

**Project Acronym:** STAR  
**Grant Agreement number:** 956573 (H2020-ICT-2020-1 – Research and Innovation Action)  
**Project Full Title:** Safe and Trusted Human Centric Artificial Intelligence in Future Manufacturing Lines  
**Project Coordinator:** INTRASOFT International



Funded by the Horizon 2020  
Framework Programme of the  
European Union

## DELIVERABLE

### D3.6 – Security and Data Governance Infrastructure-Final version

<b>Dissemination level</b>	PU -Public
<b>Type of Document</b>	Report
<b>Contractual date of delivery</b>	30/06/2023
<b>Deliverable Leader</b>	GFT
<b>Status - version, date</b>	Final – v1.0, 04/09/2023
<b>WP / Task responsible</b>	WP3
<b>Keywords:</b>	Threats Security Vulnerability Risk Scenarios Attacks

*This document is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 956573. It is the property of the STAR consortium and shall not be distributed or reproduced without the formal approval of the STAR Management Committee. The content of this report reflects only the authors' view. The European Commission is not responsible for any use that may be made of the information it contains.*

## Executive Summary

STAR project describes and implements a platform for safe data interchange across the STAR infrastructure's components. The transferred data may have a significant impact on the platform's operation; hence it must be safeguarded from tampering efforts.

This deliverable is the description of the architecture of the STAR AI Security and Data protection layer, which is an outcome of the collaborative actions of all partners of WP3. This deliverable is the description of the architecture of the STAR AI Security and Data protection layer, which is an outcome of the collaborative actions of all partners of WP3. This document is built on the previous WP3 deliverable (D3.5) which is the preliminary version of the Security and Data Governance Infrastructure. The current deliverable provides all the results of WP3 partner's efforts to deliver the final version of the several components of the protection layer. D3.6 can be read as a comprehensive document of the WP3 results, while D3.5 can be read to understand the initial development process.

Overall, this layer brings together the AI security and data governance techniques of the STAR project, which are destined to protect AI systems from poisoning and evasion attacks, while boosting the reliability of industrial data through blockchain-based data provenance mechanisms. The layer is complemented by the risk assessment and attack mitigation functionalities which are provisioned by the synergistic operation of the STAR Security Policies manager (provided by GFT) and UBITECH's OLISTIC risk management engine. In addition, the security layer of STAR adopts runtime monitoring mechanisms, able to monitor critical devices and collect important measurements that can reveal the behavioural profile of the production lines. The functionalities of this layer aim to secure existing digital manufacturing platforms and devices that comprise AI systems in the manufacturing line of the STAR demonstrator, while also supporting the safety, reliability and transparency functionalities of the upper layer of STAR that aims to augment the business operation of the STAR pilot environments.

The document provides an in-depth exploration of the Security and Data Governance Infrastructure, encompassing all the modules developed collaboratively by WP3 partners. This comprehensive summary delves into the intricacies of the technologies utilized throughout the task's development. Moreover, the document highlights the crucial interaction and partnership between WP3 and WP6 partners, which has been instrumental in creating a final tool that addresses real-world challenges and offers practical value to end-users. The seamless collaboration between the two teams has ensured that the solution is well-tailored and capable of effectively addressing the needs of the industry, making it a powerful and user-friendly tool.

All the subsequent chapters are a detailed description of the modules part of the final architecture. INTRA describes the Runtime Monitoring System, UBI describes the AI Cyber Defence Module and GFT describes the STAR Security Policies Manager. Component descriptions focus on the advancements from the previous deliverable version (D3.5).

The answers to a survey aimed at assessing the applicability of WP3 tools in WP6 pilots' environments are also summarised, and the questionnaire template utilized for this exercise is reported in Appendix A.

<b>Deliverable Leader:</b>	GFT
<b>Contributors:</b>	GFT, INTRA, UBI
<b>Reviewers:</b>	SUPSI, DFKI
<b>Approved by:</b>	INTRA

<b>Document History</b>			
<b>Version</b>	<b>Date</b>	<b>Contributor(s)</b>	<b>Description</b>
0.1	04/05/2023	GFT	Initial version
0.2	03/08/2023	UBI, INTRA	Contributions
0.3	03/08/2023	GFT	Final version
0.4	21/08/2023	DFKI, SUPSI	Internal reviewed version
1.0	04/09/2023	INTRA	Final QA'ed version

# Table of Contents

<b>EXECUTIVE SUMMARY .....</b>	<b>2</b>
<b>TABLE OF CONTENTS.....</b>	<b>4</b>
<b>TABLE OF FIGURES.....</b>	<b>6</b>
<b>LIST OF TABLES.....</b>	<b>7</b>
<b>DEFINITIONS, ACRONYMS AND ABBREVIATIONS .....</b>	<b>8</b>
<b>1 INTRODUCTION.....</b>	<b>9</b>
1.1 OVERVIEW AND PURPOSE.....	9
1.2 ADDITIONS AND UPDATES FROM LAST DELIVERABLE VERSION .....	9
1.3 RELATIONSHIP TO OTHER DELIVERABLES.....	10
1.4 DELIVERABLE STRUCTURE .....	10
<b>2 THE SECURITY AND DATA GOVERNANCE INFRASTRUCTURE ARCHITECTURE.....</b>	<b>11</b>
2.1 THE COMPONENTS .....	14
2.1.1 <i>Distributed Ledger Services for Data Reliability.....</i>	<i>14</i>
2.1.2 <i>Runtime Monitoring System .....</i>	<i>14</i>
2.1.3 <i>AI Cyber Defence ModuleFigure 2Figure 2 .....</i>	<i>15</i>
2.1.4 <i>OLISTICFigure 2.....</i>	<i>15</i>
2.1.5 <i>Security Policies Manager .....</i>	<i>15</i>
<b>3 RUNTIME MONITORING SYSTEM.....</b>	<b>17</b>
3.1 ARCHITECTURE .....	17
3.2 COMPONENT DIAGRAM AND API IDENTIFICATION .....	19
3.2.1 <i>Interface Specification.....</i>	<i>19</i>
3.3 DATA COLLECTION AND MANAGEMENT .....	20
3.4 GUI.....	22
<b>4 OLISTIC .....</b>	<b>24</b>
4.1 INTRODUCTION.....	24
4.2 ARCHITECTURE .....	25
4.3 FEATURES AND UPDATES.....	27
4.3.1 <i>Adoption of CVSS v3.1 with forward and backwards compatibility with all versions of CVSS 27</i>	<i>27</i>
4.3.2 <i>Automatic generation of Attack Scenarios.....</i>	<i>29</i>
4.3.3 <i>Update Look and Feel of the Risk Assessment and Mitigation Engine .....</i>	<i>30</i>
4.3.4 <i>New OLISTIC APIs and documentation.....</i>	<i>31</i>
4.4 INPUT.....	32
4.5 OUTPUT .....	33
4.6 GUI.....	33
<b>5 STAR SECURITY POLICIES MANAGER.....</b>	<b>34</b>
5.1 ARCHITECTURE .....	34
5.2 GUI AND SECURITY POLICIES DEFINITION .....	35
5.2.1 <i>Policies editor.....</i>	<i>35</i>
5.2.2 <i>Attack Scenario Templates creation .....</i>	<i>37</i>
5.2.3 <i>Link Rules-Scenarios definition .....</i>	<i>37</i>
5.3 POLICIES EVALUATION AND INTERACTION WITH OLISTIC.....	38
<b>6 SECURITY AND DATA GOVERNANCE IN PILOTS' ENVIRONMENTS AND THREAT LANDSCAPE ANALYSIS.....</b>	<b>41</b>

6.1	PHILIPS PILOT .....	42
6.1.1	<i>Environment setup</i> .....	42
6.1.2	<i>Current status and previously witnessed security faults</i> .....	45
6.1.3	<i>Hypothetical scenarios</i> .....	46
6.1.4	<i>Critical Assets</i> .....	46
6.1.5	<i>Properties to be monitored</i> .....	47
6.1.6	<i>Scope of AI Cyber Defence tool in the Philips pilot</i> .....	47
6.1.7	<i>Scope of Runtime Monitoring System tool in the Philips pilot</i> .....	48
6.1.8	<i>Scope of Security Policies Manager in the Philips pilot</i> .....	48
6.2	DKFI PILOT .....	49
6.2.1	<i>Environment setup</i> .....	49
6.2.2	<i>Current status and previously witnessed security faults</i> .....	52
6.2.3	<i>Hypothetical scenarios</i> .....	53
6.2.4	<i>Properties to be monitored</i> .....	53
6.2.5	<i>Scope of AI Cyber Defence tool in the DFKI pilot</i> .....	54
6.2.6	<i>Scope of Runtime Monitoring System tool in the DFKI pilot</i> .....	54
6.2.7	<i>Scope of Security Policies Manager in the DFKI pilot</i> .....	54
6.3	IBER PILOT .....	54
6.3.1	<i>Environment setup</i> .....	54
6.3.2	<i>Current status and previously witnessed security faults</i> .....	57
6.3.3	<i>Environment description</i> .....	58
6.3.4	<i>Hypothetical scenarios</i> .....	60
6.3.5	<i>Critical Assets</i> .....	60
6.3.6	<i>Properties to be monitored</i> .....	61
6.3.7	<i>Scope of AI Cyber Defence tool in the IBER pilot</i> .....	61
6.3.8	<i>Scope of Runtime Monitoring System tool in the IBER pilot</i> .....	62
6.3.9	<i>Scope of Security Policies Manager in the IBER pilot</i> .....	62
6.4	DISCUSSION ON SURVEY RESULTS .....	63
<b>7</b>	<b>CONCLUSIONS</b> .....	<b>64</b>
	<b>REFERENCES</b> .....	<b>65</b>
	<b>APPENDIX A - WP3 TOOLS IN PILOT TEMPLATE</b> .....	<b>66</b>
1.	ENVIRONMENT SETUP .....	66
	<i>Topology of the manufacturing floor or the production line</i> .....	66
	<i>Assets</i> .....	66
2.	ASSET DEPENDENCIES .....	66
3.	KNOWN VULNERABILITIES OF ASSETS .....	67
	<i>Other threats to be reported</i> .....	68
4.	SCENARIOS .....	68
	<i>Current status and previously witnessed security and production faults</i> .....	68
5.	DEFINITION USE CASE SCENARIO .....	69
	<i>Scenario Example</i> .....	69
	<i>Scenario</i> .....	70
6.	WP3 TOOL-SPECIFICS .....	71
	<i>AI Cyber defence</i> .....	71
	<i>RMS</i> .....	71
7.	STAR BLOCKCHAIN .....	72
8.	SECURITY POLICIES MANAGER .....	72

# Table of Figures

FIGURE 1 STAR FUNCTIONAL MODULES AND LOGICAL VIEW OF THE ARCHITECTURE [D2.6].....11

FIGURE 2 STAR SECURITY AND DATA GOVERNANCE FOR AI SYSTEMS IN MANUFACTURING LOGICAL VIEW .....12

FIGURE 3 DLSDR COMPONENT FLOW EXAMPLE .....14

FIGURE 4 RMS DATA FLOW DIAGRAM .....17

FIGURE 5 RMS COMPONENT DIAGRAM .....19

FIGURE 6 PROBE DATA STORAGE SEQUENCE DIAGRAM .....20

FIGURE 7 RMS INFRASTRUCTURE FLOW .....21

FIGURE 8 RMS DATA COLLECTION, TRANSFORMATION & FILTERING EXAMPLE .....22

FIGURE 9 KIBANA DISCOVERY VIEW DASHBOARD .....22

FIGURE 10 KIBANA DASHBOARD VIEW .....23

FIGURE 11 OLISTIC INTERNAL COMPONENT ARCHITECTURE .....24

FIGURE 12 INTERDEPENDENCY GRAPH VISUALISING THE MANUFACTURING ENVIRONMENT .....26

FIGURE 13 VULNERABILITY PROFILES ON THE ASSETS FOR CVSS v2.0 AND v3.x .....28

FIGURE 14 VULNERABILITY TEMPLATE SELECTION FOR CVSS v2.0 AND v3.x .....29

FIGURE 15 AUTOMATED GENERATION OF ATTACK SCENARIOS .....30

FIGURE 16 UPDATED DASHBOARD UI .....31

FIGURE 17 UPDATED RISK ASSESSMENT UI .....31

FIGURE 18 INSTANCE OF THE SWAGGER UI OF OLISTIC .....32

FIGURE 19 ENDPOINT FOR CREATING A NEW ATTACK SCENARIO IN THE OLISTIC ENVIRONMENT .....32

FIGURE 20: SSPM HIGH LEVEL ARCHITECTURE .....34

FIGURE 21 SSPM POLICIES VIEW .....36

FIGURE 22 SSPM EDITOR VIEW .....37

FIGURE 23 SSPM ATTACK SCENARIO TEMPLATES .....37

FIGURE 24 SSPM LINK RULES-SCENARIOS VIEW .....38

FIGURE 25 ATTACK SCENARIO CREATED IN OLISTIC .....39

FIGURE 26 RAISED RISK APPETITE IN OLISTIC .....39

FIGURE 27 RISK ASSESSMENT CREATED AND FIRED IN OLISTIC .....40

FIGURE 28 TOPOLOGY OF THE CHERRY INSPECTION STATION .....43

FIGURE 29 OLISTIC DIGITAL REPRESENTATION OF THE PHILIPS ENVIRONMENT .....45

FIGURE 30 GENERATION OF ADVERSARIAL EXAMPLES (DEEPOOL ATTACK) FOR THE SOOTHER DATASET .....48

FIGURE 31 DFKI ENVIRONMENT SETUP .....50

FIGURE 32 OLISTIC DIGITAL REPRESENTATION OF DFKI'S ENVIRONMENT .....52

FIGURE 33 IBER'S ENVIRONMENT SETUP .....55

FIGURE 34 OLISTIC DIGITAL REPRESENTATION OF IBER'S ENVIRONMENT .....57

FIGURE 35 ALARM NUMBER FOR EACH PLC, EXAMPLE .....58

FIGURE 36 AI CAMERA .....59

FIGURE 37 CAMERA TESTS ON REAL COMPONENTS .....59

FIGURE 38 GENERATION OF ADVERSARIAL EXAMPLES (DEEPOOL ATTACK) FOR THE IBER DATASET .....62

## List of Tables

TABLE 1 ADDITIONS AND UPDATES FROM LAST DELIVERABLE VERSION.....	9
TABLE 2 RMS INTERFACE SPECIFICATION .....	19
TABLE 3 ASSETS OF THE CHERRY INSPECTION STATION .....	43
TABLE 4 RELATIONSHIPS OF THE ASSETS OF THE CHERRY INSPECTION STATION .....	44
TABLE 5 ATTACK SCENARIOS FOR POLICIES FOR PHILIP'S SCENARIO 1.....	49
TABLE 6 ASSETS OF DFKI'S PILOT .....	50
TABLE 7 RELATIONSHIPS OF THE DFKI'S ASSETS .....	51
TABLE 8 DFKI IDENTIFIED POTENTIAL VULNERABILITIES .....	52
TABLE 9 ASSETS OF DFKI'S PILOT .....	54
TABLE 10 ASSETS OF IBER'S PILOT .....	55
TABLE 11 RELATIONSHIPS OF IBER'S ASSETS .....	56
TABLE 12 IBER IDENTIFIED POTENTIAL VULNERABILITIES .....	57
TABLE 13 IBER IDENTIFIED POTENTIAL THREATS .....	58
TABLE 14 ATTACK SCENARIOS FOR POLICIES FOR IBER'S SCENARIO 1 .....	62

