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

D2.3 Platform Design, Initial Version

Work Package	WP2 Requirements, design and building blocks
Lead author	Antonio Álvarez (ATOS)
Contributors	Javier Ramírez (ATOS), Gencer Erdogan (SINTEF), Tobias Groothuyse (RHEA), Matteo Merialdo (RHEA), Niccolò Zazzeri (TRUST-IT), Paolo Lombardi (TRUST-IT), Cristina Mancarella (TRUST-IT), Manca Bizjak (XLAB), Anze Zitnik (XLAB)
Peer reviewers	Liliana Ribeiro (EDP), Dario Varano (UNIPI)
Version	V1.0
Due Date	28/02/2018
Submission Date	28/02/2018

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	17/12/2018	Antonio Álvarez (ATOS)	Deliverable skeleton
0.1	18/12/2018	Antonio Álvarez (ATOS)	Refined skeleton, after XLAB feedback.
0.2	08/01/2019	Javier Ramírez (ATOS), Antonio Álvarez (ATOS	Input to Sections 1.4 and 3
0.3	08/01/2019	Gencer Erdogan (SINTEF)	Input to Section 5.7.1.
0.4	11/01/2019	Tobias Groothuyse (RHEA), Javier Ramírez (ATOS)	Input to sections 4.1, 4.2, 5.6. Refinements in section 3
0.5	11/01/2019	Antonio Álvarez (ATOS)	Adding references to section 5.7.1
0.6	11/01/2019	Niccolò Zazzeri (TRUST-IT)	Contribution to section 1.4, 5.1, 5.2
0.7	14/01/2019	Tobias Groothuyse (RHEA)	Feedback for improvement of section 3 and refinement of section 4.2.
0.8	15/01/2019	Antonio Álvarez (ATOS)	Contribution to section 5.9
0.9	15/01/2019	Antonio Álvarez (ATOS), Javier Ramírez (ATOS), Tobias Groothuyse (RHEA), Manca Bizjak (XLAB), Anze Zitnik (XLAB)	Contribution to section 1.4, 5.3, 5.4, 5.5, 5.7. Refinements to section 3
0.10	16/01/2019	Antonio Álvarez (ATOS), Javier Ramírez (ATOS), Tobias Groothuyse (RHEA)	Contribution to section 5.10. Refinements to sections 3.2 and 4.2
0.11	16/01/2019	Javier Ramírez (ATOS)	Refinement of sections 5.7 and 5.10
0.12	18/01/2019	Antonio Álvarez (ATOS)	Refinement of sections 5.5. and 5.7
0.13	25/01/2019	Antonio Álvarez (ATOS), Matteo Merialdo (RHEA), Tobias Groothuyse (RHEA), Manca Bizjak (XLAB), Javier Ramírez (ATOS)	Input to section 2, 5.3.1. Refinement section 5.7
0.14	26/01/2019	Manca Bizjak (XLAB), Anze Zitnik (XLAB)	Input to sections 5.8 and 5.11
0.15	28/01/2019	Manca Bizjak (XLAB), Anze Zitnik (XLAB)	Refinement of sections 3.2.10 and 5.8
0.16	28/01/2019	Antonio Álvarez (ATOS)	Intermediate clean version
0.17	30/01/2019	Antonio Álvarez (ATOS), Javier Ramírez (ATOS), Tobias Groothuyse (RHEA)	Input to sections 1.1, 1.3, 5.12. Refinement of sections 1.4, 2.2, 2.3, 3,3.2, 4.2, 5.7, 5.8, 5.9, 5.11.
0.18	31/01/2019	Antonio Álvarez (ATOS)	Executive summary. Input to section 1.2. Refinements of sections 1.1 and 1.4
0.19	02/02/2019	Antonio Álvarez (ATOS), Anze Zitnik (XLAB)	Refinement in section 5.8, 5.12

0.20	03/02/2019	Matteo Merialdo (RHEA)	Input to sections 6 and 7
0.21	12/02/2019	Tobias Groothuyse (RHEA), Gencer Erdogan (SINTEF), Antonio Álvarez (ATOS)	Input and refinements to sections 5.3, 5.4, 5.5, 5.6, 5.7.1, 5.9, 5.12
0.22	12/02/2019	Antonio Álvarez (ATOS)	Several refinements across the document
0.23	13/02/2019	Antonio Álvarez (ATOS)	Input to sections 1, 6, 8. Refinement of the Executive Summary. Update of figure in section 5.5. Cross-references adjustment
0.24	13/02/2019	Antonio Álvarez (ATOS)	Clean version ready for QA review
0.25	20/02/2019	Antonio Álvarez (ATOS), Javier Ramírez (ATOS), Niccolò Zazzeri (TRUST-IT), Paolo Lombardi (TRUST-IT), Cristina Mancarella (TRUST-IT), Liliana Ribeiro (EDP), Dario Varano (UNIPI)	Update of the Level 2 component diagram in section 3.2. Refinement of sections 3.2.2, 3.2.4, 3.2.5, 3.2.16, 5.4. Corrections in section 7. Refinement of section 5.2. QA feedback and answer. Adding a keywords section
0.26	21/02/2019	Antonio Álvarez (ATOS)	Document layout
0.27	26/02/2019	Antonio Álvarez (ATOS), Dario Varano (UNIPI), Liliana Ribeiro (EDP)	Second iteration of the QA review and fixing minor issues
0.28	26/02/2019	Antonio Álvarez (ATOS)	Clean version. Document layout
1.0	28/02/2019	María Teresa García (ATOS), Antonio Álvarez	Quality Check. 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), Niccolò Zazzeri (TRUST-IT), Tobias Groothuyse (RHEA)
2. Methodology	Matteo Merialdo (RHEA), Tobias Groothuyse (RHEA)
3. CYBERWISER.EU top-down initial design	Javier Ramírez ATOS), Tobias Groothuyse (RHEA), Manca Bizjak (XLAB), Anze Zitnik (XLAB)
4. Other design perspectives	Javier Ramírez (ATOS), Tobias Groothuyse (RHEA)
5. Detail of the building components	Antonio Álvarez (ATOS), Javier Ramírez (ATOS), Gencer Erdogan (SINTEF), Tobias Groothuyse (RHEA), Niccolò Zazzeri (TRUST-IT), Paolo Lombardi (TRUST-IT), Cristina Mancarella (TRUST-IT), Manca Bizjak (XLAB), Anze Zitnik (XLAB)
6. Impact of the business requirements	Matteo Merialdo (RHEA), Antonio Álvarez (ATOS)
7. Initial requirements traceability	Matteo Merialdo (RHEA)
Conclusions	Antonio Álvarez (ATOS)

Keywords

CYBERWISER.eu; design; components; cyber range; interface; requirements; diagram; software; deployment; implementation; integration; traceability; infrastructure; risk; library; monitoring; attack; countermeasure; event;modelling; methodology

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

List of figures

List of tables

Executive Summary

This deliverable is the first output of Task T2.2 entitled “Overall Vision and Design”. It reports the work done during the initial stage of the design of the platform (M3-M6 [November 2018 – February 2019], being the final one the period M7-M12 [March 2019 – August 2019]). 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 components of the platform are presented, and their role explained. The interfaces between different components are described, remarking which components provide or consume them.

In the first stage of the design we carried out a review of the system-level requirements, subsystem requirements and preliminary architectural vision and system overview of the CYBERWISER.eu Platform. This led to the preparation of a general component diagram, presented at two different levels (section 1), and two more diagrams presenting two more perspectives that help understand the system and provide an overall vision: one is based on the software components structure (section 4.1) and another one on the software deployment (section 4.2). For each component an internal diagram is presented to better understand how they work (section 1). This falls within the scope of D2.3 and is produced considering the requirements produced within Task T2.1 as the main source of information. Later, D2.5 will document the final design, updating the information presented in D2.3 and including new perspectives that will provide a full understanding of the platform and will ease its smooth implementation. This will be done in close monitoring of the initial implementation and integration activities in T2.3, T2.4 and T3.1, and in alignment with pilot needs expressed in T5.1 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 design has been developed in iterations, starting from the high-level system elements to continue to the subsystems, with the aim of considering all elicited requirements. The impact of the business requirements, presented in D2.1 [1] and D6.1 [2] is addressed in section 1 and in section 1 the design is confronted with the requirements documented in D2.1 [1] to study its traceability and check that all the requirements are being taken into account during the design process.

All in all, the main takeaways of the document are the following:

- Creation of high-level component diagrams, including the definition of components and the interfaces and communication protocols used.
- Creation of complementary diagrams, in particular two types: a 3-tier-diagram representing the software components structure perspective and the software deployment perspective.
- Description of the building components, namely: the Web Portal, the Cross-Learning Facilities, Simulated Infrastructure Manager, Training Manager, Performance Evaluator, Digital Library, Economic Risk Evaluator, Vulnerability Assessment Tools, Monitoring Sensors, Anomaly Detection Reasoner, Attack Simulator, Countermeasures Simulator, Centralized Logging Component, RabbitMQ Server and the Event-based Service Orchestrator. Such description includes the main features, role in the platform, internal architecture and internal operation.
- Requirements traceability.

1. Introduction

1.1 Purpose and scope of the document

Deliverable D2.3, entitled “Platform Design, Initial Version” is the first of the two deliverables produced by Task T2.2 entitled “Overall vision and design” within Work Package WP2 entitled “Requirements, Design and Building Blocks”. This task starts in month M3 (November 2018) and finishes in M12 (August 2019). D2.3 is scheduled in M6 (February 2019) while D2.5, which documents the final version of the design is delivered in M12 (August 2019). This task belongs to WP2 entitled “Requirements, Design and Building Blocks”, which starts in M1 (September 2018) and finishes in M20 (April 2020).

This deliverable compiles the first approach to the design of the platform. The components of the platform are presented and their role explained. Also, 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, following a top-down initial approach, two diagrams of different level of detail are presented reflecting the general picture. This provides an overall perspective as starting point. Then two more perspectives are 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 an internal diagram is presented to better understand how they work. This is done in section 1 where we address the internal detail of the building components.

This is the scope of D2.3, but D2.5 will include new perspectives, like the software technology view[1], activity diagrams, or the detailed explanation of use cases, to name but a few. Also, D2.5 will offer detailed information about the user interface and the integration of the CYBERWISER.eu Platform with the project portal. The data model will be presented as well. Needless to say, it will also include the update of the already existing perspectives and the internal diagram per building component.

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 methodology covers the engineering process of the system design and the architecture, the modelling languages 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. The different components are presented and their role in the platform is explained.
- Section 4: It adds, for the sake of completeness at this stage in the design, two more diagrams describing the platform from two perspectives: the software components perspective which is in essence a tier diagram, and the software deployment perspective.
- 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: It addresses the impact of the business requirements on the design.
- Section 7: It is the initial traceability between the design and the requirements.

[1] This view offers the information about the technology stack adopted for CYBERWISER.eu

- Section 8: Conclusions and closing remarks.

1.3 Relation to other work in the project

Being CYBERWISER.eu a very complex platform during the definition of the project it was decided that the task would have two stages:

- In the initial stage, a first approach is made to the challenge of designing the platform. In this stage it is very relevant the coordination with the requirements elicitation task (T2.1 “Requirements [M1-M6]) which is the main information input to the task. It is important to ensure that the design covers what is demanded in the requirements. Also, it is relevant to note that the delivery of the final requirements (D2.2, M6[2]) is synchronized with the delivery of the initial design (D2.3, M6).
- During the second stage, the design is refined and the aspects that are left at a high level in D2.3 are more and more detailed as T2.2 will run 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 is consolidated and delivered in D2.5 “Platform Design, Final Version”. In addition, there is mutual influence between T2.2 and all the tasks running in WP4, entitled “Training Materials, Scenarios and Evaluation (M1-M24).

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
COTS	Commercial-Off-The-Shelf
CLC	Centralized Logging Component
CPU	Central Processing Unit
CS	Countermeasures Simulator
DL	Digital Library
ERE	Economic Risk Evaluator
ERM	Economic Risk Models
GUI	Graphical User Interface
HTTPS	HyperText Transfer Protocol Secure
IaaS	Infrastructure-as-a-Service
ICT	Information and Communications Technology
IP	Internet Protocol
KVM	Kernel-based Virtual Machine
LCMS	Learning Content Management System
PaaS	Platform as a Service

[2] The initial requirements version is compiled in D2.1 (M3, November 2018)

Acronym	Description
PE	Performance Evaluator
RAE	Risk Assessment Engine
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
UML-CD	Unified Modelling Language – Component Diagram
URL	Uniform Resource Locator
VAT	Vulnerability Assessment Tool
VM	Virtual Machine
VNC	Virtual Network Computing

Table 1. Table of acronyms

2. Methodology

2.1 System Architecture/design engineering process

2.1.1 Methodology for defining the system architecture

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 [1] as a system overview. This preliminary architecture has been refined for this D2.3 and it will be further detailed within 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, starting from the high-level system elements and continue to the 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 [1]). With further iterations, it was possible to detail the initial design from the overview in D2.1 [1] to a detailed level (D2.3 and 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, leveraging 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 Modeling 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 are used, D2.5 will encompass also 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. Being CYBERWISER.eu 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 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 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 [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 Matrices have been created, reported in this D2.3 and in D2.5 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 initial 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 1and 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.

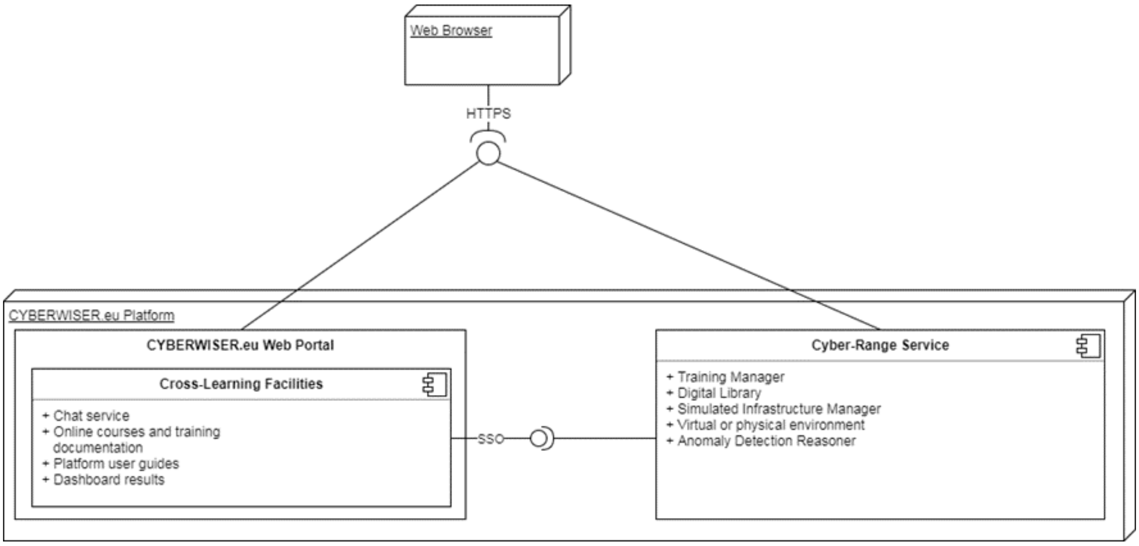

Figure 1: Level 1 component diagram

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 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.

