWASP ZAP and w3af vulnerability scanners.
T-PLAT-35	Platform	Vulnerability Assessment Tools	It MUST be possible to integrate additional scanning tools into the scanning suite offered by Vulnerability Assessment Tools.
T-PLAT-36	Platform	Monitoring Sensors	A set of Monitoring Sensors MUST be available for deployment on/alongside simulated ICT resources associated with a given training scenario instance.
T-PLAT-37	Platform	Simulated Infrastructure Manager	Monitoring sensors MUST be deployed and connected to the scenario in an automated way, without manual intervention.
T-PLAT-38	Platform	Monitoring Sensors	Network traffic within training scenario instances MUST be monitored with appropriate network-based tools, where it is relevant for the training scenario.
T-PLAT-39	Platform	Monitoring Sensors	During the execution of a training scenario, it MUST be possible to detect events on simulated hosts like changes in configuration and system logs, where it is relevant for the scenario.
T-PLAT-40	Platform	Monitoring Sensors	All sensors MUST continuously monitor simulated ICT resources in accordance with the training scenario definition.

ID	Type	Traced Asset	Description
T-PLAT-41	Platform	Monitoring Sensors	Monitoring data MUST be streamed to the Anomaly Detection Reasoner for further analysis.
T-PLAT-42	Platform	Attack Simulator	The Attack Simulator MUST report execution of every attack to the Centralized Logging Component via a message queue.
T-PLAT-43	Platform	Attack Simulator	The Attack Simulator MUST support running at least Metasploit exploits (Ruby scripts), Python and bash scripts.
T-PLAT-44	Platform	Attack Simulator	The Attack Simulator SHOULD expose a Web UI for configuration of its behaviour (attack scripts and schedules) during an ongoing cyber-range exercise.
T-PLAT-45	Platform	Countermeasures Simulator	Countermeasures Simulator MUST report every countermeasure execution to the Centralized Logging Component via a message queue.
T-PLAT-46	Platform	Centralized Logging Component	The Centralized Logging Component MUST receive the data through the Message-oriented-Middleware (MoM) service deployed within the training scenario instances.
T-PLAT-47	Platform	Centralized Logging Component	The Centralized Logging Component MUST store the received data to a document storage database.
T-PLAT-48	Platform	Centralized Logging Component	The Centralized Logging Component MUST expose a REST API for accessing and searching the data.

Table 83. Platform requirements

ID	Type	Traced Asset	Description
LEGL-1	Legal	CYBERWISER.eu Platform	The CYBERWISER.eu Platform as a whole MUST ensure compliance with applicable EU and national legislation for handling personal data (GDPR, Draft ePrivacy Regulation).
LEGL-2	Legal	CYBERWISER.eu Platform	Any data in the simulated environment of publicly accessible training exercises MUST be synthetic, openly available or anonymised.
LEGL-3	Legal	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST provide a privacy notice concerning data inserted by users during the training exercises. The privacy notice must include information about data storage location, access to data, and data processing. Users should understand who can access the data provided, and be notified that, in case they are inserting personal/sensitive data, they are responsible for obtaining needed consents for the use of these data.
LEGL-4	Legal	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST ensure that data brought by users to training exercises is protected and implement appropriate access control and security measures. Data must not be further processed (other than for simulation purposes) or shared.

ID	Type	Traced Asset	Description
LEGL-5	Legal	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST provide a privacy notice concerning personal data of users. The privacy notice must include information about data storage location, access to data, and data processing. CYBERWISER.eu must obtain needed consents for the use of these data from users.