## Definitions, Acronyms and Abbreviations

Acronym/ Abbreviation	Title
<b>AICD</b>	AI Cyber Defence
<b>API</b>	Application Programming Interface
<b>CPE</b>	Official Common Platform Enumeration
<b>CPPS</b>	Cyber Physical Production Systems
<b>DLSDR</b>	STAR Distributed Ledger Services for Data Reliability
<b>DoA</b>	Description of Action
<b>iFrame</b>	Inline Frame
<b>GUI</b>	Graphical User Interface
<b>JSON</b>	JavaScript Object Notation
<b>RMS</b>	Runtime Monitoring System
<b>SSPM</b>	Star Security Policy Manager
<b>UC</b>	Use Case
<b>URL</b>	Universal Resource Locator
<b>WP</b>	Work Package

# 1 Introduction

## 1.1 Overview and Purpose

The objectives of WP3 “Security and Data Governance for AI Systems in Manufacturing” are focused on the realization of the security and data governance layer of the STAR project.

The main pillars of WP3 are:

- the establishment of decentralized reliability for industrial data,
- the design of the cyber-defence module against poisoning and evasion attacks, and
- the design and development of the data governance platform.

The aim of deliverable D3.6 “Security and Data Governance Infrastructure-Final version” is to describe the advancements made in the development of the Security and Data Governance Infrastructure after D3.5. Inputs come from GFT, UBI and INTRA, and the focus of the report is:

- The description of the advances made on the modules of the Data Governance Infrastructure: the Runtime Monitoring System, the Explainable AI (XAI), the risk assessment module OLISTIC, the Star Security Policies Manager (SSPM);
- The final version of the Security and data Governance Infrastructure architecture including the above-mentioned modules, the input and output needed for the creation of a validation scenario;
- The final assessment of use cases assets and needs in terms of security policies required for the use cases implementation in the framework of STAR project.

As a result, in the deliverable at hand the components of the STAR AI Security and Data protection layer and the interactions between modules within the WP3 Architecture have been described. The described architecture is integrated in the overall WP3 architecture by documenting in parallel the inputs/outputs of all WP3 components and defining an initial set of interactions of the envisioned components.

## 1.2 Additions and updates from last deliverable version

In the following table an overview on relevant addition and updates from the previous deliverable (D3.5) is given.

*Table 1 Additions and updates from last deliverable version*

Additions/Updates	Section	Partner
<b>Security Policies Manager</b>		
Detailed description of OPA was removed, the reader can refer to D3.5 for detailed information on the policy engine used in the SSPM tool.	2.1.5	GFT
Advancements in the SSPM tool are described in Section 5. Among the new features: new GUI to define the Security Policies and configure the interaction with OLISTIC, new architecture to reflect the advancements in the development.	5.1, 5.2, 5.3	GFT
<b>OLISTIC</b>		
Adoption of CVSS v3.1 with forward and backwards compatibility with all versions of CVSS.	4.3.1	UBI

Automated generation of attack scenarios based on the combination of Common Platform Enumeration and Common Vulnerabilities and Exposures frameworks.	4.3.2	UBI
Updated look and feel of the OLISTIC tool.	4.3.3	UBI
New OLISTIC APIs and documentation for full-fledged integration with SPM.	4.3.4	UBI
<b>Runtime Monitoring System</b>		
Updated RMS architecture by enhancing the data collection mechanism.	3.1, 3.3	INTRA
Enhancing the data visualization capabilities by providing additional stats, visualization charts and data discovery tables.	3.4	INTRA
<b>WP3 in Pilots' environments</b>		
Section 6, 7 and 8 of D3.5 were removed.	N.A	
Section 6 describes an exercise conducted with the use case partners of WP6 to assess the applicability of WP3 tools.	6	GFT, UBI, INTRA
<b>Updates compared to D3.5 (apart from the above-mentioned updates)</b>		
Completion of WP3 tool integration actions.		All
Analysis of the Use Cases environment and threat landscape.	6	WP3 and WP6 partners
Identification of critical assets and indicators to be monitored.	6	WP3 and WP6 partners

### 1.3 Relationship to other deliverables

This deliverable is mainly linked to WP2, WP3 and WP6 deliverables which are listed below:

- D2.2 for the description of Use Cases.
- D2.6 for the STAR Reference Architecture.
- D3.1 for the technical description of STAR Blockchain infrastructure.
- D3.3 for the technical description of AI Cyber Defence components.
- D3.5 as starting point of the current deliverable.
- D6.3 provides input for the integrated STAR platform.

### 1.4 Deliverable Structure

The deliverable is divided in these sections:

- **Section 2** describes the Security and Data Governance Infrastructure architecture with a summary of each module developed by the task partners;
- **Section 3** deepens in the description of the RMS by INTRA and the way it collects data from IoT devices;
- **Section 4** provides information on OLISTIC, UBI's platform which creates asset cartographies;
- **Section 5** describes the Security Policies Manager developed by GFT;
- **Section 6** investigates how WP3 tools can be applied in Pilots environments and describes security policies that can be implemented in the future;
- **Section 7** concludes the deliverable.

## 2 The Security and Data Governance Infrastructure architecture

As detailed in D2.6 “STAR Reference Architecture and Blueprints-Initial version” Figure 1 presents the logical view of the STAR architecture. The diagram presents the main functional modules of STAR compliant systems, along with their structure and their interactions with other systems. The STAR systems are aimed at securing existing CPPS systems in manufacturing production lines (notably AI systems) based on a holistic approach that includes the following pillars, here are listed for the sake of a better understanding of the report:

- Secure and Reliable Data;
- Secure and Trusted AI algorithms;
- Trusted Human AI interactions;
- Safe AI systems.

The STAR architecture provides the structuring principles for the integration of the project’s systems for trusted AI.

As illustrated in Figure 1, the STAR systems receive data from the shop floor (i.e., digital manufacturing platforms and other AI-based CPPS systems) and provide different types of services to factory (cyber)security teams and to other factory stakeholders (e.g., industrial engineers, plant managers, factory workers).

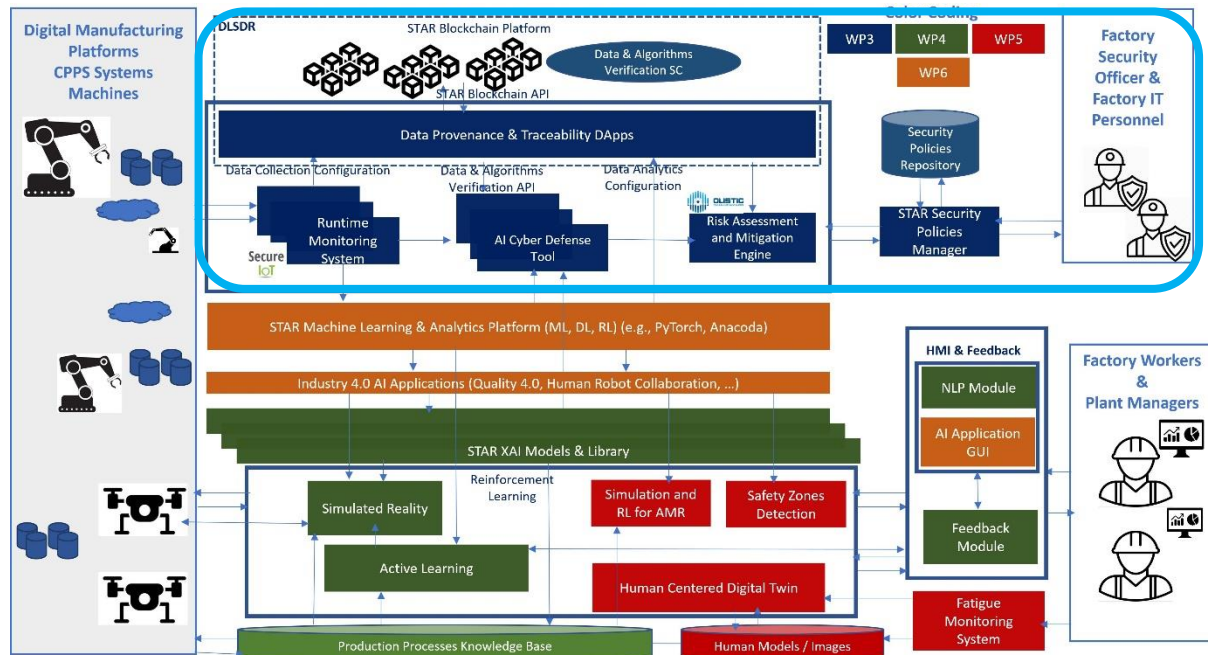


Figure 1 STAR Functional Modules and Logical View of the Architecture [D2.6]

On the other hand, Figure 2 illustrates the architecture of the STAR AI Security and Data protection layer, which is an outcome of the collaborative actions of all partners of WP3. Overall, this layer brings together the AI security and data governance techniques of the STAR project, which are destined to protect AI systems from poisoning and evasion attacks, while boosting the reliability of industrial data through blockchain-based data provenance mechanisms. The layer is complemented by the risk assessment and attack mitigation

functionalities, which are provisioned by the synergistic operation of the STAR Security Policies manager (provided by GFT) and UBITECH’s OLISTIC risk management engine. In addition, the security layer of STAR adopts runtime monitoring mechanisms, able to monitor critical devices and collect important measurements that can reveal the behavioural profile of the production lines. The functionalities of this layer aim to secure existing digital manufacturing platforms and devices that comprise AI systems in the manufacturing line of the STAR demonstrator, while also supporting the safety, reliability, and transparency functionalities of the upper layer of STAR that aim to augment the business operation of the STAR pilot environments.

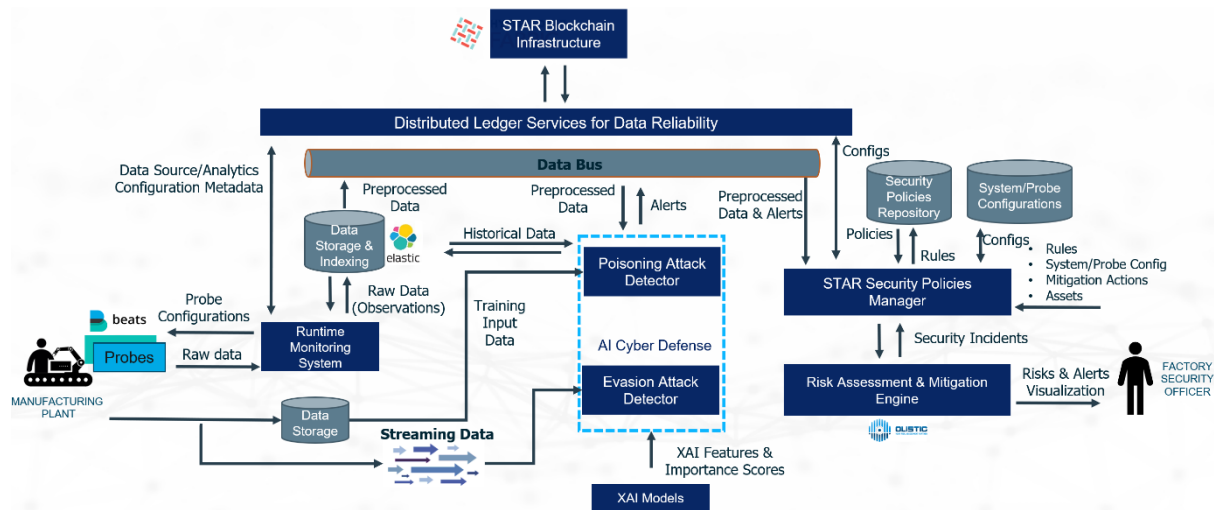


Figure 2 STAR Security and Data Governance for AI Systems in Manufacturing Logical View

The purpose of the AI Security and Data protection layer is to guarantee the operational assurance and credibility of a manufacturing floor. The aim is to offer to a Factory Security Officer the means to govern and regulate the operational behaviour of the manufacturing environment. In STAR, this is achieved by combining the individual components in a unified flow that delivers to the security officer the necessary indications and alerts that reflect the security and operational status of the monitored deployment.