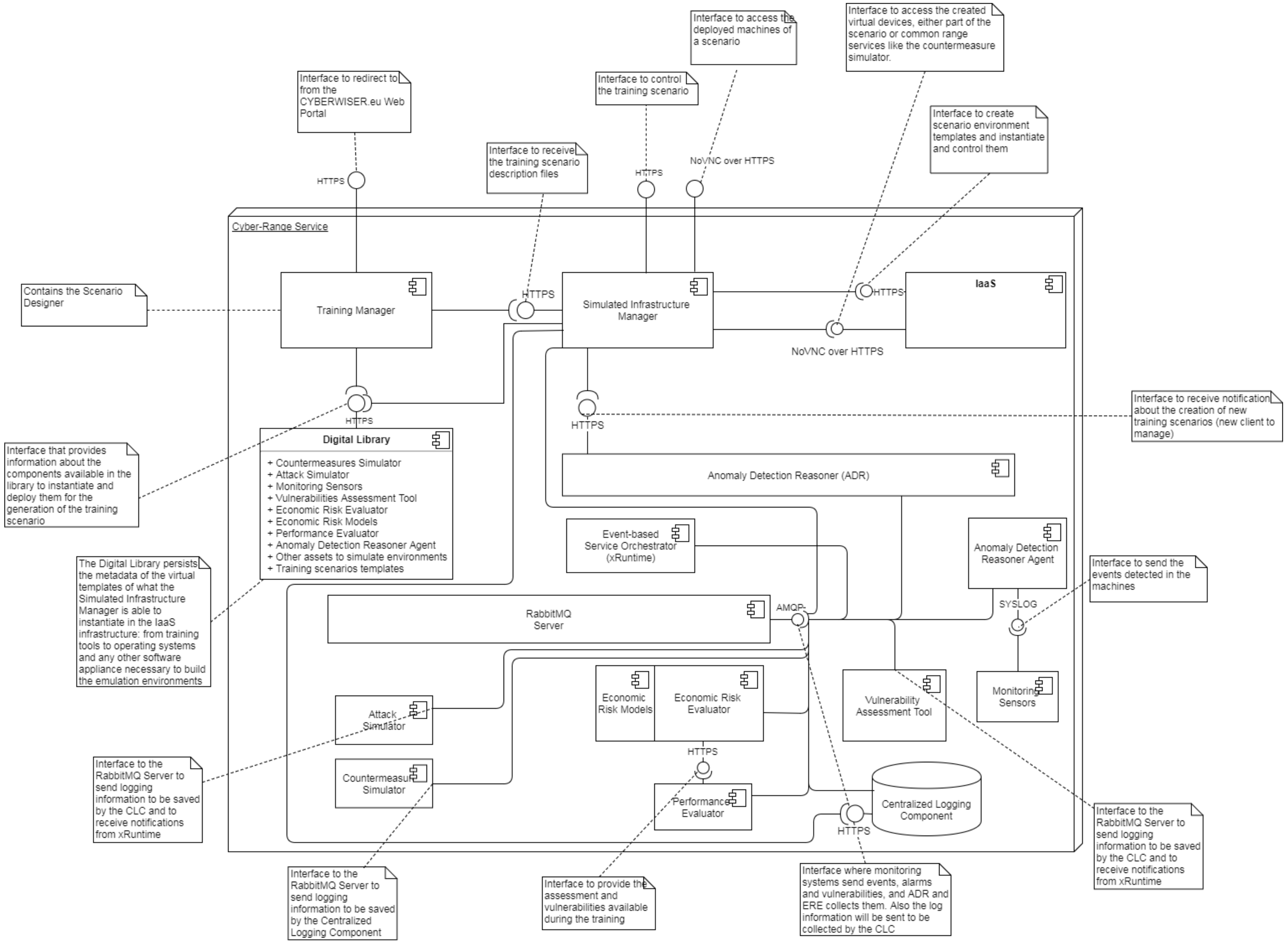
Interface to redirect to from the CYBERWISER.eu Web Portal
Interface to control the training scenario
Interface to access the deployed machines of a scenario
Interface to access the created virtual devices, either part of the scenario or common range services like the countermeasure simulator.
Interface to create scenario environment templates and instantiate and control them
Interface to receive the training scenario description files
HTTPS
HTTPS
NoVNC over HTTPS
Cyber-Range Service
Contains the Scenario Designer
Training Manager
HTTPS
Simulated Infrastructure Manager
HTTPS
IaaS
NoVNC over HTTPS
HTTPS
HTTPS
Interface to receive notification about the creation of new training scenarios (new client to manage)
Interface that provides information about the components available in the library to instantiate and deploy them for the generation of the training scenario
Digital Library
+ Countermeasures Simulator
+ Attack Simulator
+ Monitoring Sensors
+ Vulnerabilities Assessment Tool
+ Economic Risk Evaluator
+ Economic Risk Models
+ Performance Evaluator
+ Anomaly Detection Reasoner Agent
+ Other assets to simulate environments
+ Training scenarios templates
Anomaly Detection Reasoner (ADR)
The Digital Library persists the metadata of the virtual templates of what the Simulated Infrastructure Manager is able to instantiate in the IaaS infrastructure: from training tools to operating systems and any other software appliance necessary to build the emulation environments
Event-based Service Orchestrator (xRuntime)
Anomaly Detection Reasoner Agent
Interface to send the events detected in the machines
RabbitMQ Server
AMQP
SYSLOG
Economic Risk Models
Economic Risk Evaluator
Vulnerability Assessment Tool
Monitoring Sensors
Attack Simulator
Countermeasure Simulator
HTTPS
Performance Evaluator
Centralized Logging Component
HTTPS
Interface to the RabbitMQ Server to send logging information to be saved by the CLC and to receive notifications from xRuntime
Interface to the RabbitMQ Server to send logging information to be saved by the Centralized Logging Component
Interface to provide the assessment and vulnerabilities available during the training
Interface where monitoring systems send events, alarms and vulnerabilities, and ADR and ERE collects them. Also the log information will be sent to be collected by the CLC
Interface to the RabbitMQ Server to send logging information to be saved by the CLC and to receive notifications from xRuntime

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 RabbitMQ Server

The RabbitMQ server is a message broker application that exposes an Advanced Message Queuing Protocol (AMQP) interface[3] to exchange data from multiple sources. The main purpose of this interface is to support internal communications between the components in the scope of the training scenarios using the concepts of producers[4], messages[5], exchanges[6], queues[7] and consumers[8]. The interface will be used to share events, alarms or vulnerabilities detected in the virtual machines in the training scenario, to orchestrate specific components and to collect data logging coming from the components to be saved in the Centralized Login Component (CLC).

3.2.2 Centralized Logging Component (CLC)

The CLC component is a data store (it can be implemented using multiple tools or technologies) which will collect all data logging coming from different components and different training scenarios. The “Centralized” concept is not tied to the nature of the element itself, but how the component collects all data logging from all components and exercises. In the same way, the term “Logging” is applied because the main activity of the CLC is to act as a data logging storage. It will also be used for saving performance evaluation data in the form of reports which will be persisted along all the platform. The CLC is profitable for the Performance Evaluator (PE) (explained in section 3.2.16) to save evaluation reports, but also to provide platform-wide audit trail, which is very valuable for debugging purposes. The main source of input data will be the RabbitMQ component through the AMQP interface. The components will send the data logging to the RabbitMQ and the CLC will collect this data from the corresponding queues and will save it. The CLC will also provide a Representational State Transfer (REST) interface (HTTPS) to request the evaluation reports.

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 software 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.

[3] RabbitMQ in Action: Distributed Messaging for Everyone, Alvaro Videla and Jason J. W. Williams, Manning Publications
[4] Producer is an application that sends message
[5] Message is the data sent and received by the applications
[6] Exchange is an agent responsible for routing the messages coming from producers to their corresponding queues
[7] Queue is a buffer that stores a sequence of messages
[8] Consumer is an application that receives messages

The DL provides an HTTPS interface used by the Training Manager (TM) to prepare the training scenario requests to supply the templates for the set of virtual and physical elements and the training scenarios themselves. This interface is also consumed by the Simulated Infrastructure Manager (SIM) to consult the necessary metadata to translate the training scenario requests into training environments.

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 template. 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, and it allows the access to the machines and components available in the scenario environment. This component provides an HTTPS 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 HTTPS interfaces from the IaaS, which will also be used to control them. In addition, the SIM uses the NoVNC[9] 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 xRuntime (presented in 3.2.10) to orchestrate the VAT and the AS components via the RabbitMQ, and to trigger other events in the simulated environments according to timelines specified in the training scenario.
- The CS, via the RabbitMQ, to control 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 collected by the sensors deployed in the monitored simulated infrastructure coming from the ADR Agent and then sends them to the corresponding RabbitMQ exchange. It 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 RabbitMQ server, which is also employed to send the output events and alarms. The ADR will expose an HTTPS interface 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.

[9] NoVNC is a browser-based VNC client implemented using HTML5 Canvas and Web Sockets

3.2.8 Anomaly Detection Reasoner Agent (ADR Agent)

The ADR agent components collect the events coming from the monitoring sensors deployed into machines along the training scenarios. They provide a SYSLOG interface to receive the events and use the AMQP interface of the RabbitMQ server to resend them 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 events 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.

The monitoring sensors are addressed on detail on section 5.9

3.2.10 Event-based Service Orchestrator (xRuntime)

The xRuntime is a generic component that can orchestrate applications running in Docker containers. Its primary purpose is to support the business logic of VAT and AS components, however other components could be added as well. xRuntime communicates with managed applications (VAT, AS) via RabbitMQ. It receives messages from the RabbitMQ to control the component and, indirectly, control and modify the behaviour of the managed applications by triggering events.

3.2.11 Vulnerability Assessment Tool (VAT)

This component analyses and scans servers looking for security vulnerabilities. Its operation is orchestrated by the xRuntime component and uses the interface with the RabbitMQ server to send the resulting analysis reports that will be collected the ERE. VAT uses the RabbitMQ interface to send the reports and logging data that will be collected by the CLC.

3.2.12 Attack Simulator (AS)

The AS component provides to the users a series of predefined attack scripts or launches them automatically when there is no human red team. Its operation is orchestrated by the xRuntime component. AS uses the RabbitMQ interface to send the logging data that will be collected by the CLC.

3.2.13 Countermeasures Simulator (CS)

The CS component works similarly to the AS. It provides to the users a series of predefined countermeasures or applies them automatically when there is no human blue team. In this case, it can be employed, controlled and configured through the RabbitMQ connection. This application also uses the AMQP interface to send the data logging to the CLC.

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 RabbitMQ interface coming from the VAT application, while the source of the events and the alarms will be the ADR that also sends them to RabbitMQ. 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 along the training exercise which are retrieved through the HTTPS interface provided by the ERE component; and the logs coming from the different components and applications in the scenario to the RabbitMQ.

An overview of the connections among the components in the component diagram is shown in Table 2.

Required interface	Provided interface	Protocol
Web browser	CYBERWISER.eu Web Portal	HTTPS
Cross-Learning Facilities	Cyber-Range Service	SSO
Web browser	Training Manager	HTTPS
Web browser	Simulated Infrastructure Manager	HTTPS
Web browser	Simulated Infrastructure Manager	NoVNC over HTTPS
Training Manager	Simulated Infrastructure Manager	HTTPS
Training Manager	Digital Library	HTTPS
Simulated Infrastructure Manager	Digital Library	HTTPS
Simulated Infrastructure Manager	Infrastructure as a Service	HTTPS
Simulated Infrastructure Manager	Infrastructure as a Service	NoVNC over HTTPS
Simulated Infrastructure Manager	Anomaly Detection Reasoner	HTTPS
Simulated Infrastructure Manager	RabbitMQ	AMQP
Simulated Infrastructure Manager	Centralized Logging Component	HTTPS
Anomaly Detection Reasoner	RabbitMQ	AMQP
Event-based Service Orchestrator (xRuntime)	RabbitMQ	AMQP
Attack Simulator	RabbitMQ	AMQP
Vulnerability Assessment Tool	RabbitMQ	AMQP
Countermeasures Simulator	RabbitMQ	AMQP
Anomaly Detection Reasoner Agent	RabbitMQ	AMQP
Monitoring Sensors	Anomaly Detection Reasoner Agent	SYSLOG
Economic Risk Evaluator	RabbitMQ	AMQP
Performance Evaluator	Economic Risk Evaluator	HTTPS
Performance Evaluator	RabbitMQ	AMQP

Table 2: 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.