Table 84. Legal requirements

ID	Type	Traced Asset	Description
T-SECU-1	Security	CYBERWISER.eu Platform	All user interfaces of the CYBERWISER.eu Platform MUST provide a logout capability for user-initiated communications sessions. An explicit logout message will be displayed indicating the reliable termination of authenticated communication session.
T-SECU-2	Security	CYBERWISER.eu Platform	All Client and server applications of the CYBERWISER.eu Platform MUST protect the authenticity of communication sessions between them.
T-SECU-3	Security	CYBERWISER.eu Platform	All Client and server applications of the CYBERWISER.eu Platform MUST implement identification and authorization of Users. No actions are allowed on the client-side before successful authentication.
T-SECU-4	Security	CYBERWISER.eu Platform	All Client and server applications of the CYBERWISER.eu Platform MUST protect the confidentiality, integrity and authentication of transmitted information using cryptographic means (e.g. authenticated transport layer security protocols such SSL/TLS).
T-SECU-5	Security	CYBERWISER.eu Platform	All Client and server applications of the CYBERWISER.eu Platform MUST implement system permissions assigned to specific User Roles (Role Based Access Control policies).
T-SECU-6	Security	CYBERWISER.eu Platform	All Client and server applications of the CYBERWISER.eu Platform MUST invalidate session identifiers upon user logout or other session terminations.
T-SECU-7	Security	Cross-Learning Facilities	Configuration (access control) of each User Cluster MUST be possible, to direct the users of such cluster to an adequately configured Workspace Area.
T-SECU-8	Security	Cross-Learning Facilities	Workspace Areas MUST provide access reserved to a-priori-identified user groups, accessing a pre-determined (and configurable) set of services.
T-SECU-9	Security	Digital Library	The metadata of the virtual environment templates and the catalogue of pre-defined training scenarios stored within the Digital Library repositories MUST be



ID	Type	Traced Asset	Description
			encrypted with strong standardized encryption and decryption algorithms.
T-SECU-10	Security	Training Manager	Server-side Training Manager application SHOULD perform validation for all exchanged data. Data validation will be performed against the data schema of the interfaces. Invalid data will be discarded, related User (if available) will be alerted, event will be logged.
T-SECU-11	Security	Training Manager	The sensitive information stored within the server-side repository of the Training Manager MUST be encrypted with strong standardized encryption and decryption algorithms.
T-SECU-12	Security	Training Manager	Client and server-side Training Manager MUST implement system permissions assigned to specific User Roles (Role Based Access Control policies).
T-SECU-13	Security	Training Manager	Client and server-side Training Manager MUST invalidate session identifiers upon user logout or other session terminations.
T-SECU-14	Security	Simulated Infrastructure Manager	Client and server-side Simulated Infrastructure Manager MUST invalidate session identifiers upon user logout or other session terminations.
T-SECU-15	Security	Simulated Infrastructure Manager	Client and server-side Simulated Infrastructure Manager MUST implement system permissions assigned to specific User Roles (Role Based Access Control policies).
T-SECU-16	Security	Simulated Infrastructure Manager	Instances of different training scenarios simultaneously running in the cyber-range MUST be able to be isolated (from a networking point of view) from each other.
T-SECU-17	Security	Simulated Infrastructure Manager	Server-side Simulated Infrastructure Manager application SHOULD perform validation for all exchanged data. Data validation will be performed against the data schema of the interfaces. Invalid data will be discarded, related user (if available) will be alerted, event will be logged.
T-SECU-18	Security	Simulated Infrastructure Manager	The sensitive information stored within the server-side repository of the Simulated Infrastructure Manager MUST be encrypted with strong standardized encryption and decryption algorithms.
T-SECU-19	Security	Anomaly Detection Reasoner	The confidentiality and integrity of the information communicated to/from the Anomaly Detection Reasoner MUST be guaranteed.
T-SECU-20	Security	Anomaly Detection Reasoner	The Anomaly Detection Reasoner MUST not be visible for the red team.