More specifically, as depicted in Figure 2, the officer interacts with the **STAR Security Policies Manager** for the determination of security rules that reflect the legitimate or the abnormal behaviour of a monitored production line. Thus, the officer defines proper rules, system configurations, mitigation actions and the assets that need to be monitored. These inputs are maintained in the local repositories of the STAR Security Policies Manager, while the latter can perform validation checks, based on the provided rules and evidence, to infer the security and operational status of the monitored environment. The detection of security events, or discrepancies in the legitimate behavioural profile of devices, is then illustrated on the dashboard of the **Risk Management and Mitigation Engine**, realized as an extension of UBITECH’s OLISTIC risk assessment engine. Capitalizing on the components, the officer can both regulate the operation of a monitored production line and have an overview via visualisations through the OLISTIC’s dashboard.

The AI Security and Data protection layer follows an event-driven approach, meaning that actions are triggered based on the emergence of events. In the AI Security and Data protection layer there are two main components generating events. The **Runtime Monitoring System** and the **AI Cyber Defence** components. The former, as implied by its

title, aims to provide evidence on the operational behaviour of core manufacturing devices during runtime. This is achieved either through the deployment of probes or by polling/receiving data from available services (e.g., exposed APIs). This set up is configured to monitor critical device resources, based on which the Data management and Analytics Engine can form statistical measurements, called Observations, that reflect the behavioural profile of systems. As far as the AI Cyber Defence component is concerned, the aim is the triggering of alerts upon the detection of data Poisoning and data Evasion attacks that may threaten the AI-enabled systems of a production line. This component is positioned in the middle of the data pipelines used for training the AI systems of STAR or the pipelines that feed the AI systems in a dynamic manner, prior to the inference stage of the deployed AI model, for the sake of detecting malicious attempts trying to evade the classification process of systems during the inference (runtime) operational mode. Upon the detection of such incidents, proper alerts are generated to be forwarded to the Security Policies Manager for further processing and validation.

On top of the described technical components of the security layer, a permissioned (not publicly accessible) blockchain infrastructure is deployed to provide the data governance quality. More specifically, the STAR blockchain aims to improve the reliability and security of industrial data and of the analytics algorithms used to process them including Machine learning and Deep Learning tools. The blockchain infrastructure is deployed over an edge computing infrastructure, which is a typical deployment configuration for industrial applications. Over the blockchain infrastructure STAR, the AI Security and Data protection layer support the reliable storage/management of:

- Data Analytics Configurations
- Data Analytics Results

Using the Distributed Ledger (blockchain) as the distribution channel ensures a truly decentralized but also reliable system. The collected data are "signed, sealed and timestamped" so that no forgery or tampering is possible. The virtues of this workflow lie in immutability and non-repudiation: the Distributed Ledger acts as an official registry of critical data and system/algorithms configurations, where the business-critical information is owned by the owner of the monitored infrastructure and ensures the reliability of the data analytics and AI outcomes.

Overall, the abstract workflow described above is depicted in the architecture of Figure 2. The following sections provide more details on each of the components individually, elaborating on their internal architecture, inputs/outputs and the developed methodologies that drive their operation.

Note that, the STAR Blockchain infrastructure and the AI Cyber Defence components have been described in detail in the context of D3.2 and D3.4, respectively. Thus, this deliverable describes only the placement and interactions of those components in the frame of the AI Security and Data protection layer. When it comes to the Runtime Monitoring System, the OLISTIC risk assessment engine, and the Security Policies Manager, these components are introduced for the first time in the 3.5 deliverable and better defined in the current one.

## 2.1 The components

### 2.1.1 Distributed Ledger Services for Data Reliability

As depicted in Figure 2 above, the Distributed Ledger Services for Data Reliability (DLSDR) offers the following functionalities to the STAR Security & Data Governance framework:

- For persisting/retrieving the AI algorithms configurations metadata which can describe an algorithm type along with its various instantiation configurations across time by using the Analytics Engine Configuration (AEC) service (see D3.2 section 3.3.1). Information about the exposed API, data models and usage can be found in section 4.2.3 of D3.2.
- For persisting AI algorithm results by utilizing the Analytics Results Publishing (ARP) service (see D3.2 section 3.3.2) using the Observation data structure as described in D3.2 section 4.2.4. Information about the exposed API can be found in section 4.2.4 of D3.2. Samples of the blockchain persisted Analytics' results can be consumed by the Security Policies Management component to confirm their validity compared to the results that are retrieved from the Data Bus. Critical results can be directly retrieved from the Data provenance & Traceability component.

Using the Distributed Ledger as the distribution channel, STAR ensures a truly decentralized but also reliable system, as analytics manifests are "signed, sealed and timestamped" so that no forgery or tampering is possible. Moreover, the AI Cyber Defence component will be able to share analytics results on the Distributed Ledger infrastructure, thus contributing to a common data set representing the combined results across the entire distributed system (see Figure 3 below). The virtues of such a workflow lie in immutability and non-repudiation.



*Figure 3 DLSDR component flow example*

### 2.1.2 Runtime Monitoring System

The Runtime Monitoring System (RMS), depicted in Figure 2, is a Data collection framework which provides the specifications and relevant implementation to enable a real time data collection, transformation, filtering, and management service to facilitate data consumers (e.g., AI Cyber Defence Module and Security Policies Manager). The framework can be applied in IoT environments supporting solutions in various domains (e.g., Industrial, Cybersecurity, etc.). For example, the solution may be used to collect security related data (e.g., network, system, solution proprietary, etc.) from monitored IoT systems and store them to detect patterns of abnormal behaviour by applying simple (i.e., filtering and pre-processing) mechanisms. The design of the framework is driven by configurability, extensibility, dynamic setup and stream handling capabilities. One of the key features of the framework is that it is detached from the underlying infrastructure by employing a specialized data model for modelling the solution's Data Sources, Processors and Results which facilitates the data interoperability discoverability and configurability of the offered solution.

### 2.1.3 AI Cyber Defence Module

The AI Cyber Defence tool is positioned in the middle, between the manufacturing plants (Figure 2 left) and the Security Policies manager and the Risk Assessment & Mitigation Engine (Figure 2 right). As already mentioned in the D3.5, the purpose of the tool is the evaluation of the training and streaming data stemming from the data lakes and the deployed systems, respectively, so that to detect possible poisoning and evasion attacks.

In addition, the AI Cyber Defence module takes advantage of XAI models to detect abnormal behaviours in time-series data stemming from the sensors operating in the manufacturing floor (annotated as “Historical Data” in Figure 2). This is a newly reported feature in D3.4, which has been approached from a research perspective and it is not destined to modify the execution behaviour of the tool or the flow of actions.

Upon the detection of an incident, Alerts are generated and pushed through the Data Bus to the Security Policies manager and the Risk Assessment & Mitigation Engine for further analysis and for informing the security administrator about the detected incidents. A complete documentation of the AI Cyber Defence tool is given in D3.4. The interested reader can refer to D3.4 to check the updates made to the AI Cyber Defence engine since the 1<sup>st</sup> release of the tool that it was initially reported in D3.3.

### 2.1.4 OLISTIC

OLISTIC is the technical component that will complement the Security Policies Manager and will provide the dashboard of the Risk Assessment and Mitigation Engine of STAR. More specifically, OLISTIC is UBITECH’s Risk Assessment tool which can support the security officer in getting an overview of the security status of the factory, and more specifically, of the production lines and business processes of interest. Thus, OLISTIC contributes in the flow of the architecture illustrates in Figure 2, as the component that receives the security incidents that are being detected by Security Policies Manager, as a result of policy violations, and offers to the security officer an interactive dashboard in order to understand the security posture of the manufacturing environment, considering the existing vulnerabilities and weak points of systems.

Towards the completion of the WP3 actions, the scope and purpose of OLISTIC remains the same to what was initially reported in D3.5. However, as will be documented with details on section 4, OLISTIC has been significantly updated in the context of the STAR project, providing a) a new back-end and front-end design that fits better to the needs of the STAR project, b) final definition of the APIs that enable the communication of other tools with OLISTIC so that to exchange the information on detected abnormal events, and c) definition of APIs that enable the remote enactment of risk assessment actions upon the detection of incidents.

The updates on the Risk Assessment and Mitigation Engine of STAR, are reported in Section 4.3.

Overall, OLISTIC enables the risk management and the identification and visualization of risks through comprehensive and reactive visualization, while it provides the means to the security officer to manage the life cycle of mitigation actions.

### 2.1.5 Security Policies Manager

The STAR Security Policies Manager (SSPM) is a STAR Platform Security and Data Governance module for AI Systems architecture, whose objective is data protection and reliability against poisoning and evasion attacks.

SSPM is a tool to be used by the personnel of the factory, in particular security/IT officers, to configure security policies according to specific business and security requirements. The main purpose of the SSPM is to detect poisoning and evasion attacks and report this risk to the Risk Assessment module OLISTIC, generating alerts.

SSPM integrates the cyber defence mechanism of the Star Blockchain infrastructure, Data Provenance & Traceability, RMS, and AI Cyber Defence module.

SSPM receives inputs from the RMS and AI cyber defence through the Data Bus and validates data provenance and traceability components. SSPM implements the risk assessment functionalities based on OLISTIC, Risk Assessment Engine, giving input to the tool and communicating the existence of a threat, generating alerts. The SSPM software considers various types of attacks, including poisoning or evasion attacks.

The main module of the software is an open-source, general-purpose policy engine that unifies policy enforcement across the stack, named Open Policy Agent (OPA)<sup>1</sup>. In D3.5 an extensive overview of OPA functionalities is given along with a comparison with other similar tools and the reasons to select OPA over other options. We invite the interested reader to refer to D3.5 for additional information. D3.6 focuses instead on the advancements in the development of the SSPM solution, which are described in Section 5.

---

<sup>1</sup> <https://www.openpolicyagent.org/docs/latest/>

### 3 Runtime Monitoring System

#### 3.1 Architecture

As mentioned above, the RMS enables a real time service that collects security-related data from monitored IoT system components or applications and stores them for further processing. Analytics algorithms, like the AI Cyber Defence component, analyse the collected data to detect abnormal patterns. Additionally, the collected data can be directly used by the Security Policies Manager after applying special filters for reporting data exceeding “normal” thresholds. The system also features monitoring probes responsible for the data collection and publishing to the monitoring platform. The RMS provides appropriate configuration and management mechanisms over the monitoring probes as well as appropriate data models and data transformation engines that will maintain the probe information along with their status and will enable the probe creation, reconfiguration, and discovery. The RMS component has been adapted and extended from previous EU projects like H2020-SecureIoT and H2020-IoTAC projects.

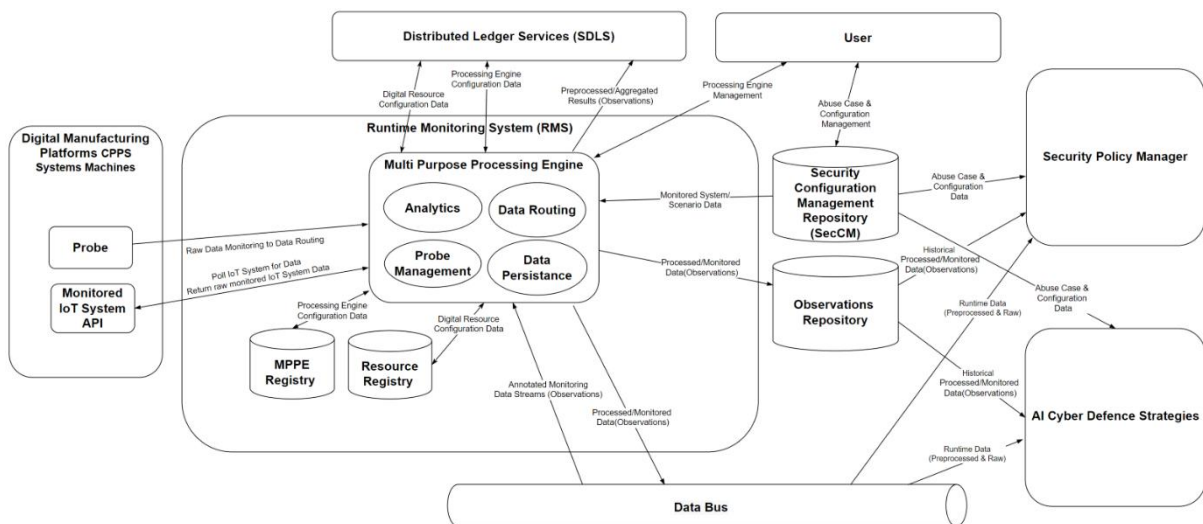


Figure 4 RMS Data Flow Diagram

As shown in Figure 4 RMS offers the following functions:

- **Multi-Purpose Processing Engine (MPPE):** The MPPE provides a wrapper for data processing instances (such as an algorithm or a data persistence service) that allows them to be managed and data compatible (input/output) with the Runtime Monitoring System. The RMS data models like, Processor Definition (an entity containing the characteristics of a processor such as description, vendor, availability, supported attributes, and so on), Manifest (an entity containing the instantiation of a processor based on the processor description), and Orchestrator (an entity containing the instantiation of a processor based on the processor description) are all used by Processing Engine (an entity containing a list of processor manifests capable of describing a complex processing flow). More information of the MPPE data models can be found in D3.1 section 4.2.3.1. Below we can find a list of supported MPPE wrappers that enables different functionalities to RMS:
  - **Probe Management:** The wrapper is responsible for managing and configuring the deployed probes. It can receive automatic probe configuration commands and correspondingly configures the managed probes.