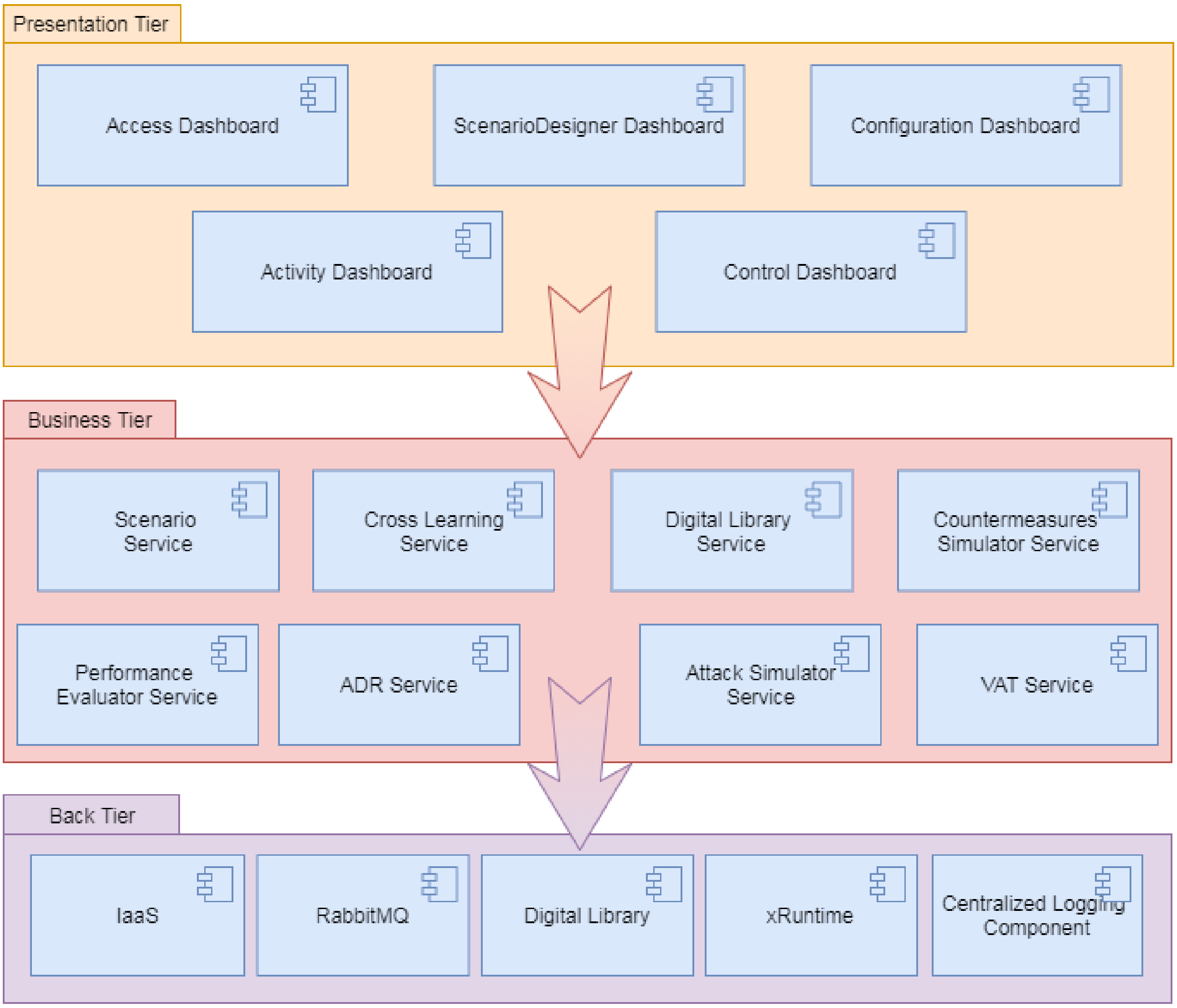

Figure 3. Three-tier software structure perspective

4.1.1 Presentation Tier

The presentation tier provides an interface to the end-users and presents the results to the browser (client) 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 3 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 RabbitMQ will facilitate the exchange of information and the CLC will provide persistent storage. The DL will provide the metadata for the components in the training scenarios. The xRuntime will orchestrate the AS and VAT components in the infrastructure.

4.2 Software deployment perspective

The CYBERWISER.eu Platform contains a lot of components that each need to be deployment in order to provide the overall functionality envisaged for the platform. 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.

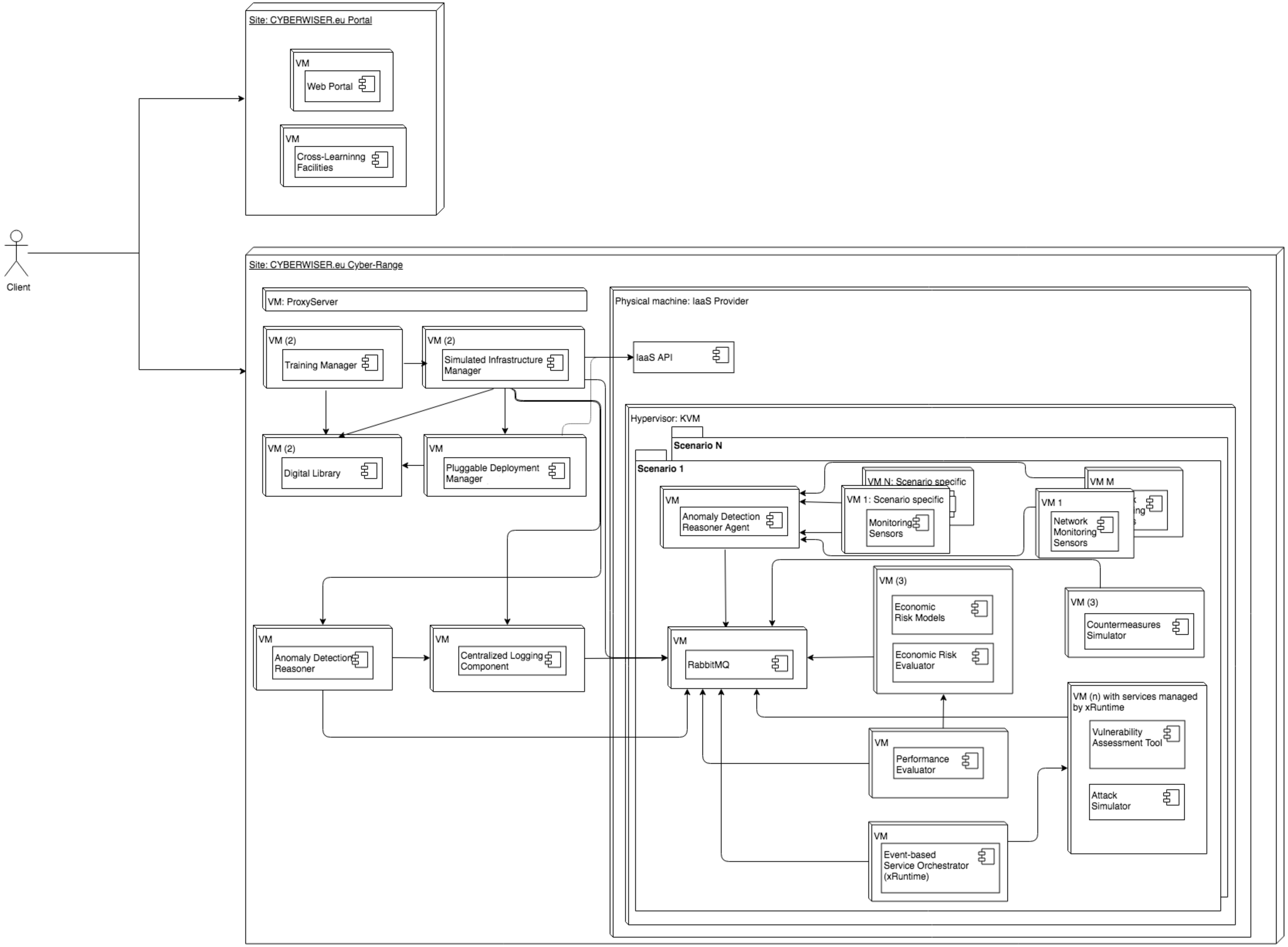
Site: CYBERWISER.eu Portal
VM
Web Portal
VM
Cross-Learninng Facilities
Client
Site: CYBERWISER.eu Cyber-Range
VM: ProxyServer
VM (2)
Training Manager
VM (2)
Simulated Infrastructure Manager
VM (2)
Digital Library
VM
Pluggable Deployment Manager
VM
Anomaly Detection Reasoner
VM
Centralized Logging Component
Physical machine: IaaS Provider
IaaS API
Hypervisor: KVM
Scenario N
Scenario 1
VM N: Scenario specific
VM 1: Scenario specific
Monitoring Sensors
VM
Anomaly Detection Reasoner Agent
VM M
VM 1
Network Monitoring Sensors
VM (3)
Economic Risk Models
Economic Risk Evaluator
VM (3)
Countermeasures Simulator
VM
RabbitMQ
VM (n) with services managed by xRuntime
Vulnerability Assessment Tool
Attack Simulator
VM
Performance Evaluator
VM
Event-based Service Orchestrator (xRuntime)

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 for use by the clients. These components are the TM, SIM, DL, Pluggable Deployment Manager (xOpera). They are deployed on one or more virtual machines. 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 RabbitMQ, ADR Agents, Monitoring Sensors, CS, VAT, PE and the Event-based Service Orchestrator (xRuntime). 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.

5. Detail of the building components

5.1 Web Portal

In its first release at M1, the website was designed on top of the already existing WISER project website as a gateway to updated web sections including News, Events, highlight of the CYERWISER.eu vision and partners.

The web portal hosted at www.cyberwiser.eu will evolve into a training hub designed for a variety of users, which will host the CYBERWISER.eu Platform and all training resources as a one-stop-shop.

The web portal will offer a unique single point of access to both trainers and trainees to the Cross-Learning Facilities that will host dedicated Workspace areas for specific users/teams.

The web portal will guarantee 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. The idea behind OpenID is to allow a user who has already registered on a major website, like Google, to log in to a Drupal site using his Google account information instead of creating a new account. OpenID supports several major website logins and its goal is facilitating users to login with universal credentials.

5.2 Cross-Learning Facilities

Cross-Learning Facilities will be fully integrated into the web portal.

By accessing the web portal, the user will be able to reach the Cross-Learning Facilities by means of a Workspace organized in different areas which will be defined by the matching of the user’s own profile and skills and the CYBERWISER.eu Offering levels.

Each area will be based upon Trust-IT’s Learning Content Management System (TRUST-LCMS) which will offer the possibility to enable tailor-made training courses to specific users/teams. According to the level of complexity of each training course, the Workspace area will guarantee the integration with specific components of the CYBERWISER.eu platform as depicted in Figure 5.

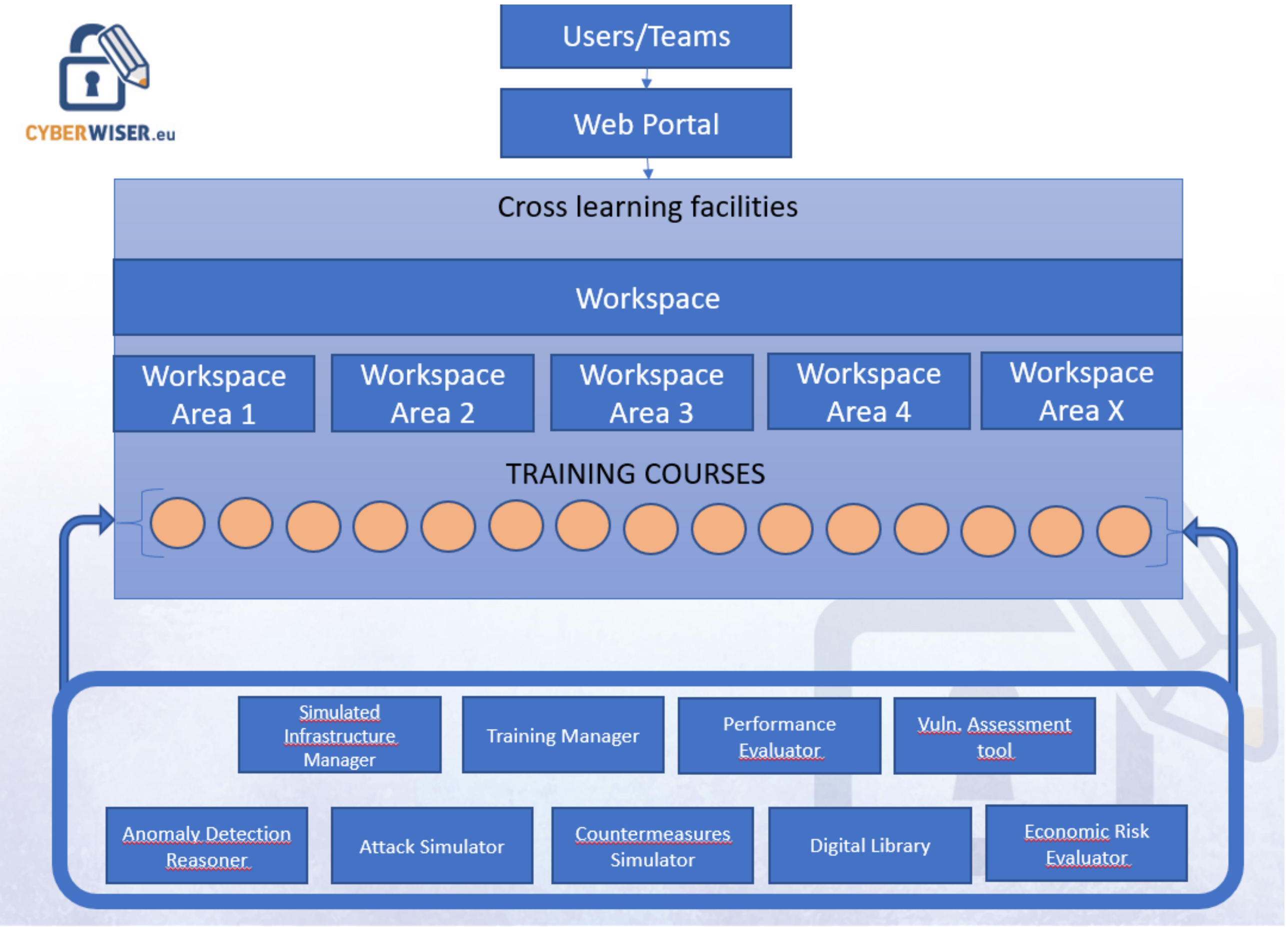

Figure 5 - Web Portal & Cross-Learning-Facilities high level view

Furthermore, for each of the training course, the LCMS will support the performance of theoretical validation of the competences acquired by the trainees with the awarding of a certification on competences acquired.

To enrich the user experience, a number of “badges” will be issued by the Workspace to those users that will demonstrate proficiency and dedication to the training path, in particular, “Pioneer badges”, “Best trainee” award and “Best student” contest will be handled as part of the user profile/credential.

In terms of innovation, the LCMS in CYBERWISER.eu will be able to handle the following new features:

- High integration with external training tools/modules, providing better user experience to the trainee.
- High interactivity behavior from the Platform depending on the training path demonstrated by the trainee (e.g. usage of interactive trees, integration in the tests of external resources).
- Personalized certificates for successful completion of the various levels of training in cybersecurity, including the steps envisaged for the specific level of training undertaken. Provisioning of smooth user experience with the “badges” (provided by the Workspace).

5.3 Simulated Infrastructure Manager

The SIM is responsible for converting a designed training scenario into a template the 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 6 shows its internal components.