Table 85. Security requirements

ID	Type	Traced Asset	Description
T-USAB-1	Usability	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST support design of training scenarios appropriate for training of individuals/teams of various skill levels.
T-USAB-2	Usability	CYBERWISER.eu Platform	From the standpoint of end-users, training scenario instantiation in the cyber-range infrastructure MUST be asynchronous.
T-USAB-3	Usability	CYBERWISER.eu Platform	All user interfaces of the CYBERWISER.eu Platform MUST be reactive enough to allow efficient user interactions.
T-USAB-4	Usability	CYBERWISER.eu Platform	The user interfaces of the CYBERWISER.eu Platform MUST be compatible with Chrome web browser.
T-USAB-5	Usability	CYBERWISER.eu Platform	All user interfaces of the CYBERWISER.eu Platform SHOULD be compatible with Firefox, Edge and Safari web browsers.
T-USAB-6	Usability	CYBERWISER.eu Web Portal	The CYBERWISER.eu Web Portal (www.cyberwiser.eu) MUST be the single point of access to the CYBERWISER.eu Platform.
T-USAB-7	Usability	Cross-Learning Facilities	Workspace Areas MUST integrate/implement a mail service to enable email notification when a new event (for example: new message/upload) is triggered in a given Workspace.
T-USAB-8	Usability	Cross-Learning Facilities	The Cross-Learning Facilities MUST provide a user-friendly graphical user interface.
T-USAB-9	Usability	Cross-Learning Facilities	Cross-Learning Facilities SHOULD present a different dashboard according to the user's role.
T-USAB-10	Usability	Scenario Designer	The Scenario Designer SHOULD provide a user-friendly environment for the definition and configuration of training scenario requests.
T-USAB-11	Usability	Scenario Designer	All drawing dashboards of the Scenario Designer MUST allow to zoom in/out on the drawing canvas.
T-USAB-12	Usability	Scenario Designer	All drawing dashboards of the Scenario Designer MUST allow to group elements on the drawing canvas.
T-USAB-13	Usability	Centralized Logging Component	The Centralized Logging Component's web UI MUST support viewing, filtering, and searching of data.

Table 86. Usability requirements

ID	Type	Traced Asset	Description
T-PERF-1	Performance	CYBERWISER.eu Web Portal	Availability of the entry point of CYBERWISER.eu Web Portal to the CYBERWISER.eu Platform SHOULD be 99%, measured in the timeframe of one month.
T-PERF-2	Performance	Scenario Designer	All drawing dashboards of the Scenario Designer MUST support



ID	Type	Traced Asset	Description
			ICT/Application/Business topologies with up to 500 elements each.
T-PERF-3	Performance	Scenario Designer	The Scenario Designer MUST be capable of managing up to 10 definitions of training scenario requests at the same, from different Users.
T-PERF-4	Performance	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST automatically scale available computational resources of the cyber-range infrastructure with respect to the amount of training scenario instances and the needed capacity of each scenario instance.
T-PERF-5	Performance	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST take no more than 500 milliseconds to process (receive and initiate the validation activities) a simulation environment templates leveraging a training scenario request.
T-PERF-6	Performance	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST be capable of managing up to 10 training scenario requests from the Scenario Designer at the same time.
T-PERF-7	Performance	Simulated Infrastructure Manager	The Simulated Infrastructure Manager Virtualization system CPU/Memory/Hard Disk capability MUST be scalable by adding new IaaS/Hypervisor host nodes.
T-PERF-8	Performance	Economic Risk Evaluator	The Economic Risk Evaluator MUST inform about the economic risks during the execution of the exercises in real-time.
T-PERF-9	Performance	Vulnerability Assessment Tools	Presence (or absence) of vulnerabilities, defined in the scenario (known to the trainer), SHOULD be detected with a frequency that allows observability. Applicable to scenarios where the blue team's goal is to fix known vulnerabilities.

Table 87. Performance requirements

## 9 Conclusions

Deliverable D2.5 documents the final design of the CYBERWISER.eu Platform. The design has been carried out in a continuous and smooth way encompassing the information coming from different sources. These are the requirements defined by the Consortium (also considering the specific set coming from the pilots), the continuous feedback received from the development and integration activities, as well as the outputs produced by the go-to-market activities. This design has consolidated and extended the work done in the initial version (D2.3), giving more details on the internal operation of the components and adding more views that complete the approach given by the Consortium. This allows a comprehensive understanding of the platform. The user perspective has been added by means of the sections on user interfaces and use cases. The traceability with the requirements and the alignment with the market needs have been revised.

The results documented in this deliverable guide the implementation and integration activities which will lead to the delivery of the CYBERWISER.eu platform as main output of the project.