- **Data Routing:** The wrapper enables the transformation, annotation, filtering, and routing of incoming data streams to temporary (i.e., Data Bus) or permanent (i.e., Observation repository) data storage.
  - HTTP Poller: Is part of the Data Routing and collects user data (i.e., image production rate) by polling the exposed services (repository).
- **Analytics Algorithm:** The wrapper oversees analysing the data and issuing alarms when inappropriate behaviour is discovered. The Processing Engine helps with Analytics Algorithm configuration, data management, and system interaction.

RMS (see Figure 4) interacts with the following External entities:

- **Probes:** Probes collect data from the target IoT system or application and stream them to the IoT platform through the data routing wrapper.
- **User:** The user utilizes the RMS configuration and management APIs to control the deployment characteristics and supported scenarios.
- **External Application:** It receives the analytic results of the data processing and can execute further processing or necessary reaction to the anomalies. As shown in Figure 4 in the STAR project we have two main RMS external applications which are the Security Policies Manager and the AI Cyber Defence Strategies.

RMS requires various data stores for persisting its configurations and results. The following list provides the core data stores supported by RMS as depicted in Figure 4:

- **Data Bus:** Data Bus is a communications channel through which all real time data is routed. Platform components may subscribe to the data bus to receive data of specific interest to them.
- **Resource Registry:** The Resource Registry keeps track of all the resources that are available (e.g., probes). The registry keeps track of resource-related data, as well as status and configuration data. The registry allows you to create, reconfigure, and search for resources. It also makes dynamic resource finding easier.
- **Observations Repository:** Observations Repository contains historic security data that have been collected by the deployed probes. These data can be used by the Data Analytics to train itself and produce a set of security templates that will be used subsequently for identifying security issues on the target IoT system.
- **Security Configuration Management (SecCM) Repository:** SecCM contains information about all assets of the RMS related to the monitored System along with their attributes and configuration parameters. Some of the entities that comprise the SecCM repository are:
  - Monitored System: providing a description, location, organization, etc.
  - Monitored Asset: providing the vendor, asset category, system that belongs to, descriptions, installation/inventory dates, relationships with other assets, configuration attributes, etc.
  - Control actions over the assets that might be applied,
  - System Vulnerabilities, and Attack scenarios which are comprised of several misuse cases.
- **MPPE Registry:** MPPE Registry maintains a record of the deployed processors. Processors' type and instance data are maintained by the registry. The registry provides processor definition, instantiation reconfiguration, and search capabilities. This repository is utilized by the Processor Engine.

## 3.2 Component Diagram and API identification

The RMS includes five core components, as follows: Probe Management & Configuration, Probe Registry, MPPE Registry, Data Routing, and Multipurpose Processing Engine. The interfaces between RMS and other STAR modules and Common Components (i.e., Data Bus, Observational Repository, and SecCM Repository) are shown in Figure 5 below, and described in Section 3.2.1.

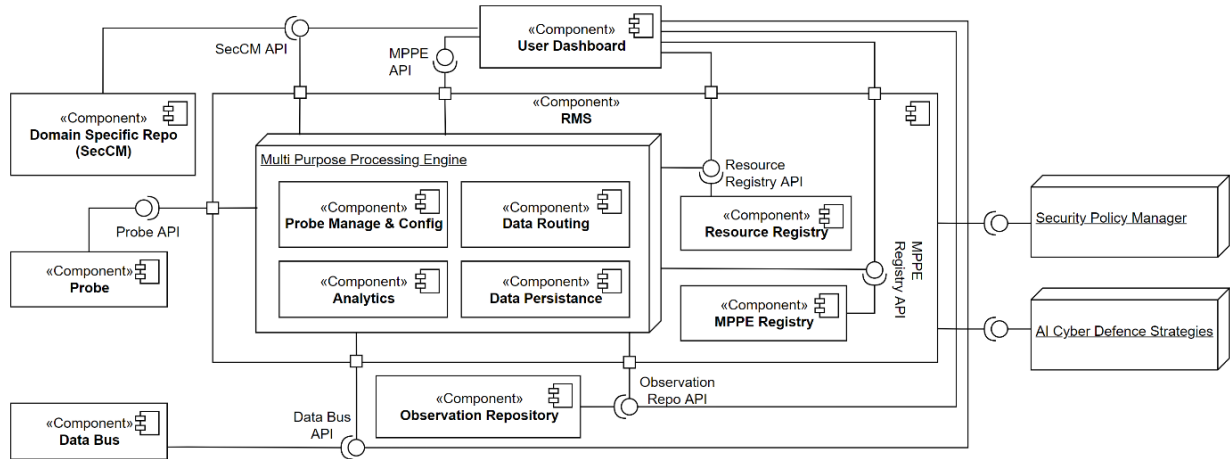


Figure 5 RMS component Diagram

### 3.2.1 Interface Specification

Table 2 provides a list of the interfaces depicted in Figure 5 above. The table distinguishes the interfaces between the ones provided from the RMS and the ones that are required/used from the RMS to interact with other components.

Table 2 RMS Interface Specification

No	API	Description	Provided	Required
1	<b>Probe API</b>	Probe API enables the control of a Probe by exposing configuration (sending a probe configuration file) and control (start/stop) interfaces.	X	
2	<b>PMC API</b>	Probe Management & Configuration API exposes appropriate endpoints that enables the discoverability, configurability, and management of the deployed probes.	X	
3	<b>MPPE API</b>	Multi-Purpose Processing Engine API exposes appropriate endpoints that enable the discoverability, configurability, and management of deployed processors.	X	
4	<b>MPPE Registry API</b>	Multi-Purpose Processing Engine Registry API exposes appropriate endpoints that enable the discoverability and configurability of deployed processors. This API is utilized by the MPPE API.	X	
5	<b>DR API</b>	Data Routing API exposes appropriate endpoints that enable the configuration of data streams within the annotation and routing of incoming data streams to persistence or data management components.	X	
6	<b>AR API</b>	Automatic Reconfiguration API exposes appropriate endpoints that enable the	X	

No	API	Description	Provided	Required
		configuration and control and triggering of the Automatic Reconfiguration component.		
7	<b>PR DB API</b>	Probe Registry API exposes appropriate endpoints that enable the discoverability and configurability of deployed Probes. This API is utilized by the Probe Management & Configuration API.	X	
8	<b>Observation Repo API</b>	Observation Repository API exposes appropriate endpoints that enable the discoverability and usage of captured, pre-processed, and processed data		X
9	<b>Data Bus API</b>	Data Bus API exposes appropriate endpoints that enable the temporary persistence, publishing, subscribing and retrieval of data streams.		X

In Figure 6 we can find a sequence diagram providing the flow of initiating and storing measurements from the monitored system. We see that the User initiates the process (starts the Probe) through the probe management wrapper. Then a continuous loop of the Probe collecting the required measurements, pushing them to the data routing wrapper and persisting them to the Monitoring Data Storage and Data bus is performed.

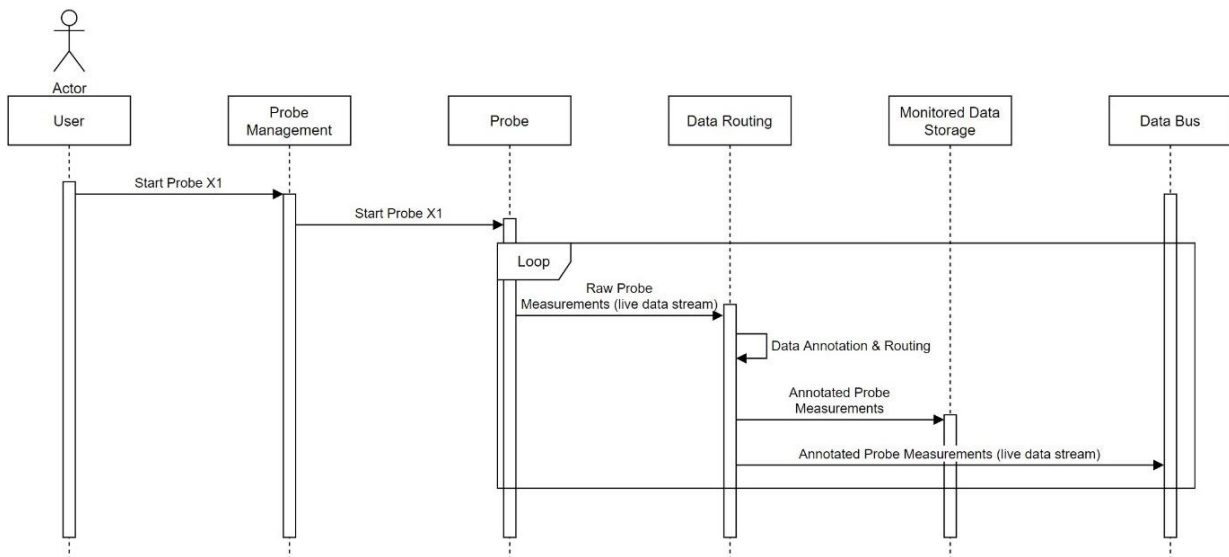


Figure 6 Probe data storage sequence diagram

### 3.3 Data Collection and Management

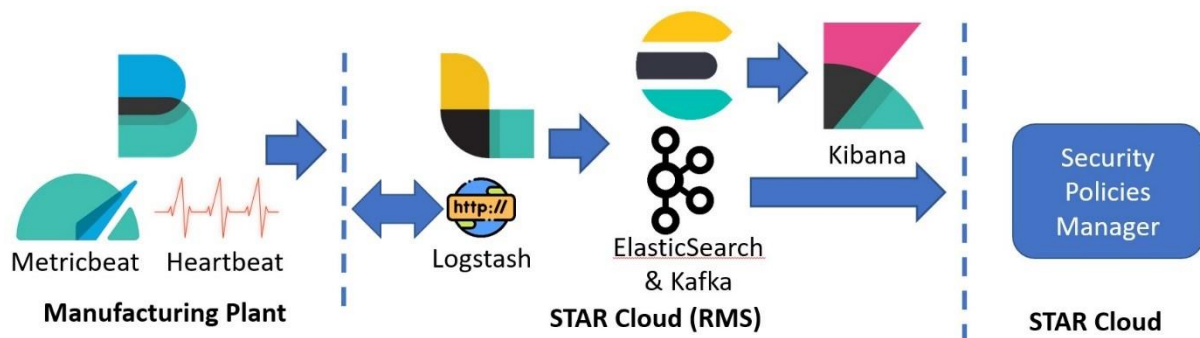
For the RMS component’s implementation, we are using elastic stack<sup>2</sup> which is comprised of Elasticsearch, Kibana, Beats, and Logstash (also known as the ELK Stack) and Kafka for the Data Bus. The different components are used as follows:

- **MetricBeats, HeartBeat:** collects monitored data (i.e. CPU utilization data) and availability status (i.e., network Camera availability) using Beats deployed to the Manufacturing Plant (demo VM).

<sup>2</sup> <https://www.elastic.co/elastic-stack/>

- **HTTP Poller:** collects user data (i.e., image production rate) by polling the exposed services (repository).
- **Logstash:** Raw monitored Data are transformed and filtered to match the used Data Model (i.e., Observations) and identified rules (i.e., report values between specific thresholds).
- **Kafka & ElasticSearch:** the collected preprocessed data are published to the Data Bus (Kafka) in order to be accessed by the Security Policies Manager & ElasticSearch for permeate persistence, visualization and monitoring.
  - Security Policies Manager retrieves the preprocessed data by the Data Bus (Kafka) in order to be combined with other alerts/data (i.e., the AI Cyber Defense Strategies).
- **Kibana:** for persisted data visualization.

A typical flow of usage of the abovementioned components is depicted in Figure 7 below.



*Figure 7 RMS infrastructure flow*

As mentioned above we are using Elastic Beats<sup>3</sup> for collecting monitored data. Elastic Beats are lightweight data shippers, written in Go, that have a small installation footprint and use limited system resources with no runtime dependencies. There are several different options for installing Elastic Beats which are:

- As an operating system service (DEB, RPM, MacOS, Brew, Linux, Windows)
- Docker Environment
- Kubernetes (DaemonSet)

Finally, there are several different Elastic Beats supported for retrieving various type and format of data. The main options are offered by Elastic<sup>4</sup> but there are also third-party ones offered by the Elastic community<sup>5</sup>. The initial options investigated in STAR to be used for collecting monitored data are the:

- **Filebeat**<sup>6</sup> which tails and ships log files.
- **Metricbeat**<sup>7</sup> which fetches sets of metrics from the operating system and services.
- **Packetbeat**<sup>8</sup> which monitors the network and applications by sniffing packets.

<sup>3</sup> <https://www.elastic.co/guide/en/beats/libbeat/current/index.html>

<sup>4</sup> <https://github.com/elastic/beats>

<sup>5</sup> <https://www.elastic.co/guide/en/beats/libbeat/master/community-beats.html>

<sup>6</sup> <https://github.com/elastic/beats/tree/master/filebeat>

<sup>7</sup> <https://github.com/elastic/beats/tree/master/metricbeat>

<sup>8</sup> <https://github.com/elastic/beats/tree/master/packetbeat>

A data flow sample of Elastic Beats format data collection, Logstash configuration for data transformation and filtering and result persistence in observation format is depicted in Figure 8 below.