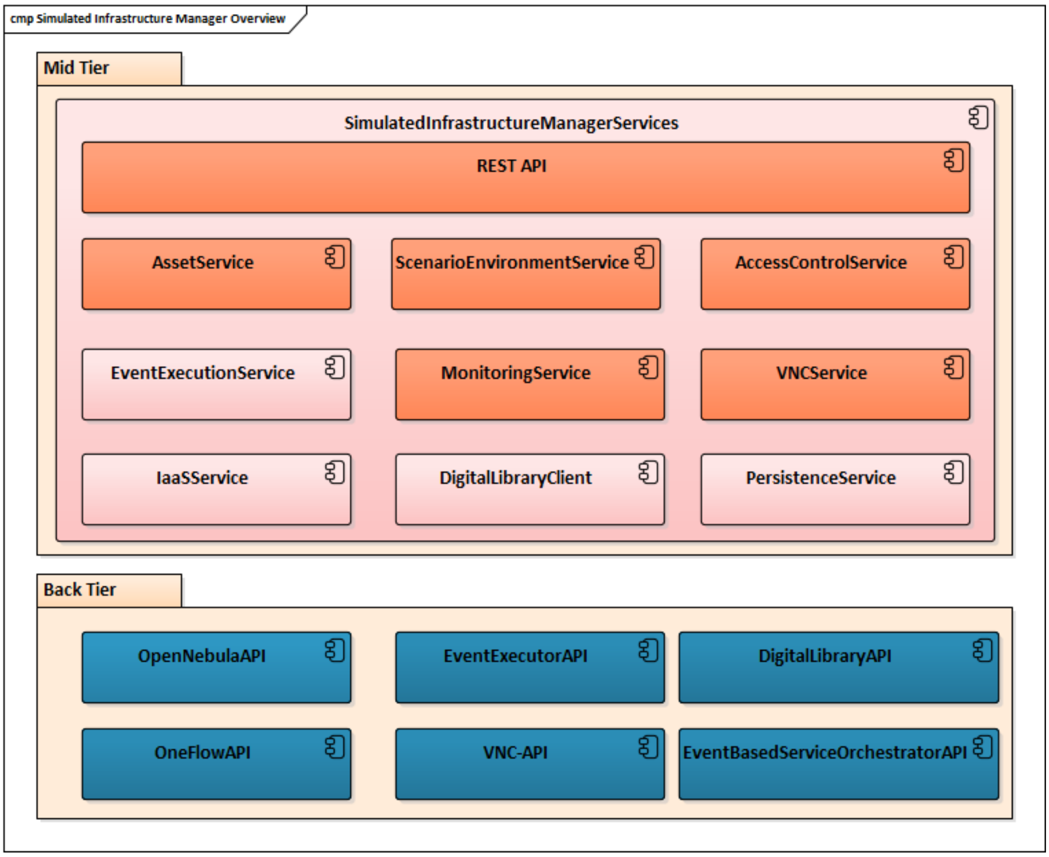

Figure 6. Simulated Infrastructure Manager components

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 Event-based Service orchestrator 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 Training Manager. Re-using the designed drawings of the scenario, functionality of the SIM is contextually shown in the user interface of the Training Manager.

The Simulated Infrastructure Manager 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 xOpera component is a deployment manager and orchestrator that can be leveraged to deploy simulated environments on various IaaS providers (for instance OpenStack and Amazon Web Services [AWS]) in an IaaS-provider-agnostic manner. As input, it expects a description of simulated environment in Topology and Orchestration Specification for Cloud Applications (TOSCA) standard. This description is obtained through the process of translation of training scenario requests. The xOpera component augments SIM’s native integration with OpenNebula. Logically, it covers functionalities of SIM’s ScenarioEnvironmentService and AssetService, and may be used as their alternative. Details of this component and its relationship to the existing services of the SIM will be refined in the next release of the document.

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 training manager 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 for determining 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.

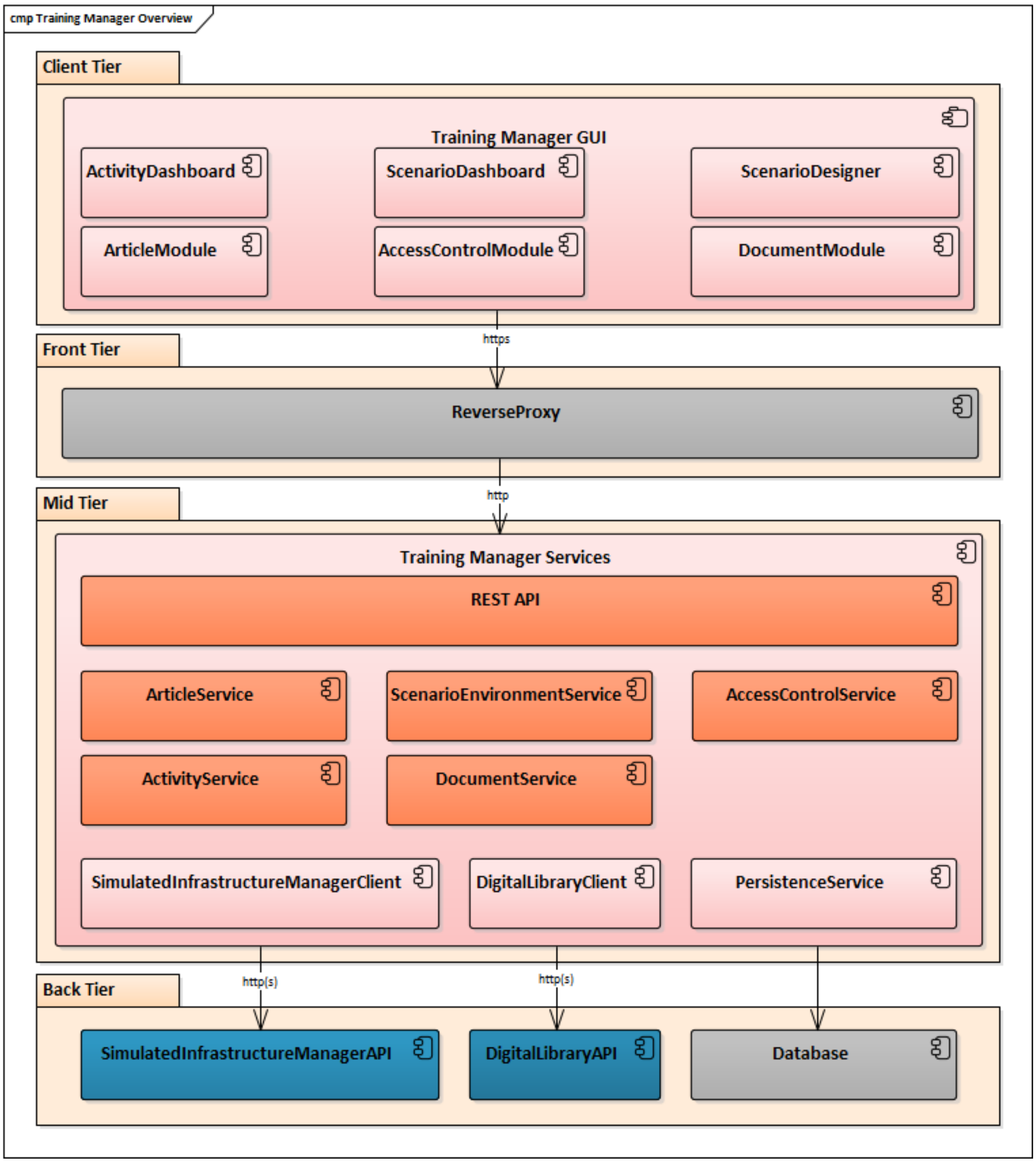

Figure 7. Training Manager components

The TM´s internal design (see Figure 7) 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 Performance Evaluator 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.

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.

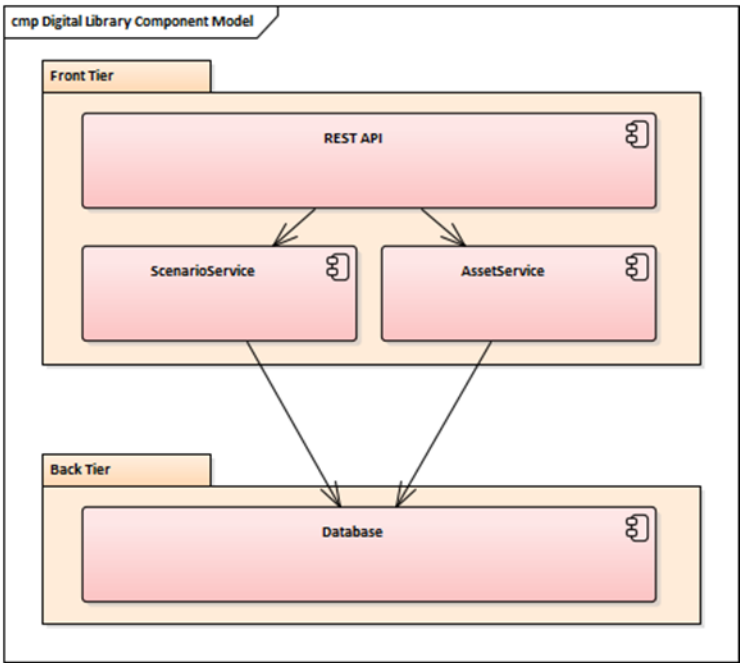

Figure 8. Digital Library Component Diagram

The above diagram (see Figure 8) shows the design of the Digital Library 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 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[10]. 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[11] values. These indicator values are generated from different sources of information:

- Questionnaire configuration 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, 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 RabbitMQ 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.

[10] H2020 Project WISER. Grant Agreement n. 653321. www.cyberwiser.eu
[11] The concept of indicator is envisioned and presented in the WISER project, in particular in deliverable D3.1: “Cyber risk patterns”

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

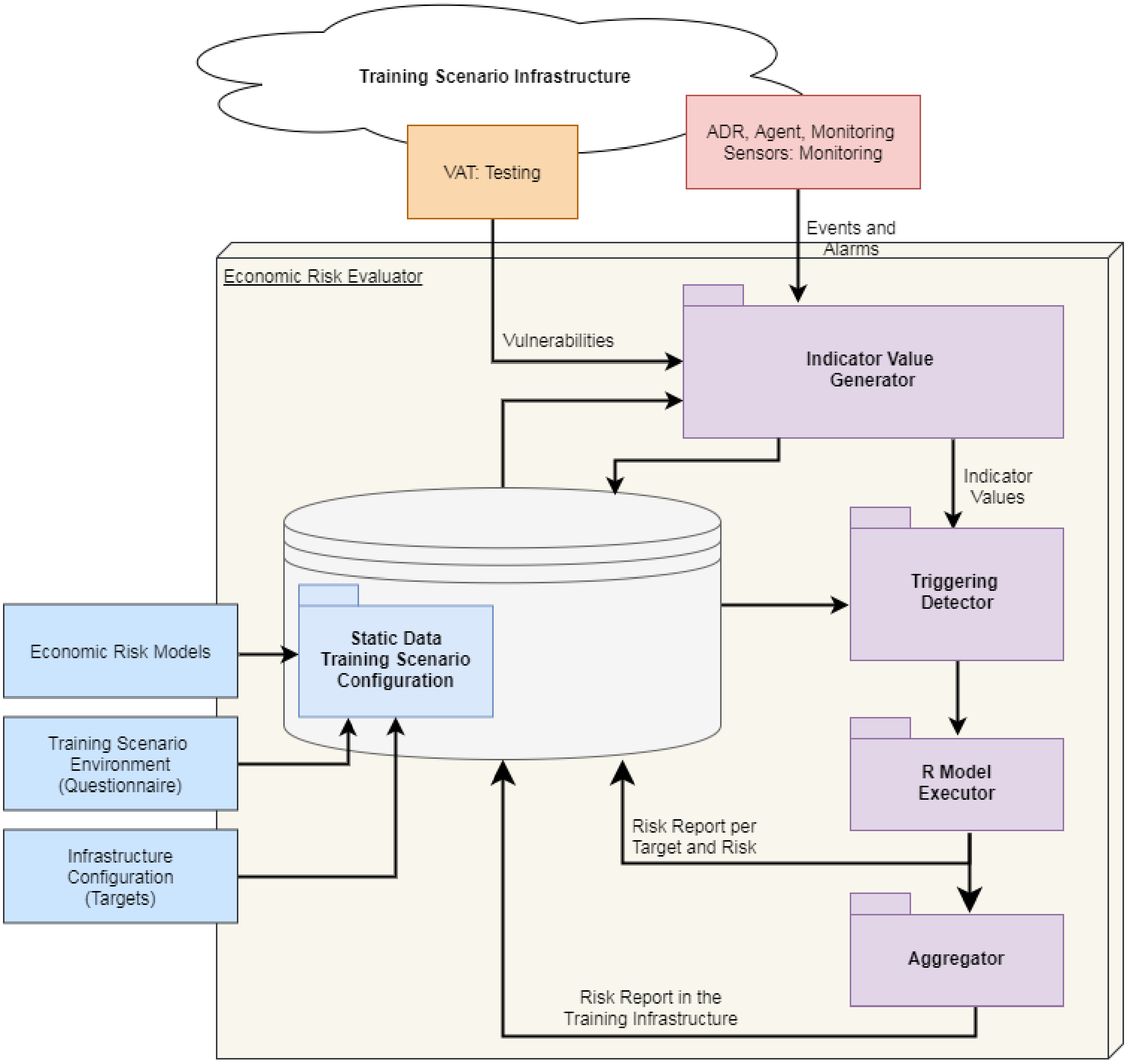
Training Scenario Infrastructure
VAT: Testing
ADR, Agent, Monitoring Sensors: Monitoring
Events and Alarms
Economic Risk Evaluator
Vulnerabilities
Indicator Value Generator
Indicator Values
Triggering Detector
Static Data Training Scenario Configuration
Economic Risk Models
Training Scenario Environment (Questionnaire)
Infrastructure Configuration (Targets)
R Model Executor
Risk Report per Target and Risk
Aggregator
Risk Report in the Training Infrastructure

Figure 9. Economic Risk Evaluator internal functional design[12]

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 involve in their corresponding 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

[12] This figure is an adaptation of that included in WISER deliverable D5.2 “WISER Real-time assessment infrastructure”

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 5.12.

ERE application consists of two tiers, the business logic and the back tier, which are represented in Figure 10. 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.

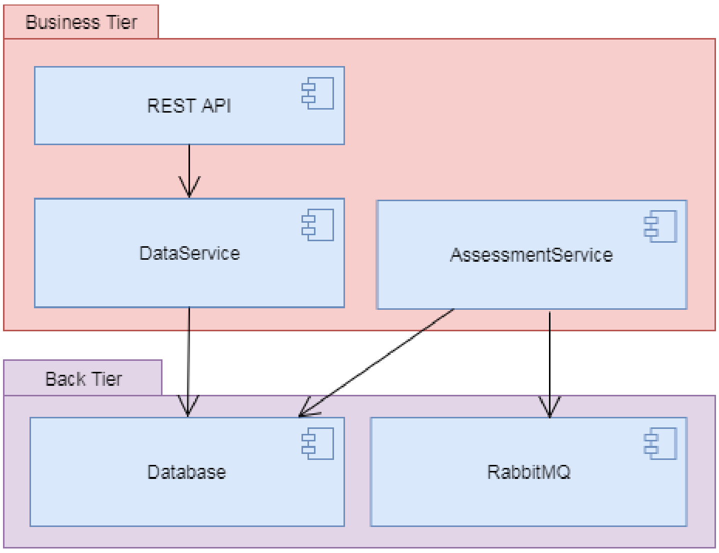

Figure 10. Economic Risk Evaluator layer design

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 risk models to establish and communicate understanding among (human) stakeholders with a wide range of different background; and b) corresponding machine-readable 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.

Graphical risk models are created using the CORAS risk modelling language[13] as well as the CORAS risk modelling tool adapted for WISER[14]. The risk assessment algorithms are developed schematically with respect to the graphical risk models mainly following guidelines also provided by WISER[10]. The initial version of guidelines for risk modelling and algorithm development in CYBERWISER.eu will be explained in D4.1 – training material, initial version. However, in the following we provide a high-level explanation of the modelling approach in CYBERWISER.eu and how produced models and corresponding algorithms are related to the ERE.

As illustrated in Figure 11, 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 in D4.1. In Step 1, the analyst needs to 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 threats, vulnerabilities, and threat scenarios. The first step produces graphical CORAS risk models with indicators which is used as input to Step 2. In Step 2, the analyst schematically translates the CORAS risk model with indicators to a machine-readable risk assessment algorithm expressed in the **R** programming language[15]. 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.

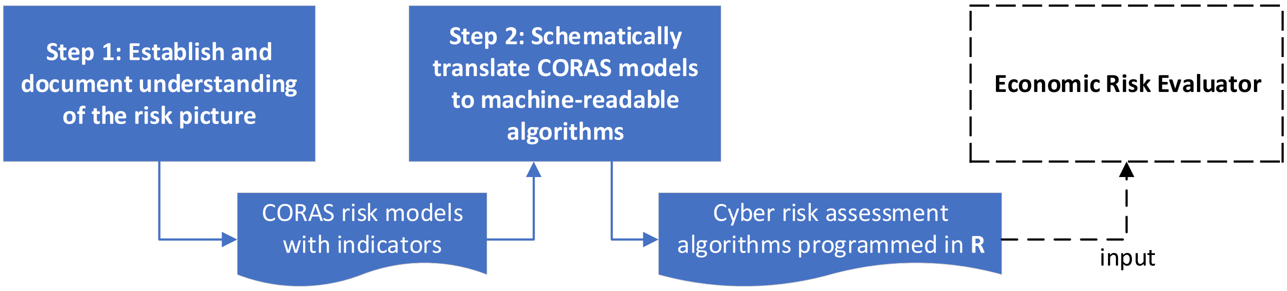
Step 1: Establish and document understanding of the risk picture
Step 2: Schematically translate CORAS models to machine-readable algorithms
Economic Risk Evaluator
CORAS risk models with indicators
Cyber risk assessment algorithms programmed in R
input

Figure 11 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 11 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 relevant indicators in the simulated infrastructure and feeding this information to the corresponding parameters in the R scripts.

As pointed out above, 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 by CYBERWISER.eu.
- Place the risk modelling ang calculation method in an overall risk analysis process. This includes the consideration of context establishment, risk assessment, and risk treatment.

[13] M. S. Lund, B. Solhaug and K. Stølen: Model-Driven Risk Analysis. The CORAS Approach. Springer, 2011.

[14] 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 (accessed 08.01.2019).

[15] The R Project for Statistical Computing. Online: https://www.r-project.org / (accessed 08.01.2019)

- Develop new risk models following the new overall risk analysis process.

5.8 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[16] and w3af[17] scanners for web vulnerability scanning). The integration of other modules (e.g. Nmap[18], OpenVAS[19], 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.

   > Generic scanning can be used *automatically by the platform* (to aid subsequent economic risk assessment and trainee evaluation), *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 detection modules.** To detect the presence of specific vulnerabilities deliberately introduced in the exercise infrastructure as a part of a training scenario, custom detection modules that check for the presence of such vulnerabilities can be implemented. Custom detection modules 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 us to assess trainees’ progress during the exercise.

   > Custom vulnerability detectors can be used *automatically by the platform* (to aid in trainee evaluation) and *manually by the white team* to help keep track of trainees´ progress during the exercise.

VAT support 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 exercise execution.

The output of VAT is a *vulnerability report* in a format understandable by other parts of the CYBERWISER.eu Platform. Vulnerability report is provided to the ERE, where it is considered in the risk assessment, indirectly affecting the trainees’ performance evaluation. The vulnerability report could also be presented to the trainees (both red and blue teams) to help them guide their attack or defense actions.

The internal component diagram of VAT is presented on Figure 12. Its subcomponents and their interactions are detailed below.

.

[16] https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
[17] http://w3af.org/
[18] https://nmap.org/
[19] http://www.openvas.org/

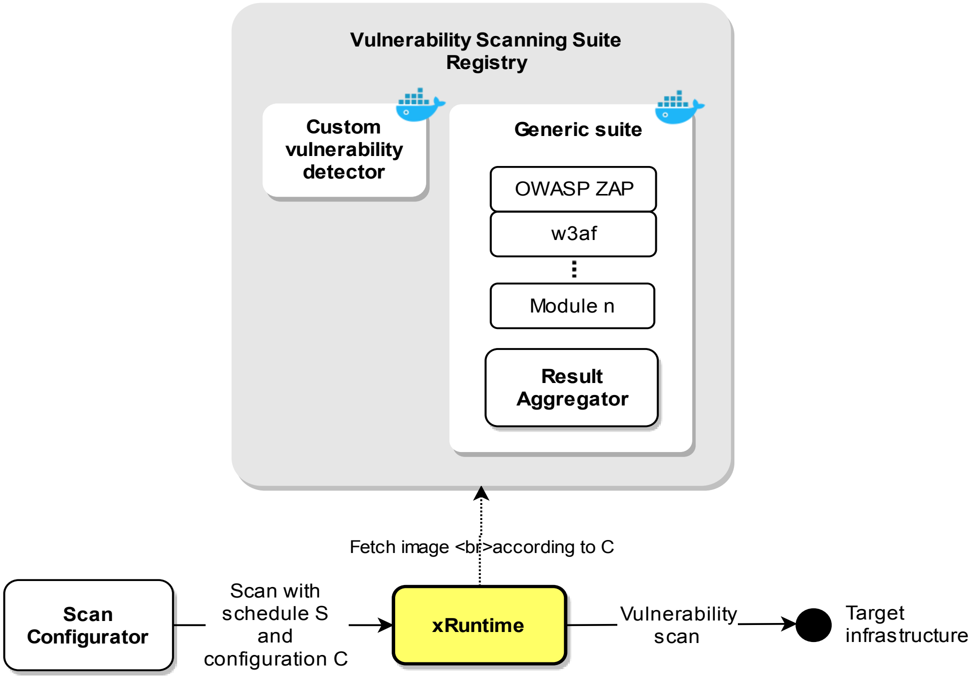
Vulnerability Scanning Suite Registry
Custom vulnerability detector
Generic suite
OWASP ZAP
w3af
Module n
Result Aggregator
Fetch image
according to C
Scan Configurator
Scan with schedule S and configuration C
xRuntime
Vulnerability scan
Target infrastructure

Figure 12. Logical view of vulnerability assessment tools

Scan Configurator handles all aspects of vulnerability scanning configuration, specifically:

- target of the scan (by means of an URL or an IP address),
- which suite/scanner to run (a generic suite or some custom scanning module),
- suite/module-specific configuration (if any), and
- scheduling of scans.

Scan Configurator has a front-end part with the Web interface to be utilized either at the time of scenario design or by trainers and/or trainees during an ongoing cyber-range exercise.

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

Custom Vulnerability Detectors are simple scripts/programs performing custom automated checks (vulnerability scans). Note that in contrast to the Generic Suite, which wraps various vulnerability scanners, each Custom Vulnerability Detector is packaged as a self-contained Docker image. Consequently, a single execution of a Custom Vulnerability Detector always corresponds to a single vulnerability scan.

Vulnerability Scanning Suite Registry is a collection of (generic and custom) vulnerability scanners. It is implemented as a Docker registry[20] that keeps one Docker image for the Generic Suite and one image per Custom Vulnerability Detector.

xRuntime is a generic component that schedules and orchestrates vulnerability scanning modules. The same component is also used in the scope of AS and it is additionally described in Section 3.2.10. xRuntime receives the schedule and configuration of scanning tasks from the Scan Configurator and executes them. In a context of a single scanning task, either a custom vulnerability scanning module or the Generic Suite of scanners is

[20] https://docs.docker.com/registry/

run with a configuration that describes which modules to execute (in the case of Generic Suite only) and which target should be scanned. xRuntime instantiates Docker containers from the corresponding Docker images. The Docker containers can be seen as short-lived workers whose workloads are single launches of a vulnerability scan. Their output is collected by xRuntime and forwarded to the CLC from where it becomes available to other components.

Regarding the innovation roadmap envisioned for this component, the Generic Suite, 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 addressed in CYBERWISER.eu for VAT comprise changing the interfaces of the Generic Suite of scanner to fit into the described architecture, and integrating it with xRuntime to manage scheduling more efficiently and generically (xRuntime is also used by AS). Apart from the integration-related changes, new functionalities will be added in the form of Custom Vulnerability Detectors and possibilities will be studied to include new types of generic scanning modules into the Generic Suite.

5.9 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, presented in the following. Logs are collected by means of a *Rsyslog*[21] 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 Anomaly Detection Reasoner is represented in Figure 13.

[21] Rsyslog website: https://www.rsyslog.com/ (last accessed on 12/02/2019)

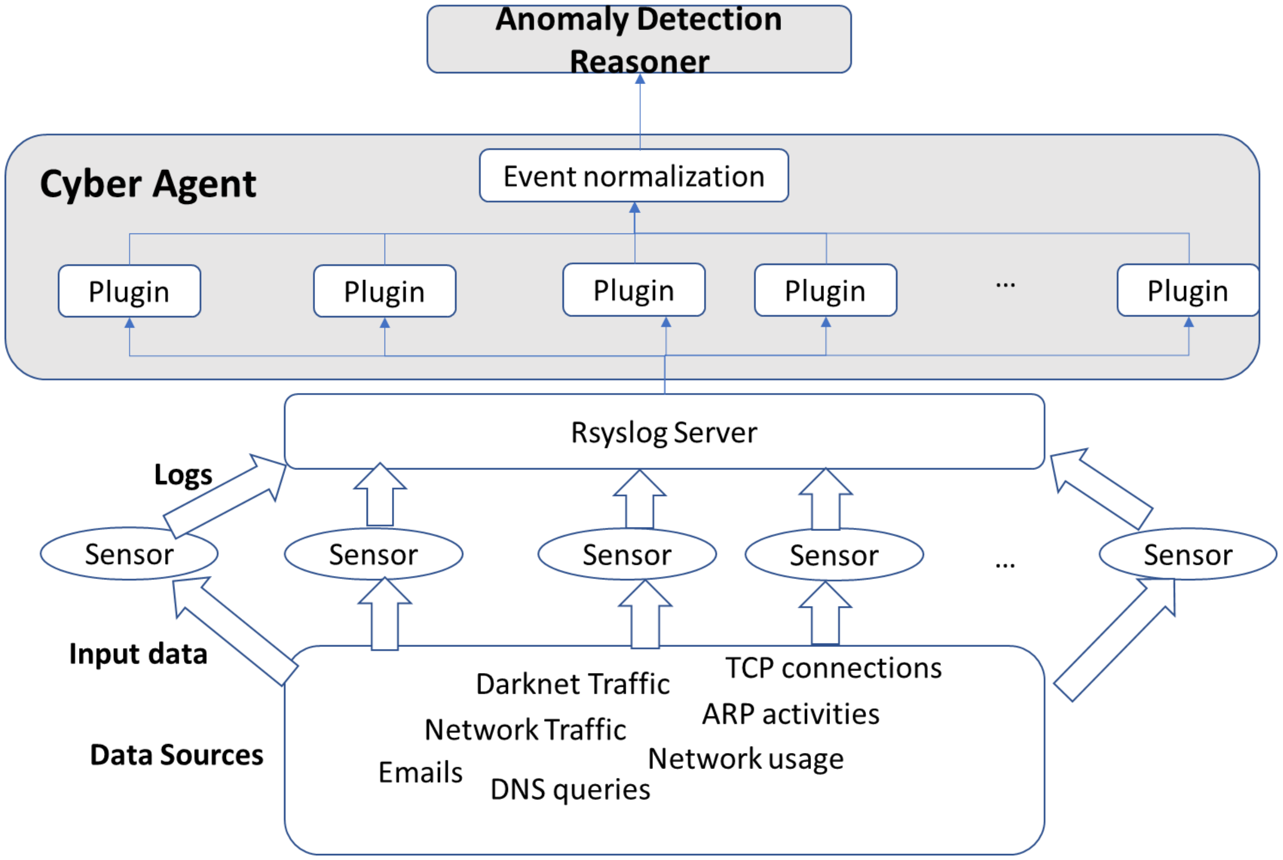

Figure 13. Monitoring sensors and Cyber Agent

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[22]: 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[23] signature language[24]. Suricata detects complex threats, supports multi-thread and multi-processor systems and inspects huge amounts of traffic data (gigabits).
- OSSEC[25]: 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[26].
- 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[27].
- 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

[22] https://suricata-ids.org/ Last accessed 15/01/2019
[23] https://www.lua.org/about.html. Last accessed 15/01/2019
[24] 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
[25] http://www.ossec.net. Last accessed 15/01/2019
[26] WISER Project Deliverable D4.1: Design of the WISER Monitoring Infrastructure
[27] WISER Project Deliverable D4.2: WISER Monitoring Infrastructure

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[23]

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.

5.10 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[28].

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 14 shows the internal component diagram of the ADR.