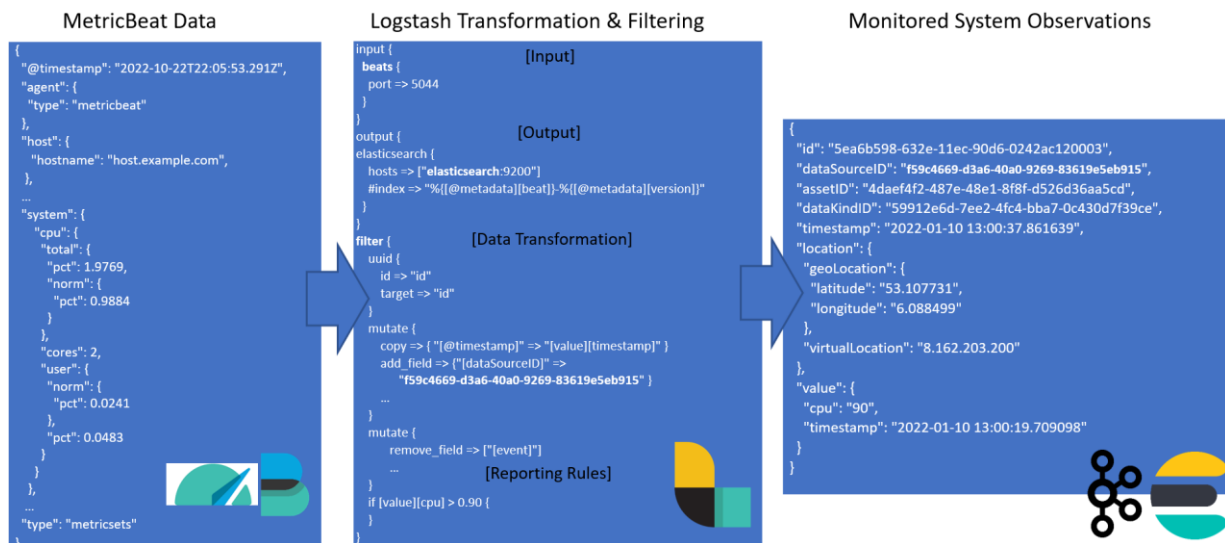


Figure 8 RMS data collection, Transformation & Filtering example

### 3.4 GUI

RMS offers a monitoring dashboard for depicting the collected monitoring data. The dashboard is based on Elastic Kibana<sup>9</sup> technology and offers querying functionalities over the Observation repository (which is based on Elastic Search<sup>10</sup>) and appropriate widgets to report to the security expert the monitored values. This dashboard is an intermediate to the Risk Assessment and Mitigation dashboard (which will be offered by OLISTIC tool). In Figure 9 and Figure 10, we can see an examples of a system utilization monitoring by providing the data discovery view and the dashboard view respectively.

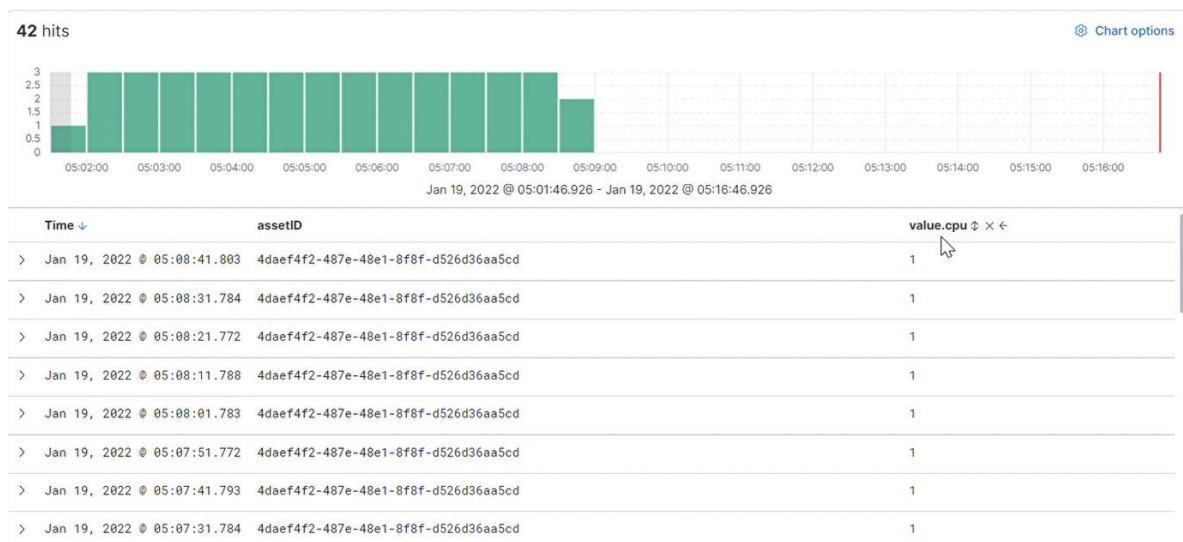


Figure 9 Kibana discovery view dashboard

<sup>9</sup> <https://www.elastic.co/kibana/>

<sup>10</sup> <https://www.elastic.co/elasticsearch/>

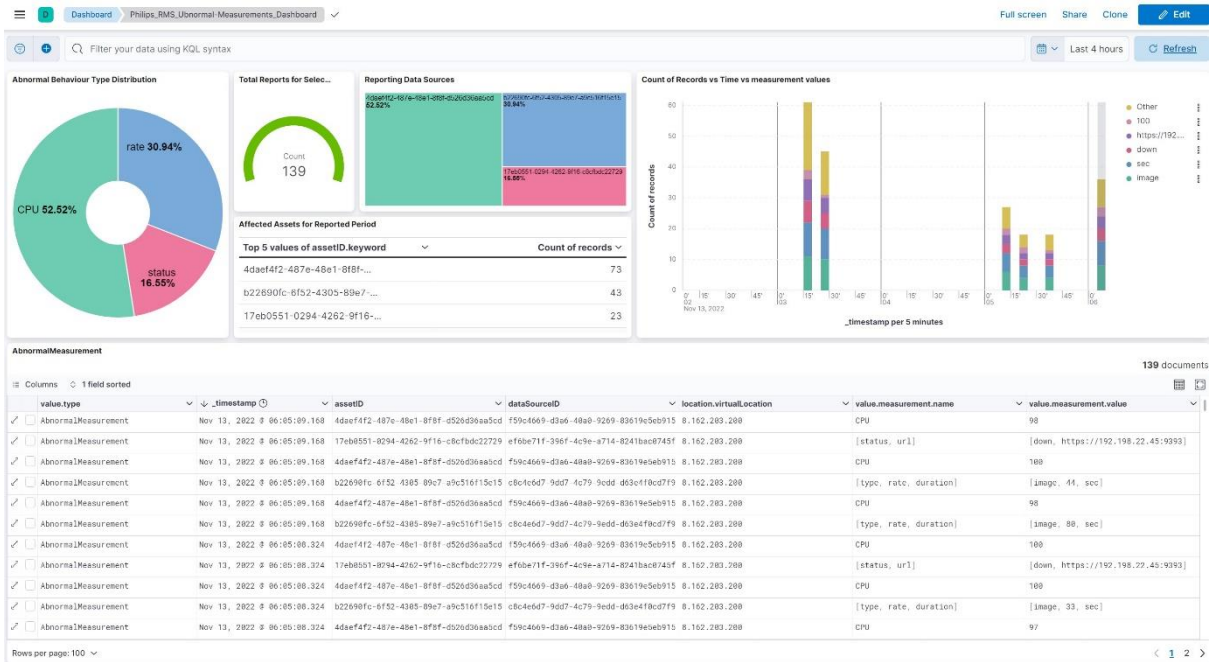
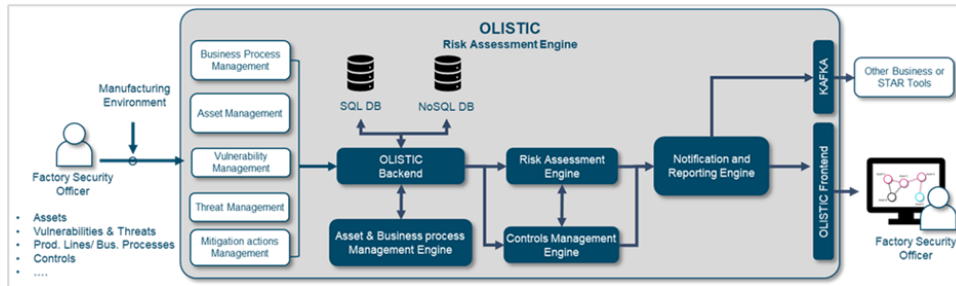


Figure 10 Kibana dashboard view

## 4 OLISTIC

### 4.1 Introduction



*Figure 11 OLISTIC Internal Component Architecture*

OLISTIC is used as one of the dashboards of the AI Security and Data Governance layer of STAR and enables the factory security officer to get an overview of the security state of the production lines of the factory and perform risk assessment functionalities. The internal architecture of the OLISTIC tool is given in Figure 11.

A thorough documentation of OLISTIC was given in D3.5, highlighting the following core points of the tool:

- The **asset modelling and visualisation methodology** that enables the digital representation of the components that comprise the manufacturing floor to be monitored.
- The **Risk Assessment metamodel** which represents the abstract model and rationale followed by the tool for enabling systematically the formation of the assessed environment and the correlation of the assets with the detected vulnerabilities, threats and abnormalities detected.
- The **modelling of Risks, Vulnerabilities and Threats**, along with the definition of the “Attack Scenario” notion which enables the formation of “Asset-Vulnerability-Threat” tuples.
- The documentation of the **attack patterns and classification** considered in the context of the STAR architecture, as a results of the integration of the **CAPEC** (Common Attack Pattern Enumeration and Classification) and **ATLAS MITRE** (Adversarial Threat Landscape for Artificial-Intelligence Systems for the STAR architecture) frameworks.
- The documentation of **Inputs/Outputs** the Risk Assessment and Mitigation Engine.
- The main forms and templates of the **GUI** of the Risk Assessment and Mitigation Engine to enable data entry by the security officer of the manufacturing environment.

We prompt the interested reader to refer to D3.5 for all the necessary details regarding the above-mentioned points of the Risk Assessment and Mitigation Engine. The scope of this deliverable is limited to reporting the updates and the new developments that took place in between the delivery of D3.5 (M17) and the completion of WP3 development actions (M30).

Thus, the following sections elaborate the newly designed features and extensions of the Risk Assessment and Mitigation Engine of STAR, while for purposes of completeness Section 4.2 offers a recap of the architecture of the tool, in line with Figure 11.

## 4.2 Architecture

As illustrated in Figure 11, on the left side the basic management functionalities offered by OLISTIC are given:

- Business Process management
- Asset management
- Vulnerability management
- Threat management
- Mitigation actions management

These management operations work individually but a unified operation is achieved using the OLISTIC backend engine that combines their operation for the provision of the risk assessment functionality.

As depicted, there are five main components that comprise the system, namely:

- OLISTIC Backend;
- OLISTIC Frontend;
- Risk Assessment Engine;
- Asset and Business process management engine;
- Controls Management Engine;
- Notification and Reporting Engine;
- KAFKA

The OLISTIC backend offers the necessary APIs to orchestrate all the backend operations of the engine and works in synergy with the frontend to offer the functionalities. As is documented in Section 4.3.4, the 2<sup>nd</sup> release of the Risk Assessment and Mitigation Engine of STAR comes with a complete documentation of the APIs of the OLISITC backend. The OLISTIC backend is interconnected with all the other internal components, as well as with SQL and NoSQL internal databases used for OLISTIC’s storage and internal data management purposes.

The OLISTIC frontend offers an interactive dashboard which is used for visualising assets taking part in the cyber-physical environment. The dashboard offers management operations for the addition/editing/deletion and the creation of attack scenarios, management of vulnerability and threat profiles of assets, consideration of controls and mitigation actions, the execution of the risk assessment and many others.

OLISTIC is a Quarkus application which incorporates the following enabling technologies:

- **Quarkus:** Kubernetes Native Java stack tailored for OpenJDK HotSpot and GraalVM, crafted from the best of breed Java libraries and standards, and for the design of a reactive application.
- **MongoDB, PostgreSQL** – For the management of both structured and unstructured data objects.
- **Neo4j** - graph database that combines native graph storage, advanced security, scalable speed-optimized architecture for the realisation of the graph-based risk assessment framework.
- **Apache KAFKA** - open-source distributed event streaming platform used for high-performance data pipelines, streaming analytics, data integration through pub/sub model.



that such an operation is completed a specific message is placed in a predefined topic of the pub/sub que of KAFKA.

**The OLISTIC backend** is the Quarkus-based engine that offers the whole functionality of the platform and enabled the interfacing of all the sub-components of the internal OLISTIC architecture, including the interaction with the OLISTIC Frontend.

**The controls management engine** is the one that enables the security officer to manage the life cycle of applying appropriate controls that could mitigate or eliminate a risk state of the monitored environment. Thus, the engine is enriched with off-the-shelf controls that come from the domain knowledge and the experience of the officer or from selected standards of the security and manufacturing domains. It must be stated that in STAR we do not aim to provide any kind of fully automated controls enforcement to the monitored assets of the environment. The controls management engine of OLISTIC aims to support the operator in the life cycle of the controls management and keep track of all controls that have been applied to the underlined systems.