[28] https://www.alienvault.com/products/ossim (Last accessed on 12/02/2019)

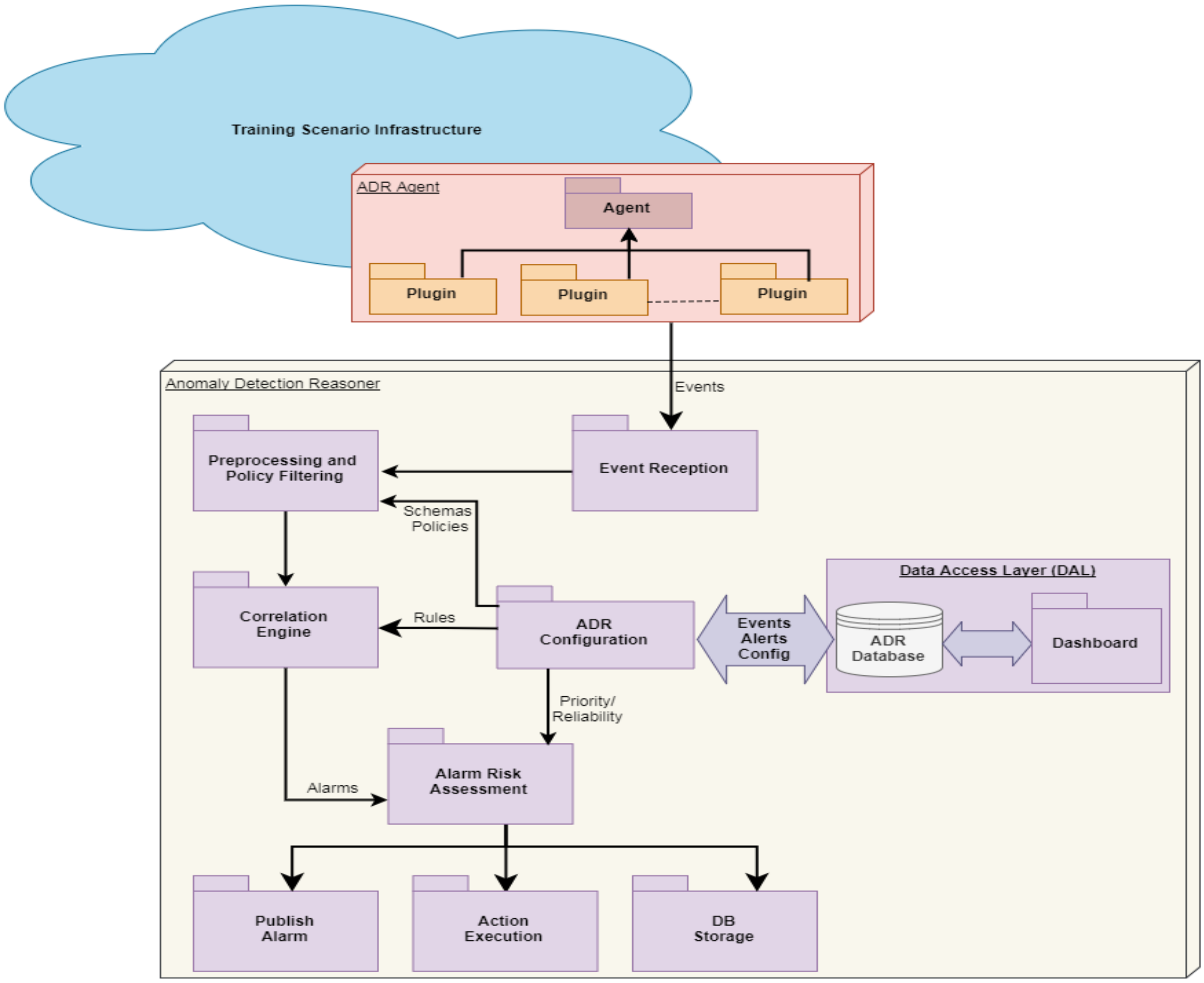

Figure 14. Anomaly Detection Reasoner Diagram[29]

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 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 optimization of the use of available resources[30]

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

[29] This diagram is an adaptation of that included in CIPSEC Deliverable D2.2: "CIPSEC Unified Architecture First Internal Release"

[30] 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 16/01/2019)

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 15), the presentation tier, the business tier and the back tier, which is depicted in Figure 10. 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 14. The presentation tier provides the GUI for the end-users.

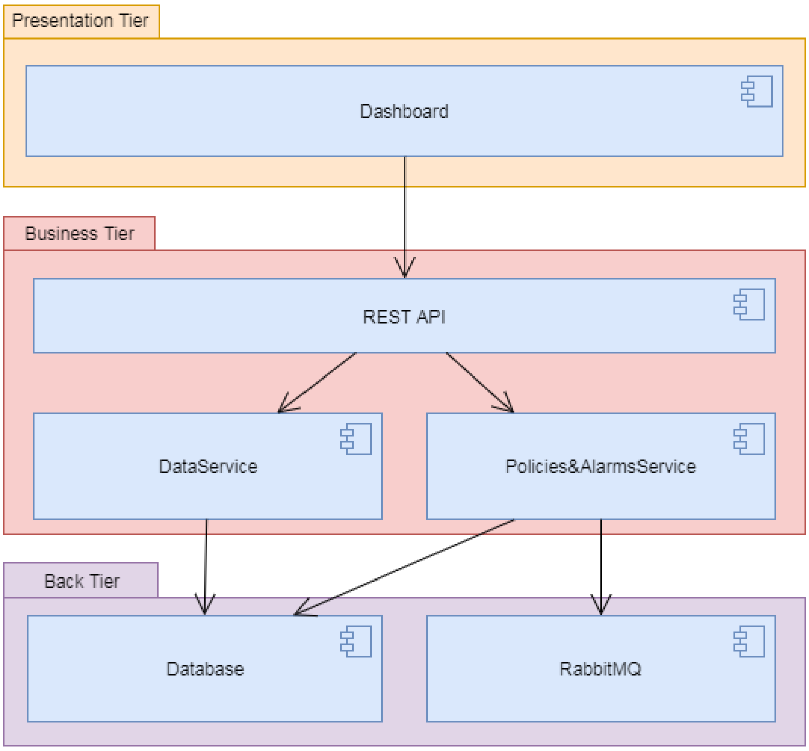

Figure 15. Anomaly Detection Reasoner layer design

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.

5.11 Attack Simulator

At the high level, AS supports the following functionalities:

1. **Running pre-defined automated attacks**. As a part of scenario design via the Web interface, 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 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 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 decides to increase it to allow smoother progression of the exercise. He might even modify the attack script itself. As with automated attacks, the trainer (or another white team member acting on trainer’s behalf) controls the AS through the Web interface.
3. **Easier launching of automated attacks in the 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, it is also paramount that they understand the specifics of (possibly complicated) attacks, even without deeper knowledge of the software itself. AS supports the second use case, as it allows a red team trainee to configure and launch attack scripts via the Web interface.
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* (e.g. available by default either across the CYBERWISER.eu Platform or for a given training scenario) and *shared* (contributed by other trainees) attack scripts or attack script templates. These can be used as a starting ground for configuring attacks, to avoid writing everything from scratch.

Figure 16 depicts 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.

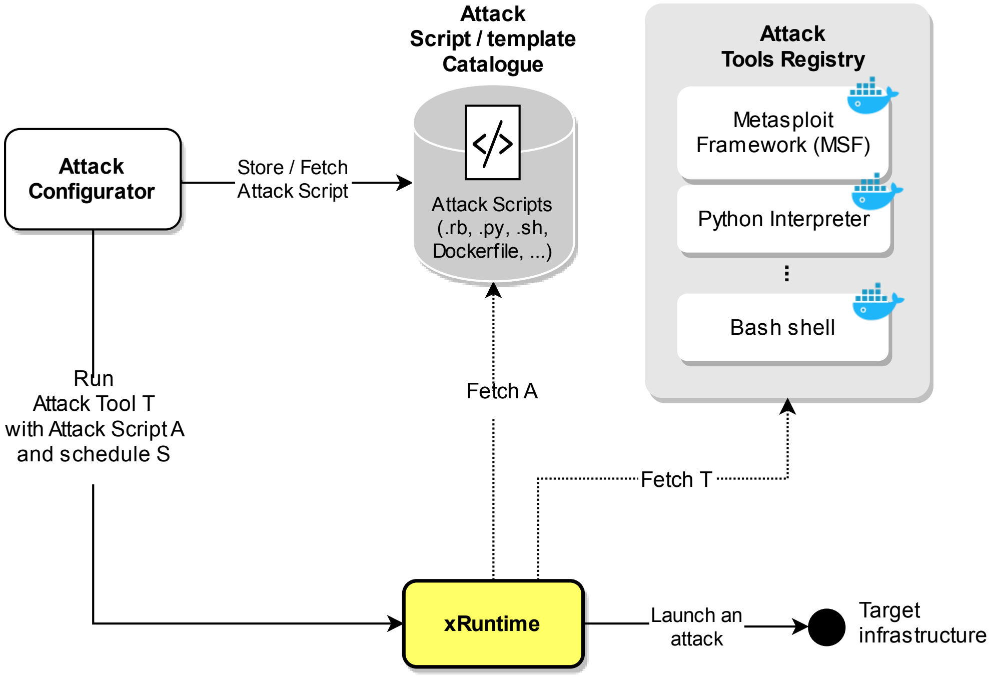
Attack Script / template Catalogue
Attack Tools Registry
Attack Configurator
Store / Fetch Attack Script
Attack Scripts (.rb, .py, .sh, Dockerfile, ...)
Metasploit Framework (MSF)
Python Interpreter
⋮
Bash shell
Run Attack Tool T with Attack Script A and schedule S
Fetch A
Fetch T
xRuntime
Launch an attack
Target infrastructure

Figure 16. Logical view of attack simulator

Attack Script is a 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 Python, 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 true Attack Scripts.

Attack Script / Template Catalogue is a storage component where attack scripts and attack script templates are stored. It exposes a simple interface for storage and retrieval of attack scripts based on access control policies (RBAC). It is envisioned that the DL will cover some of these aspects (for instance global persistence of pre-defined scripts, to be included at the time of scenario design), however further details related to persistence (related to shared scripts and the script sharing in general) which affect the complexity of the subcomponent are yet to be defined.

Attack Configurator is an interface with the catalogue of attack scripts and attack script templates. It has a front-end part with the Web interface to be utilized either at the time of scenario design or directly by trainers and trainees during scenario runtime. Attack Configurator allows configuration of all aspects of attacks, including attack scripts to run, and their scheduling.

Attack Tools are Docker images describing environments for running a particular kind of attack script. In general, attack tools will be based on Linux images and will package environments necessary to run the associated attack script type, if necessary, as an input to a particular offensive security/pentesting tool. 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).

Attack Tools are meant to be relatively *lightweight* Docker images. They are indirectly (via Attack Configurator) available both to red team trainees during the exercise as well as to the CYBERWISER.eu Platform, so that it can launch scheduled attacks (as dictated by the training scenario definition). 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 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 Attack Simulator (via Attack Configurator), while others require hands-on knowledge of offensive security tools available from the workstation.

Attack Tools Registry is a Docker registry[20] that stores all the Attack Tools in a single location.

xRuntime is a generic component at the heart of the AS (please refer to Section 3.2.10 for details). In the context of AS, xRuntime oversees that attack scripts are launched against the target infrastructure according to a given timeline (based on configuration coming from a message queue via Attack Configurator or potentially some other CYBERWISER.eu component). It instantiates Docker containers from the given Attack Tools (Docker images). These Docker containers can be seen as short-lived workers whose workloads are single launches of an attack script. In addition to orchestrating Docker containers, xRuntime handles logging of events related to the CLC.

5.12 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).

This tool is based on the assumption that there is a set of data that describes the exercise and are stored in a central database. 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.

These assumptions made, the logical sequence (see Figure 17) this tool would follow is:

1) The ADR sends through the RabbitMQ 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 scenario element in question.
5) The user runs the chosen mitigation, and in the backend a script will be executed. These scripts will be fed the needed information leveraging the existing data about the scenario. This will be automatic if the blue team role is played by the platform.

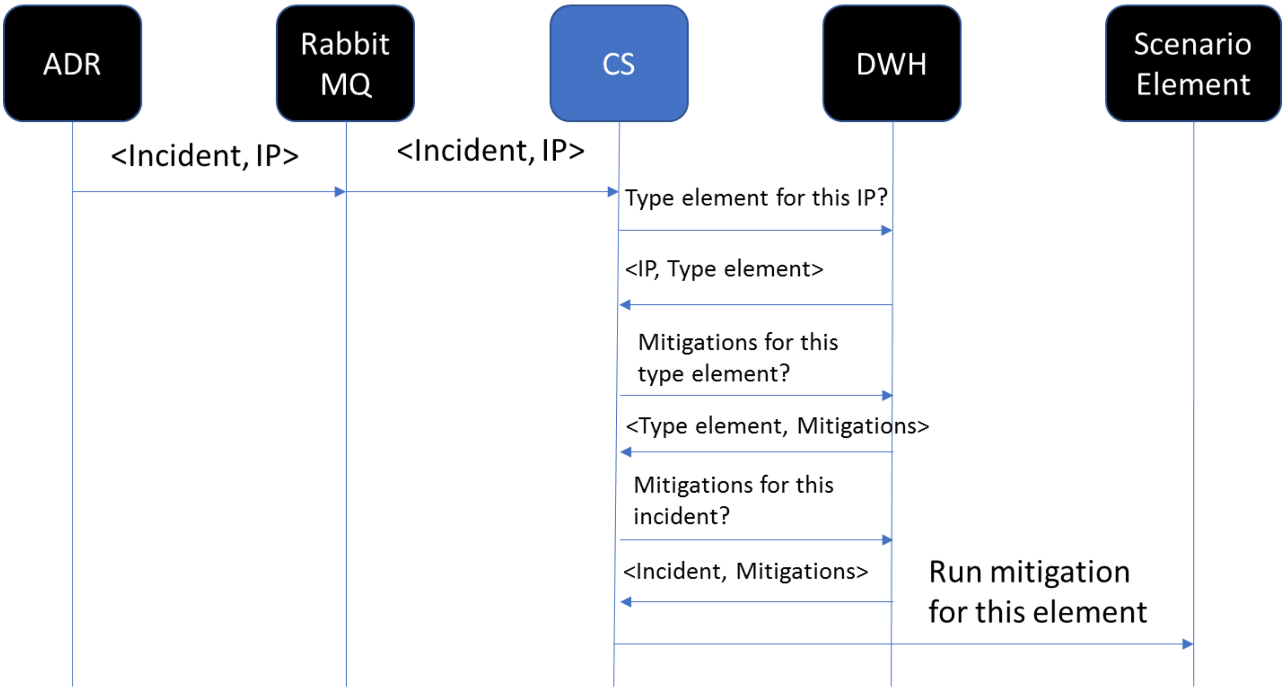
ADR
Rabbit MQ
CS
DWH
Scenario Element
<Incident, IP>
<Incident, IP>
Type element for this IP?
<IP, Type element>
Mitigations for this type element?
<Type element, Mitigations>
Mitigations for this incident?
<Incident, Mitigations>
Run mitigation for this element

Figure 17. Concept of Countermeasures Simulator

Not being the simulator itself, the platform will offer two more utilities related to the countermeasure simulation:

- The ERE (section 5.7) will present not only the risk status, but also some proposed mitigations that are triggered within the model algorithm itself depending on the cyber climate status (this is, the indicator values). The ERE will offer the user the possibility to mark a specific measure as applied to simulate how this would affect the economic risk evaluation. Typically, the application of a specific mitigation affects the value of some input indicators, which affects the computation of the risk, diminishing its value. In other words, this means forcing the indicators to simulate what would happen if the mitigation is applied.

- The Cross-Learning facilities (section 5.2) will compile the information about the different mitigations that can be applied. For each mitigation the user will learn what the mitigation is about, what implies applying it, which emergency situations it applies to and how to check that the mitigation has been correctly applied, among other aspects. This will be offered as part of the learning materials existing in CYBERWISER.eu.

6. Impact of the business requirements on the design

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 Section 1.

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 3, 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 3.

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

ASSET / PLATFORM Version	PRIMER	BASIC	INTERMEDIATE	ADVANCED
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 intances
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 3: 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).

6.1 Primer Version Deployment Diagram

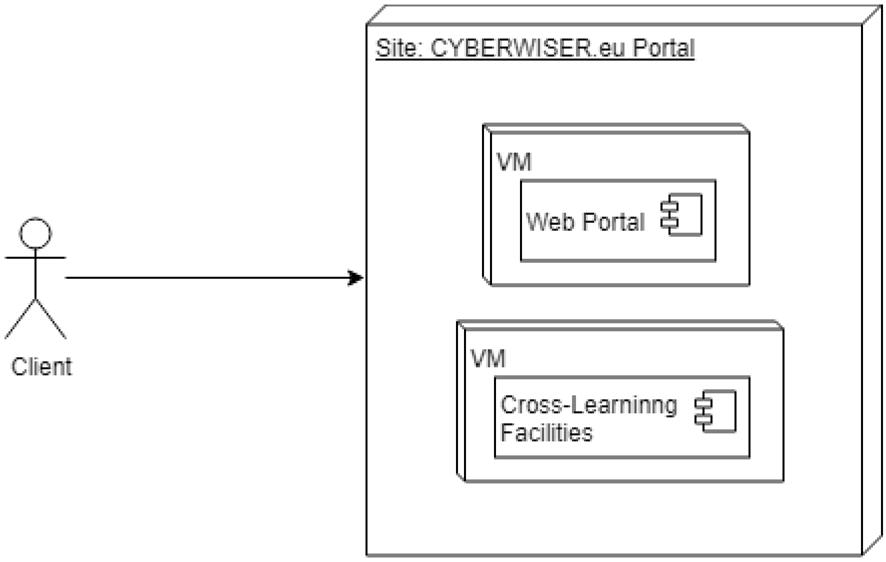

Figure 18: CYBERWISER.eu Primer version deployment diagram

The Primer version considers a very simple deployment (see Figure 18) 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.

6.2 Basic Version Deployment Diagram

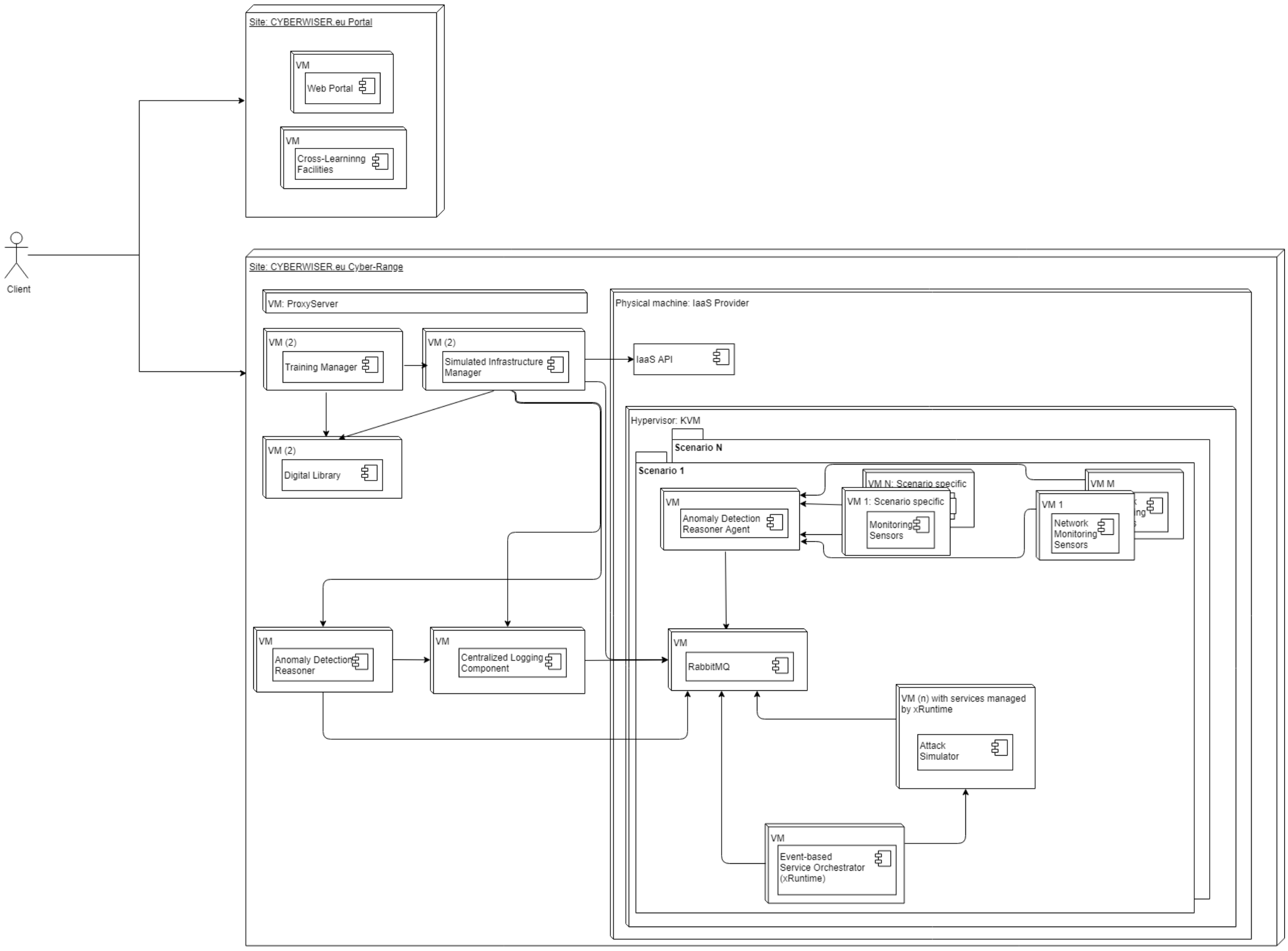
Site: CYBERWISER.eu Portal
VM
Web Portal
VM
Cross-Learninng Facilities
Client
Site: CYBERWISER.eu Cyber-Range
VM: ProxyServer
VM (2)
Training Manager
VM (2)
Simulated Infrastructure Manager
VM (2)
Digital Library
VM
Anomaly Detection Reasoner
VM
Centralized Logging Component
Physical machine: IaaS Provider
IaaS API
Hypervisor: KVM
Scenario N
Scenario 1
VM
Anomaly Detection Reasoner Agent
VM N: Scenario specific
VM 1: Scenario specific
Monitoring Sensors
VM M
VM 1
Network Monitoring Sensors
VM
RabbitMQ
VM (n) with services managed by xRuntime
Attack Simulator
VM
Event-based Service Orchestrator (xRuntime)

Figure 19: 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.

6.3 Intermediate Version Deployment Diagram

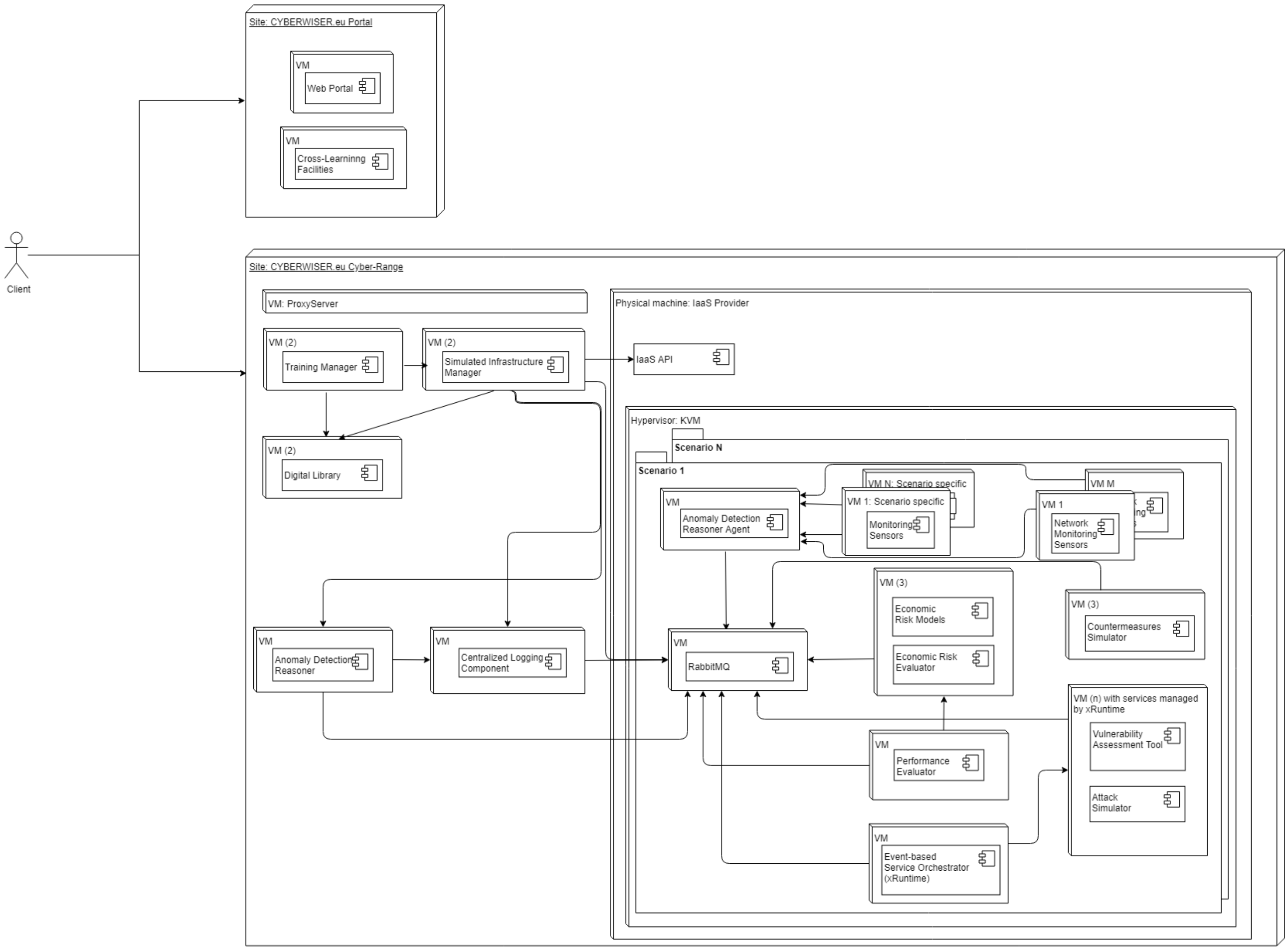
Site: CYBERWISER.eu Portal
VM
Web Portal
VM
Cross-Learninng Facilities
Client
Site: CYBERWISER.eu Cyber-Range
VM: ProxyServer
VM (2)
Training Manager
VM (2)
Simulated Infrastructure Manager
VM (2)
Digital Library
VM
Anomaly Detection Reasoner
VM
Centralized Logging Component
Physical machine: IaaS Provider
IaaS API
Hypervisor: KVM
Scenario N
Scenario 1
VM
Anomaly Detection Reasoner Agent
VM N: Scenario specific
VM 1: Scenario specific
Monitoring Sensors
VM M
VM 1
Network Monitoring Sensors
VM
RabbitMQ
VM (3)
Economic Risk Models
Economic Risk Evaluator
VM (3)
Countermeasures Simulator
VM (n) with services managed by xRuntime
Vulnerability Assessment Tool
Attack Simulator
VM
Performance Evaluator
VM
Event-based Service Orchestrator (xRuntime)

Figure 20: 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).

7. Initial requirements traceability

Within Section 2.3, 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 D2.1 [1] 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 D2.1 [1]. In the final version of the design (D2.4) the Matrices will be updated considering the final version of the requirements in 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 4 to Table 9.

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 include Workspace Areas to enable users to exchange messages, manage documents, directly access "external" services as training and eLearning sections.
FUNC-6	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST support multiple simultaneous instantiations of training scenarios.
FUNC-7	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-8	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST support training of individual trainees.
FUNC-9	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.

ID	Type	Traced Asset	Description
FUNC-10	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.
FUNC-11	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-12	Functional	CYBERWISER.eu Platform	Training scenarios MUST be dynamically reconfigurable during their execution, in order to support on-the-fly adjustments of scope and difficulty.
FUNC-13	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform COULD allow trainees to suspend ongoing training scenarios and resume them later on.
FUNC-14	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-15	Functional	CYBERWISER.eu Platform	In a given training scenario instance, The CYBERWISER.eu Platform MUST differentiate between various end-user types (for instance trainer, trainee), by assigning each user an appropriate role.
FUNC-16	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform COULD allow streaming of on-going training sessions (or selected fractions of them).
FUNC-17	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform SHOULD allow to create nicknames for the trainees.
FUNC-18	Functional	CYBERWISER.eu Platform	With the nickname, The CYBERWISER.eu Platform SHOULD show a performance rating (e.g. number of participated training sessions, time vs number of attacks, time for each attack…).
FUNC-19	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform MUST integrate learning materials supporting entry-level training.
FUNC-20	Functional	CYBERWISER.eu Platform	The CYBERWISER.eu Platform SHOULD allow a customized 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-21	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 has to

ID	Type	Traced Asset	Description
			be anonymised and filtered considering sensitive information.
FUNC-22	Functional	Cross-Learning Facilities	It MUST be possible to associate at least one cyber-range training scenario with a particular online course.
FUNC-23	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-24	Functional	Cross-Learning Facilities	An online course MUST allow the exchange of files between trainees and trainers.
FUNC-25	Functional	Cross-Learning Facilities	An online course MUST trace the point of the course actually reached by a student following the course.
FUNC-26	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-27	Functional	Cross-Learning Facilities	An online course MUST provide a final certificate to the students who completed the entire course.
FUNC-28	Functional	Cross-Learning Facilities	The Workspace Area MUST provide the capability to create/delete folders, upload/download files respecting the user profile.
FUNC-29	Functional	Training Manager	The Scenario Designer MUST support the design of the ICT topology of the simulated environments leveraging the training scenario requests.
FUNC-30	Functional	Training Manager	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 request. The dashboard will present: · Network diagram area, · Network taxonomy objects toolbar, · Wizard object characterization tabbed box.
FUNC-31	Functional	Training Manager	The Scenario Designer MUST support the design of the application topology of the simulated environments leveraging the training scenario requests.
FUNC-32	Functional	Training Manager	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: · Application diagram area, · Application taxonomy objects toolbar,

ID	Type	Traced Asset	Description
			· Wizard object characterization tabbed box.
FUNC-33	Functional	Training Manager	The Scenario Designer MUST support the definition of scenario events and events triggers for the simulated environments leveraging the training scenario requests.
FUNC-34	Functional	Training Manager	The Scenario Designer MUST support the definition of the monetary value of identified elements in the simulated environments leveraging the training scenario requests.
FUNC-35	Functional	Training Manager	The Scenario Designer MUST support the definition of the instructions for the trainees involved in the training scenario via a training design dashboard.
FUNC-36	Functional	Training Manager	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: · Training diagram area, · Training taxonomy object toolbar, · Wizard object characterization tabbed box.
FUNC-37	Functional	Training Manager	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-38	Functional	Training Manager	The Scenario Designer MUST support training scenario requests versioning.
FUNC-39	Functional	Training Manager	The Scenario Designer MUST allow to define training scenarios with limited time duration.
FUNC-40	Functional	Training Manager	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-41	Functional	Training Manager	The Scenario Designer MUST be able to leverage on the catalogue of pre-defined set of training scenarios provided by the Digital Library.
FUNC-42	Functional	Training Manager	The Scenario Designer SHOULD support the design of the business topology of the simulated environments leveraging the training scenario requests.
FUNC-43	Functional	Training Manager	The Scenario Designer SHOULD have a Business Design Dashboard where the user can draw the underlying business processes of a training scenario request. This dashboard will present:

ID	Type	Traced Asset	Description
			· Business Processes diagram area, · Business Processes taxonomy objects toolbar, · Wizard object characterization tabbed box.
FUNC-44	Functional	Training Manager	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-45	Functional	Training Manager	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-46	Functional	Training Manager	The Scenario Designer MUST have a dashboard where relevant information related to the training scenario requests will be summarized: · Unique ID, · Creation date/time, · Status (Design in Progress, Waiting, Validated, Instantiated), · Description and/or motivation.
FUNC-47	Functional	Training Manager	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: · ICT Topology and its assets' attributes, · Application Topology and its assets' attributes, · Business Topology and its assets' attributes, · Description and/or motivation.
FUNC-48	Functional	Training Manager	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-49	Functional	Training Manager	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-50	Functional	Training Manager	The Scenario Designer MUST, after pre-validation, send the training scenario requests to the Simulated Infrastructure Manager.
FUNC-51	Functional	Performance Evaluator	The input to the trainee's performance evaluation MUST be based on the cyber-range exercises carried out by the trainee.