**The OLISTIC Frontend**, is the interactive dashboard that is used for the representation of the vital system and risk information to the security officer. The outputs of the processes supported by the internal components of the OLISTIC are visualized on the dashboard. In addition, the dashboard offers the necessary reactive visual components that can trigger all the necessary functionalities of the OLISTIC Backend.

## 4.3 Features and updates

### 4.3.1 Adoption of CVSS v3.1 with forward and backwards compatibility with all versions of CVSS

The 2<sup>nd</sup> release of the Risk Assessment and Mitigation Engine of STAR addresses the development action for the complete adoption of the CVSSv3. The transition to the latest version of CVSS was a key requirement in order to ensure the alignment of the methodology with the latest standards. In fact, CVSSv2 has been widely criticised by the community for not providing sufficient metrics to distinguish between different types of vulnerabilities. That is, the FIRST<sup>11</sup> working group worked in CVSSv3 to introduce a scoring system that can capture the reality of vulnerabilities more accurately. The two versions have both the three major groups of metrics, i.e., Base, Temporal, and Environmental metrics, but the following updates are introduced in the latest version:

- New scoring values are used in CVSSv3 for the Confidentiality, Integrity, and Availability metrics. More specifically, the values *None*, *Low*, or *High* are used, while in v2 the values *None*, *Partial*, *Complete* were used.
- In CVSSv3 the Attack Vector includes the *Physical* value, which was absent in the previous version. This value indicates that an adversary needs to have physical access to the system/asset in order to exploit it.
- The new version introduces a new option, namely *User Interaction (UI)*, which denotes that successful exploitation of a vulnerability requires a user to take some action before the vulnerability can be exploited. For example, a successful exploit may only be possible during the installation of an application by a system administrator. This metric was not present in the previous version.

---

<sup>11</sup> Common Vulnerability Scoring System SIG <https://www.first.org/cvss/>

- The new version introduces also the *Privileges Required* metric. This metric describes the level of privileges an attacker must possess before successfully exploiting the vulnerability. The possible values of the metric are *None, Low and High*.
- Apart from the additions and changes in the base metrics, the environmental group has been changed. The environmental group completely replaced the Modified Base Score of v2, enabling the risk assessor to modify critical factors of the Base metrics and tailor the scoring to the security situation of her organisation. In this way the assessor can reflect differences between the organisation’s situation and the generic perception reflected by the defined values of the CVSS scores provided by the community.

Overall, despite the notable differences between CVSSv3 and v2, the Risk Assessment methodology adopted in STAR is not affected. As documented in D3.5, our methodology capitalises on the CVSS scoring only for quantifying the severity of vulnerabilities. In fact, CVSS is a model for measuring severity, not risk. Thus, the methodology used in STAR is built on top of CVSS and offers the risk quantification model. Intuitively, the only change in the risk assessment model is the slight aggravation of the risk levels because of the CVSSv3 tendency to provide slightly higher scores than CVSSv2. Overall, it must be noted that the methodology that initially introduced in D3.5 was completely aligned with the CVSSv3 model from the beginning. Thus, no additional modifications needed in the final adoption of the CVSSv3.

In STAR, to enable forward and backwards compatibility of the risk assessment framework with all versions of CVSS, we have developed a new feature in the assessment tool that enables the selection of the CVSS version of a specific vulnerability. This development triggered changes both in the backend and frontend of the system. Figure 13 gives an overview of the graphical interface where the user can select the CVSS profile to be used. Figure 14 depicts the structure of the template which enables the assessor to select between the CVSSv3 and CVSSv2 structure.

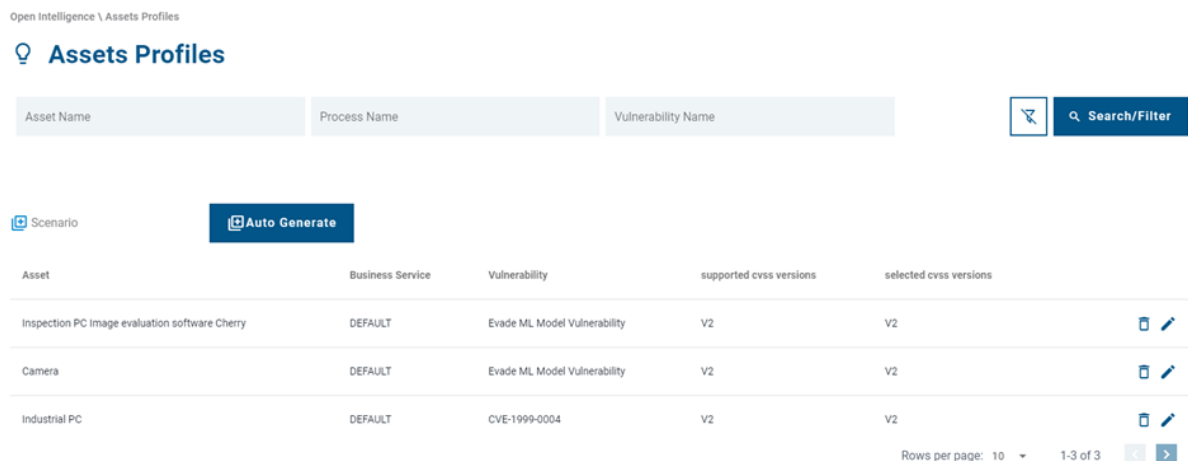


Figure 13 Vulnerability profiles on the assets for CVSS v2.0 and v3.x

Assets Profiles

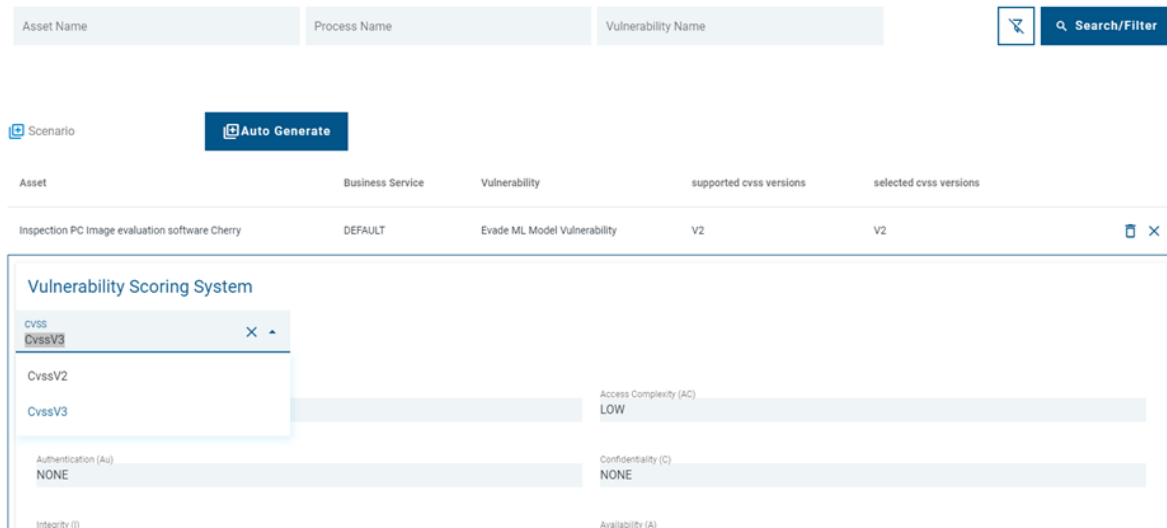


Figure 14 Vulnerability template selection for CVSS v2.0 and v3.x

The forward and backwards compatibility with CVSS version became necessary. This is because there are several vulnerabilities that do not offer CVSSv3 scores. These are mainly older vulnerabilities that were not possible to be refactored once the CVSSv3 was introduced. However, it is known that several legacy systems remain operational and serve as the backbone of infrastructures and especially in the manufacturing environments there are old systems that still support the operation the production lines. That is, in STAR we had to ensure backwards compatibility with the previous version of CVSS. In addition, NVD<sup>12</sup>, the most well-known vulnerability database retired CVSS v2<sup>13</sup> for new vulnerabilities starting from July 2022. This implies that only CVSSv3 will be used in the future. Despite these domain changes, the new developments of STAR enable compatibility under all circumstances and offer an interoperable risk assessment methodology.

It needs to be noted that the development of a new risk assessment methodology is out of the scope of the STAR project. UBITECH provides the OLISTIC enterprise risk management suite to be used as the frontend of the Risk Assessment and Mitigation Engine of STAR, utilising the embedded risk assessment methodology, as has been described thoroughly in D3.5.

### 4.3.2 Automatic generation of Attack Scenarios

The 2<sup>nd</sup> release of the Risk Assessment and Mitigation Engine of STAR comes with an additional feature that aims to increase automation on vulnerability discovery and ease the manufacturing floor life-cycle management on behalf of the security administrator.

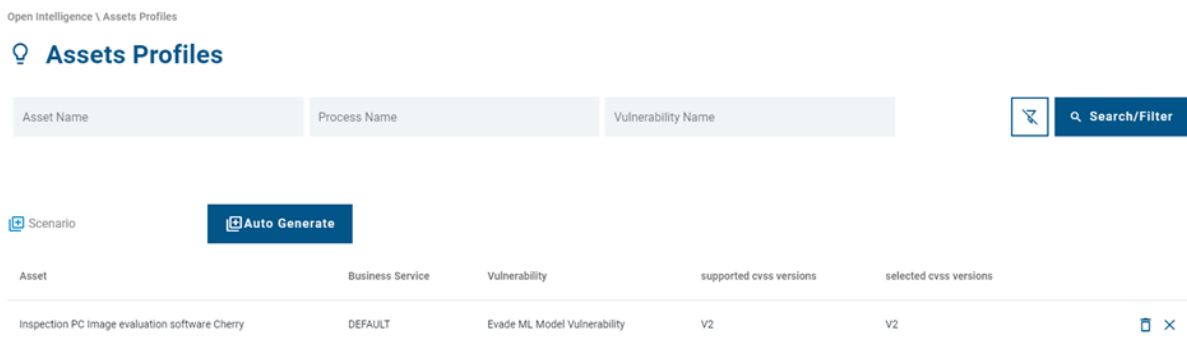
More specifically, under the Assets Profile menu item, the platform UI, and the backend part of OLISTIC, have been extended with the automatic generation of attack scenarios. The auto generation button (see Figure 15) triggers an automated process for discovering vulnerabilities which are applicable to the assets incorporated by the administrator in the manufacturing environment. To take advantage of this automated feature, the administrator needs to document during the asset insertion process the CPE (Official Common Platform

<sup>12</sup> NIST - National Vulnerability Database <https://nvd.nist.gov/>

<sup>13</sup> <https://nvd.nist.gov/General/News/retire-cvss-v2>

Enumeration) of the inserted asset. The CPE provides a standard machine-readable format for encoding names of IT products and platforms. This is exploited by CVE to perform an automated correlation between the CPE IDs and the CVE IDs to identify a mapping between IT products/platforms and existing vulnerabilities.

This feature comes to enhance the existing functionalities of OLISTIC with an automated way of mapping assets to vulnerabilities. It has to be noted that this feature does not substitute the manual operation that the administrator can follow in order to define attack scenarios, while the Security Policies Manager is still in position to trigger the OLISTIC APIs in order to create new attack scenarios upon the detection of attacks against the manufacturing floor assets.



*Figure 15 Automated generation of attack scenarios*

### 4.3.3 Update Look and Feel of the Risk Assessment and Mitigation Engine

The 2<sup>nd</sup> release of the Risk Assessment and Mitigation Engine offers an updated and more modern UI to assist the security officer of the manufacturing floor to perform the necessary actions for the risk management of the monitored environment. An overview of the environment is given in Figure 16 and Figure 17, while the interested reader can refer to Figures 27 and 28 of D3.5 to notice the updated look and feel of the new version.

Note that, the Security Policies Manager comes integrated in OLISTIC, as highlighted in the green box in Figure 16. The updates of the Security Policies Manager are reported in Section 5.



Figure 16 Updated Dashboard UI

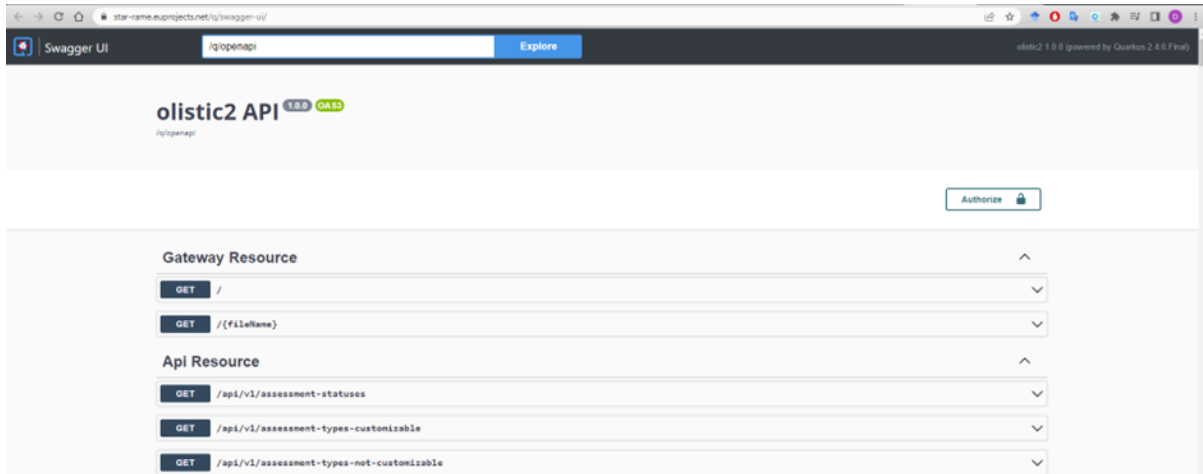
The screenshot shows the 'Risk Assessments' section of the OLISTIC interface. It features a table with the following columns: Name, Status, Type, Author, Business Service, Created, and Completed. There are also icons for actions like edit, delete, and refresh.

Name	Status	Type	Author	Business Service	Created	Completed
SSPM-vhuVDe	Completed	Real	PCL	DEFAULT	2023-06-15 13:03	2023-06-15 13:03
SSPM-aixabm	Completed	Real	PCL	DEFAULT	2023-06-15 12:55	2023-06-15 12:58
SSPM-ajbZiA	Completed	Real	PCL	DEFAULT	2023-04-04 15:26	2023-04-04 15:26
RA1	Completed	Real	PCL	DEFAULT	2023-04-04 12:34	2023-04-04 12:34

Figure 17 Updated Risk Assessment UI

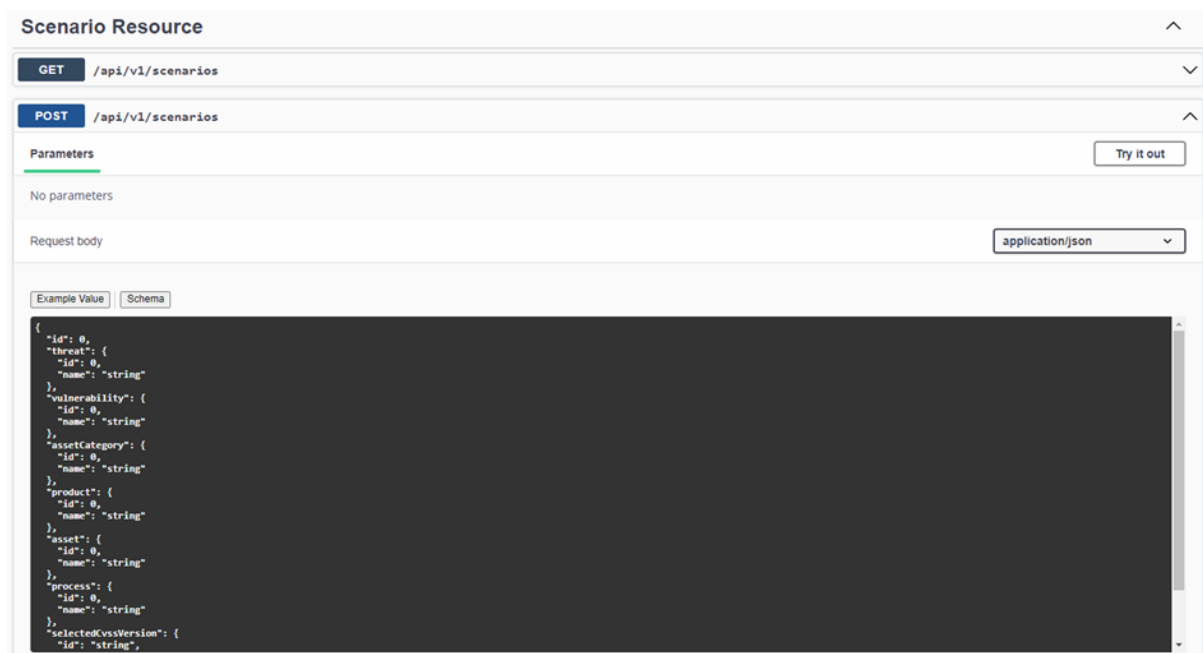
#### 4.3.4 New OLISTIC APIs and documentation

In the context of the action of releasing an updated version of OLISTIC in the context of the STAR project, UBITECH’s technical team released a complete documentation of the APIs of the tool in order to enable full integration with other STAR tools that may need to interact with the Risk Assessment and Mitigation Engine of STAR. The main component interacting with the Risk Assessment and Mitigation Engine of STAR is the SSPM. Thus, in order to enable the complete integration with the SPM and expose the full functionality of OLISTIC, UBITECH offers the complete documentation of the APIs of the tool using swagger. An instance of the Swagger environment is given in Figure 18.



*Figure 18 Instance of the Swagger UI of OLISTIC*

The provided Swagger documents all the necessary APIs, also providing the required structure of the data required by the endpoint to complete the required actions. For instance, Figure 19 documents the POST request that the SPM needs to perform to create a new attack scenario in the OLISTIC environment.



*Figure 19 Endpoint for creating a new attack scenario in the OLISTIC environment*

A complete Swagger UI is uploaded in <https://star-rame.eu/projects.net/q/swagger-ui/>. Given the complete documentation of the OLISTIC APIs, GFT, the responsible partners for the design and implementation of the STAR SPM, was in position of complete the integration with OLISTIC and integrate into the logic of the detection rules creation the necessary APIs calls for the automated creation of attacks scenarios and the remote execution of risk assessments, because of the detection of anomalous incidents in the monitored environment.

## 4.4 Input

As denoted also in the general architecture of Figure 2, OLISTIC receives from the SSPM information related to detected security incidents to feed the risk assessment process. The

security incident includes the necessary information that describes which asset has been attacked and which type of attack/or anomaly has been detected by the monitoring systems. This information is conveyed to OLISTIC by triggering the necessary API endpoints. The inputs sent to the OLISTIC engine have not been updated since D3.5.

## 4.5 Output

The risk assessment is performed on asset basis. In other words, the presence of a vulnerability and/or a threat refer to a specific asset, and thus, a risk level will be associated to that specific asset. The same applies to the controls and mitigation actions that may be applied to a vulnerability or a threat, on an asset.

Thus, overall, the risk assessment outputs a collective report that highlights the risk levels, attack scenarios for each asset, as well as the controls that may have been in place by the assessor to mitigate the generated risk level. This report can be exported by OLISTIC in a human readable format in PDF, while it is also push to the KAFKA component of OLISTIC to be shared with other STAR components which are interested in the risk levels of the assets.

The inputs generated by the OLISTIC engine have not been updated since D3.5.

## 4.6 GUI

The previous subsections have provided several images that correspond to the data templates used in the GUI of OLISITC. As described in Section 4.3.3, the 2<sup>nd</sup> release of the Risk Management and Mitigation Engine comes with an updated look and feel that enables the security officer to interact and trigger the risk assessment on demand and grasp vital information on the security state of the monitored environment.

## 5 Star Security Policies Manager

The following section focuses on the advancements in the SSPM tool development from the previous deliverable version (D3.5), namely:

- A more refined version of the SSPM architecture to include the SSPM GUI to configure security policies and highlight the interactions between the SSPM module, the RMS, the AICD and OLISTIC;
- The development of a User Interface, integrated in OLISTIC, to configure security policies;
- The integration with OLISTIC to create attack scenarios, by exploiting its APIs;
- The interaction with the AI Cyber Defence and the Runtime Monitoring System through the data bus.

### 5.1 Architecture

The SSPM is implemented as a Python application, with a user interface and backend that exploits OPA as an external service for policies evaluation. The SSPM acts as a middle-man, as it aggregates the inputs received from the RMS and the AI Cyber Defence Module, evaluates the received information on the basis of security policies defined by the security officer, and interacts with OLISTIC to create and assess risk scenarios.

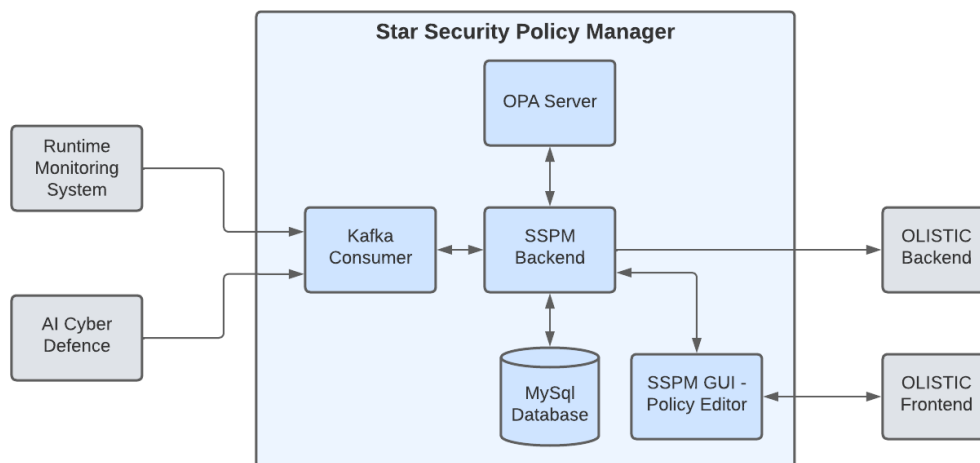


Figure 20: SSPM high level architecture

Figure 20 depicts the architecture of the SSPM, which is composed of 5 modules:

- **Kafka Consumer:** it reads from the specific topics published by the RMS and AICD on the Data Bus and passes the received messages to the SSPM backend;
- **OPA Server:** the policy engine used to evaluate the received inputs against the policies defined by the security officer;
- **SSPM database:** it stores security policies for persistence and SSPM configuration;
- **SSPM backend:** it manages inputs from the RMS and AICD components, interacts with OPA for policies evaluation and calls OLISTIC APIs based on the evaluation results;
- **SSPM GUI:** allows the security officer to create and update policies, and to configure the attack scenario to be created in OLISTIC when a policy has been violated.

SSPM supports the logic for multiple kinds of policies, which can be applied to the following scenarios:

- Poisoning attack detection;
- System CPU workload detection;
- Heavy traffic or other probe’s data that can signal a suspicious behaviour detection;
- Cyberattacks identifications;
- Evasion attacks detection.

## 5.2 GUI and Security Policies definition

Security policy definition is responsibility of the security officer, the SSPM makes available a policy editor where it is possible to define and update the policies used by the OPA engine.

As the SSPM relies on OPA for policy evaluation, security policies are expressed in the REGO language<sup>14</sup>. Policies consist of multiple rules that can also refer to other rules and are generated accordingly to the needs of the use case.

An extensive overview on how OPA and security policies work was provided in this deliverable’s previous version (D5.3), please refer to it for additional information.

To facilitate policies definition to the security officer, who may not be familiar with the REGO language, the SSPM provides a policy editor for the aided creation of simple rules.

The policy editor is hosted in an iFrame on the OLISTIC application, with which the officer interacts, and presents four different views:

- An editor where the security officer can check the active rules and update them using the REGO language;
- A form-based editor for the aided creation of simple rules;
- A tab in which it is possible to define the templates of the attack scenario to be automatically created in OLISTIC when a rule is violated;
- A tab where the connections between rules and attack scenario template are established.

### 5.2.1 Policies editor

The *Rules* view allows the security officer to check the currently active rules, and to update them, if needed, using the REGO format. When the policy is saved, it is automatically updated on the OPA server and saved in the SSPM as well for persistence.

---

<sup>14</sup> <https://www.openpolicyagent.org/docs/latest/policy-language/>

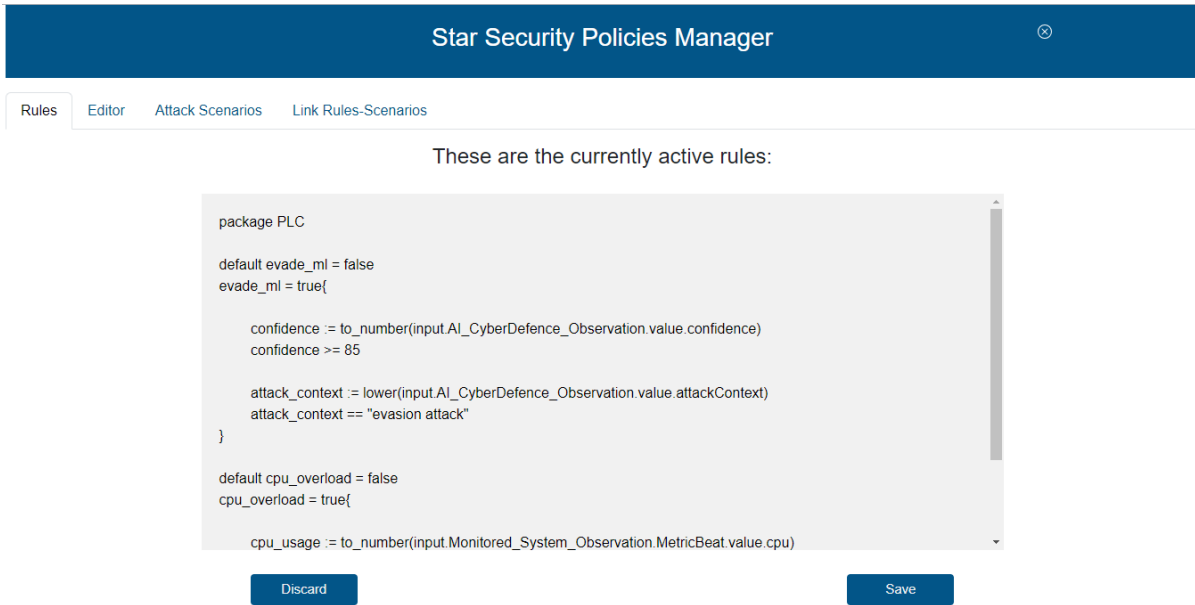


Figure 21 SSPM Policies View

The *Editor* view (Figure 22) enables the creation of simple rules without using the REGO format. It is possible to define multiple rules, each one can comprehend multiple threats. For each rule, the security officer has to define a *Rule Name* which identifies the policy and fill the following fields to define a threat (a rule can comprehend several threats):;

- *Parameter to Check* which can be chosen among the parameters/assets monitored by the RMS and AICD;
- The *Value* of the parameter, which can be a reference value or a threshold;
- And the condition to check (e.g. parameter is equal to reference value).