ID	Type	Traced Asset	Description
FUNC-52	Functional	Performance Evaluator	The CYBERWISER.eu Platform 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-53	Functional	Performance Evaluator	Performance evaluation MUST be based on clear criteria for evaluating the trainee.
FUNC-54	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-55	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-56	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-57	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-58	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-59	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-60	Functional	Training Manager	When a trainee concludes the cyber-range exercise, the CYBERWISER.eu Platform MUST show evaluation criteria and achieved scores to both the trainee and the trainer.
FUNC-61	Functional	Training Manager	The Training Manager MUST encompass an activity area where the information regarding each activity will be aggregated.
FUNC-62	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-63	Functional	Training Manager	The Training Manager MUST allow the users to create/edit/delete activities.
FUNC-64	Functional	Training Manager	The Training Manager MUST allow the users to create/edit/delete training scenarios within an activity.
FUNC-65	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-66	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-67	Functional	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST provide the capability to validate the simulation environment templates leveraging the training scenario requests.
FUNC-68	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-69	Functional	Simulated Infrastructure Manager	Authorized operators MUST be able to command the Simulated Infrastructure Manager to instantiate/deinstantiate/delete virtual environment templates leveraging the training scenario requests.
FUNC-70	Functional	Simulated Infrastructure Manager	Authorized operators MUST be able to Start a training scenario instance instantiated in the cyber-range infrastructure.
FUNC-71	Functional	Simulated Infrastructure Manager	Authorized operators MUST be able to Stop a training scenario instance instantiated in the cyber-range infrastructure.
FUNC-72	Functional	Simulated Infrastructure Manager	Authorized operators 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-73	Functional	Simulated Infrastructure Manager	Authorized operators MUST be able to open remote sessions (if available) with any involved Virtual Machine in the active (Started) instantiated training scenario instance.
FUNC-74	Functional	Simulated Infrastructure Manager	Authorized operators MUST be able to manually command training scenario events and event triggers, during training scenario instance execution.

ID	Type	Traced Asset	Description
FUNC-75	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-76	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-77	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-78	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-79	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-80	Functional	Simulated Infrastructure Manager	Authorized users MUST be able to check the status and availability of the resources of the cyber-range infrastructure.
FUNC-81	Functional	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST be able to handle training scenarios with limited time duration.
FUNC-82	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-83	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 likelihood of the risk as well as the impact of the risk in terms of monetary loss.
FUNC-84	Functional	Economic Risk Models	The risk models MUST reflect risks the individual client organisation and system are exposed to.
FUNC-85	Functional	Economic Risk Models	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-86	Functional	Economic Risk Models	The training material on development and selection of risk models provided by The

ID	Type	Traced Asset	Description
			CYBERWISER.eu Platform MUST be aimed at different levels of trainee skills (basic and advanced).
FUNC-87	Functional	Economic Risk Evaluator	Each automatic mitigation action suggested by the CYBERWISER.eu Platform to the blue team MUST be associated with a certain cost.
FUNC-88	Functional	Economic Risk Evaluator	It MUST be possible to assign values to organization's virtual resources (simulated in the training scenarios), reflecting their monetary values.
FUNC-89	Functional	Economic Risk Evaluator	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-90	Functional	Economic Risk Evaluator	Economic Risk Evaluator MUST inform about the economic risks during the execution of the exercises in real-time.
FUNC-91	Functional	Economic Risk Evaluator	Economic Risk Evaluator MUST offer a flexible interface to execute different risk models.
FUNC-92	Functional	Economic Risk Evaluator	Economic Risk Evaluator MUST generate risk reports and store them.
FUNC-93	Functional	Economic Risk Evaluator	Economic Risk Evaluator MUST show both economic exposure and possible mitigation actions.
FUNC-94	Functional	Anomaly Detection Reasoner	Based on the recorded events from the training scenario, it MUST be possible to detect anomalous activity in the cyber-range.
FUNC-95	Functional	Anomaly Detection Reasoner	Anomaly Detection Reasoner MUST support several types of sensors used in the training scenarios.
FUNC-96	Functional	Anomaly Detection Reasoner	Anomaly Detection Reasoner MUST process the information collected by the sensors deployed in the monitored simulated infrastructure.
FUNC-97	Functional	Anomaly Detection Reasoner	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-98	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-99	Functional	Vulnerability Assessment Tools	Vulnerability Assessment Tools SHOULD be able to automatically execute network reconnaissance procedures towards the exercise infrastructure.

ID	Type	Traced Asset	Description
FUNC-100	Functional	Attack Simulator	Red team MUST be able to implement new attack scripts during an exercise.
FUNC-101	Functional	Attack Simulator	Trainees SHOULD be able to share attack scripts with other users.
FUNC-102	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 attacks.
FUNC-103	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 modify the existing ones with the specifics pertaining a particular training scenario.
FUNC-104	Functional	Attack Simulator	During the cyber-range exercise, red team MUST have access to appropriate offensive security & penetration testing tools in order to carry out attacks on organization's simulated assets.
FUNC-105	Functional	Attack Simulator	Attack Simulator MUST be able to execute attack scripts provided in the training scenario description, according to the specified schedule.
FUNC-106	Functional	Attack Simulator	The attack simulation scripts and their schedules MUST be dynamically adaptable during training scenario execution.
FUNC-107	Functional	Countermeasures Simulator	The CYBERWISER.eu Platform 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-108	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-109	Functional	Countermeasures Simulator	When a specific mitigation action selected by the blue team would take too long to execute in real-time, the CYBERWISER.eu Platform SHOULD be capable of simulating it.
FUNC-110	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-111	Functional	Countermeasures Simulator	Blue team players MUST be offered a series of pre-defined countermeasures in the shape of templates/scripts or any other formats, which

ID	Type	Traced Asset	Description
			they can use as tools to protect the infrastructure.
FUNC-112	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-113	Functional	Countermeasures Simulator	Blue team MUST be able to implement new countermeasure scripts during an exercise.
FUNC-114	Functional	Countermeasures Simulator	Trainees SHOULD be able to share attack scripts with other users.
FUNC-115	Functional	Countermeasures Simulator	Blue team MUST be able to search for specific countermeasures within the catalogue basing on keywords.
FUNC-116	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.
FUNC-117	Functional	Countermeasures Simulator	The countermeasures and their schedules MUST be dynamically adaptable by the white team during the scenario execution.
FUNC-118	Functional	Countermeasures Simulator	The white team MUST be able to add new countermeasures dynamically during the execution of the training.

Table 4: Functional Requirements design traceability

ID	Type	Traced Asset	Description
T-PLAT-1	Platform	CYBERWISER.eu Platform	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-PLAT-2	Platform	CYBERWISER.eu Platform	Each Workspace Area MUST have at least one administrator.
T-PLAT-3	Platform	CYBERWISER.eu Platform	All relevant managed events that occur in a cyber-range exercise (for instance attack triggered, attack detected, configuration changed, progress status changed) MUST be logged.
T-PLAT-4	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-5	Platform	Scenario Designer	The Scenario Designer SHOULD be a sub-component of the Training Manager.

ID	Type	Traced Asset	Description
T-PLAT-6	Platform	Scenario Designer	The Scenario Designer MUST be able to outline the training scenario requests in a human-readable format.
T-PLAT-7	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-8	Platform	Scenario Designer	The Scenario Designer MUST implement an interface with the Simulated Infrastructure Manager.
T-PLAT-9	Platform	Scenario Designer	The Scenario Designer MUST implement an interface with the Digital Library.
T-PLAT-10	Platform	Scenario Designer	The Scenario Designer's underlying Data Model MUST be in common with the Simulated Infrastructure Manager and the Digital Library.
T-PLAT-11	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-12	Platform	Simulated Infrastructure Manager	The training scenario requests generated by the Scenario Designer MUST be translatable to simulated environment templates for the underlying Simulated Infrastructure Manager.
T-PLAT-13	Platform	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST implement an interface with the Scenario Designer.
T-PLAT-14	Platform	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST implement an interface with the Digital Library.
T-PLAT-15	Platform	Simulated Infrastructure Manager	The Simulated Infrastructure Manager MUST restore its state with the last persisted configuration after crash or reboot of its server-side application.
T-PLAT-16	Platform	Simulated Infrastructure Manager	Data Model used by the Simulated Infrastructure Manager MUST be the same as the Data Model of the Simulated Infrastructure Manager and the Digital Library.
T-PLAT-17	Platform	Anomaly Detection Reasoner	Anomaly Detection Reasoner MUST report and store in a central exercise log the activation/deactivation of every anomaly during the execution of the training scenarios.
T-PLAT-18	Platform	Anomaly Detection Reasoner	Anomaly Detection Reasoner SHOULD send the information about events and alarms to a queue, enabling the subsequent storage of the information in a central exercise log.
T-PLAT-19	Platform	Vulnerability Assessment Tools	Vulnerability Assessment Tools MUST produce results (list of detected vulnerabilities) that can be used as input for the Economic Risk Evaluator.
T-PLAT-20	Platform	Vulnerability Assessment Tools	Providing Vulnerability Assessment Tools to trainees of both blue and red team MUST be supported.

ID	Type	Traced Asset	Description
T-PLAT-21	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-22	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-23	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-24	Platform	Monitoring Sensors	All sensors MUST continuously monitor simulated ICT resources in accordance with the training scenario definition.
T-PLAT-25	Platform	Monitoring Sensors	Monitoring data MUST be streamed to Anomaly Detection Reasoner for further analysis.
T-PLAT-26	Platform	Attack Simulator	Attack Simulator SHOULD report every attack execution and its success status to an exercise log.
T-PLAT-27	Platform	Countermeasures Simulator	Countermeasures simulator SHOULD report every countermeasure execution to an exercise log.

Table 5: Platform Requirements design traceability

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 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 computational resources of the cyber-range infrastructure with respect to the amount of training scenario instances.
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.

ID	Type	Traced Asset	Description
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	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 6: Performance Requirements design traceability

ID	Type	Traced Asset	Description
T-SECU-1	Security	CYBERWISER.eu Platform	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-2	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-3	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-4	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-5	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-6	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-7	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-8	Security	CYBERWISER.eu Platform	Workspace Areas MUST provide access reserved to a-priori-identified user groups, accessing a pre-determined (and configurable) set of services.

ID	Type	Traced Asset	Description
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 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	Simulated Infrastructure Manager	Instances of different training scenarios simultaneously running in the cyber-range MUST be isolated (from a networking point of view) from each other.
T-SECU-13	Security	Simulated Infrastructure Manager	Server-side Simulated Infrastructure Manager application MUST 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-14	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-15	Security	Anomaly Detection Reasoner	The confidentiality and integrity of the information communicated to/from the Anomaly Detection Reasoner MUST be guaranteed.
T-SECU-16	Security	Anomaly Detection Reasoner	Anomaly Detection Reasoner MUST not be visible for the red team.

Table 7: Security Requirements design traceability

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.

ID	Type	Traced Asset	Description
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 MUST provide a user-friendly graphical user interface to both Platform operators and trainees.
T-USAB-7	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-8	Usability	CYBERWISER.eu Web Portal	The CYBERWISER.eu Web Portal SHOULD allow customization of the dashboard.
T-USAB-9	Usability	Scenario Designer	The Scenario Designer SHOULD provide a user-friendly environment for the definition and configuration of training scenario requests.
T-USAB-10	Usability	Scenario Designer	All drawing dashboards of the Scenario Designer MUST allow to zoom in/out on the drawing canvas.
T-USAB-11	Usability	Scenario Designer	All drawing dashboards of the Scenario Designer MUST allow to group elements on the drawing canvas.
T-USAB-12	Usability	Simulated Infrastructure Manager	The log of exercise events SHOULD support filtering of events (e.g. by type, time ...).

Table 8: Usability Requirements design traceability

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 anonymized.
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.
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 9: Legal Requirements design traceability

As a conclusion, it is easy to notice that all requirements from D2.1 [1] can be traced to specific CYBERWISER.eu components or to the overall Platform.

8. Conclusions

This deliverable reports the current status of the design process of the CYBERWISER.eu Platform. This process has already covered its first stage, in which the results of the requirements elicitation activity have been the main source of information to carry out the design. In the second stage, the main sources of information will be the requirements set by the pilots and the first results of the implementation and integration activities, that will incorporate refinements and nuances to the design. In this second stage the Consortium will dig into the different design aspects considered in the first stage and will incorporate new views to contribute to a comprehensive understanding of the platform. The ultimate objective is to have a solid design that effectively drives the implementation and integration activities. The traceability with the requirements will be ensured as well as the alignment with the market needs.

References

[1] CYBERWISER.eu Project. D2.1 Requirements initial version. November 2018

[2] CYBERWISER.eu Project. D6.1 Communication & stakeholder plan first version. December 2018