The Package Name is user specific (i.e.: pilot specific) and cannot be changed.

When the form is submitted the just created policy, with its rules, is translated in REGO format, then updated on the OPA server and saved in the SSPM database.

Figure 22 SSPM Editor View

### 5.2.2 Attack Scenario Templates creation

To configure the SSPM in its interaction with OLISTIC, the security officer must define templates (Figure 23) of an attack scenario that need to be created in case one of the defined rules is violated. An attack scenario is composed by a threat, the asset that is threaten and the vulnerability that is being exploited in the attack. Threat, assets, and vulnerability that can be selected are the ones available in OLISTIC.

#	Name	Threat	Asset	Vulnerability	Delete
1	camera-down	Blockage	Camera	Camera Availability Vulnerability	
2	evasion-attack	Denial of ML Service	AI-based Quality Inspection	CVE-1999-0004	
3	image-injection	Evade ML Model	Camera	Evade ML Model Vulnerability	
4	cpu-overload	Blockage	Industrial PC	System Resources Vulnerability	

Figure 23 SSPM Attack Scenario Templates

### 5.2.3 Link Rules-Scenarios definition

To complete the SSPM configuration, the security officer needs to associate the defined rules and attack scenario templates, this can be done in the tab *Link Rules-Scenarios* (Figure 24).

Attack scenario templates and rule-scenario links are saved in the SSPM database and are read from the SSPM backend whenever a rule is violated to create the corresponding attack scenario in OLISTIC.

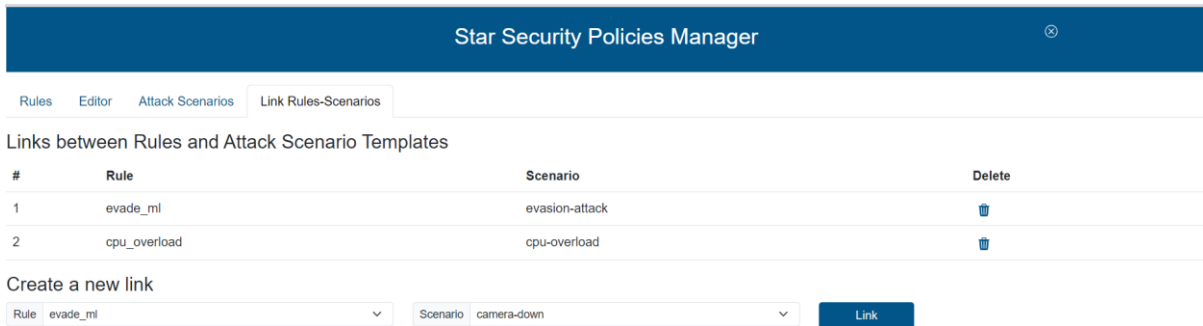


Figure 24 SSPM Link Rules-Scenarios View

### 5.3 Policies evaluation and interaction with OLISTIC

When the RMS and AICD push a message to the data bus, this is read by the SSPM Kafka consumer and passed to the SSPM backend. The message is sent to OPA for evaluation and if one or more of the active rules is violated, it may mean that one of the production line’s assets is in an abnormal state and that an attack may be ongoing. To alert the security officer that will ultimately evaluate the situation, a cascade of events is triggered in OLISTIC to analyse the status of the production line.

SSPM is always listening to the STAR data bus and each time a new message is received, this is forwarded to OPA as a JSON object. OPA assesses whether the message received violates one or more of the active rules and sends back its response.

If no rule was violated, the backend performs no action and resumes listening. If a rule has been violated, instead, it starts the process to create an alert in OLISTIC. The SSPM authenticates and interacts with OLISTIC using its APIs.

The SSPM checks in the database which attack scenario is linked to the violated rule and, if it does not exist yet, proceeds on creating it (e.g. last scenario in Figure 25). Afterwards it rises the threat probability of the threat in the attack scenario just created (Figure 26) and, eventually, creates and fires a risk assessment (Figure 27). The security officer is then alerted by OLISTIC and can check the risk assessment result on the platform dashboard (described in section 4.3.3).

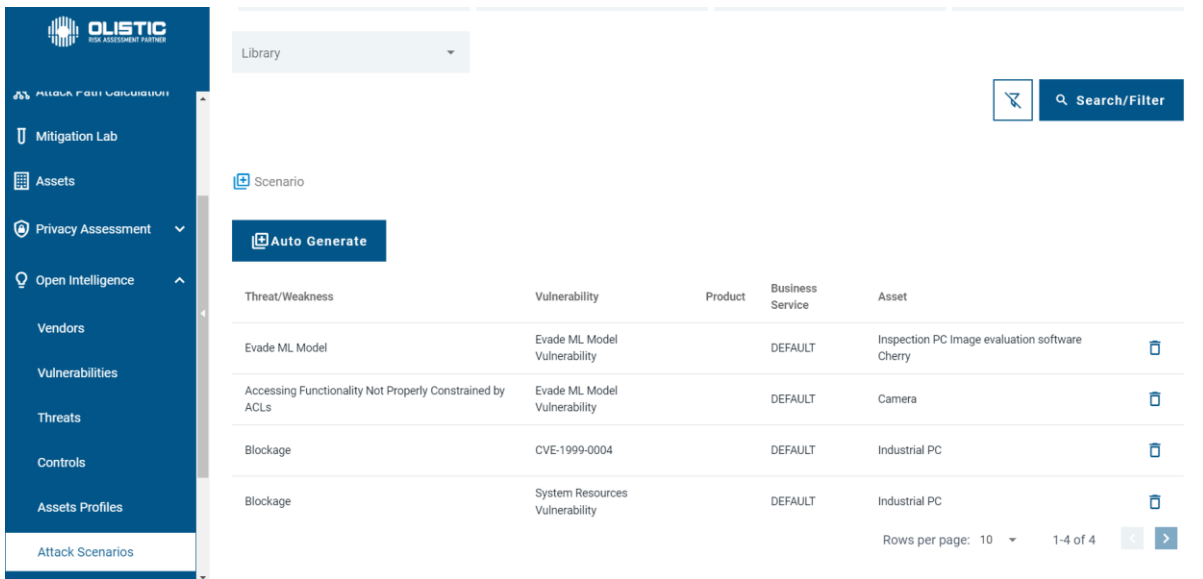


Figure 25 Attack Scenario created in OLISTIC

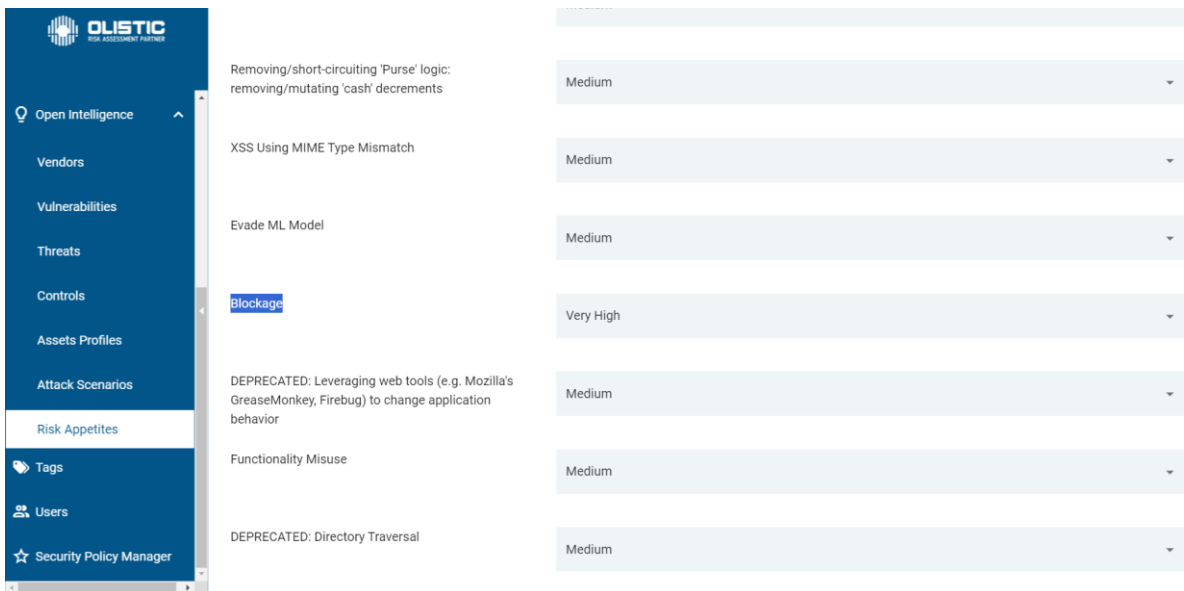
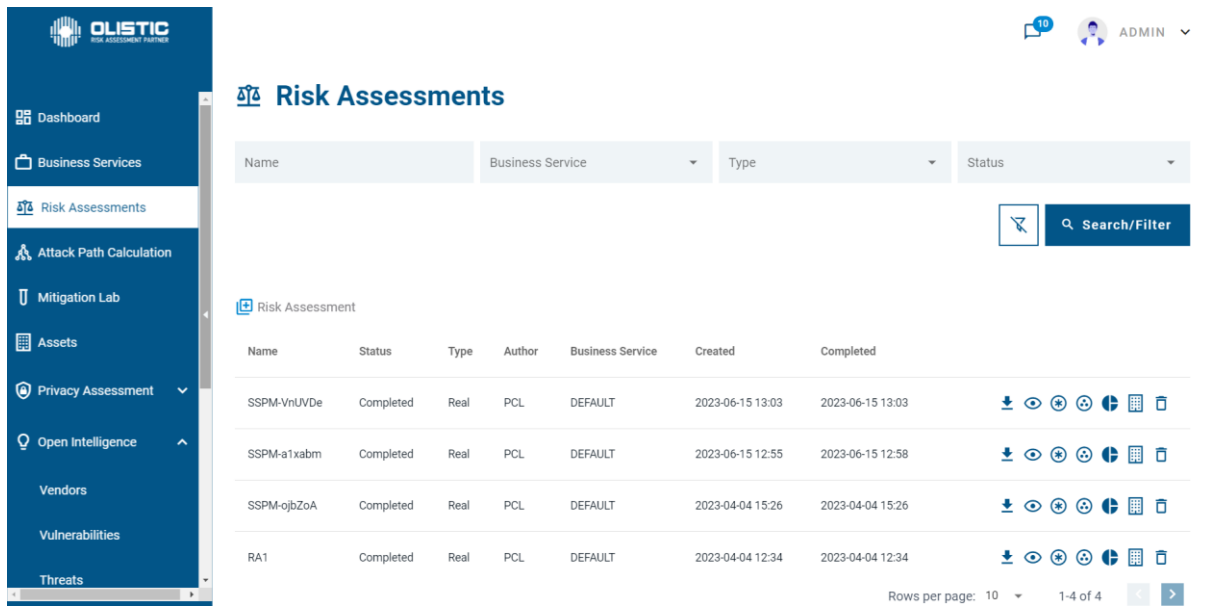


Figure 26 Raised Risk Appetite in OLISTIC



The screenshot shows the OLISTIC Risk Assessments interface. On the left is a navigation menu with options: Dashboard, Business Services, Risk Assessments (selected), Attack Path Calculation, Mitigation Lab, Assets, Privacy Assessment, Open Intelligence, Vendors, Vulnerabilities, and Threats. The main content area is titled 'Risk Assessments' and features a table with columns: Name, Business Service, Type, Status, Created, and Completed. There are also filters for Name, Business Service, Type, and Status, and a 'Search/Filter' button. The table contains four rows of risk assessments, all with a status of 'Completed'.

Name	Status	Type	Author	Business Service	Created	Completed	
SSPM-VnUVDe	Completed	Real	PCL	DEFAULT	2023-06-15 13:03	2023-06-15 13:03	[Icons]
SSPM-a1xabm	Completed	Real	PCL	DEFAULT	2023-06-15 12:55	2023-06-15 12:58	[Icons]
SSPM-ojbZoA	Completed	Real	PCL	DEFAULT	2023-04-04 15:26	2023-04-04 15:26	[Icons]
RA1	Completed	Real	PCL	DEFAULT	2023-04-04 12:34	2023-04-04 12:34	[Icons]

Rows per page: 10 | 1-4 of 4

*Figure 27 Risk Assessment created and fired in OLISTIC*

Through this process, the Star Security Policies Manager enables the automation of assets monitoring, eliminating the need of security officer constant presence. Key assets in the production line are constantly monitored and their parameters and behaviours are continuously compared to reference values defined by the security officer by means of security policies. Suspicious conditions are automatically detected and brought to the security officer’s attention, which will then evaluate the situation and put in place mitigation actions if needed.

## 6 Security and Data Governance in Pilots' Environments and threat landscape analysis

The following paragraph is continuing the effort of D3.5 on the surveys to collect all the useful information about the UCs related to STAR Pilots. Aim of the current section is to implement the final layout of the tools, further developed in WP6 to create the final version of the tools applied to the specific pilots' conditions.

The technical partners of WP3 conducted an exercise with the use case partners of WP6 to assess the applicability of WP3 tools, as described in this deliverable, within the context of the Security & Data Governance solution of STAR. The primary objective was to determine how the Pilots could utilize these tools to effectively achieve their operational objectives. The exercise aimed to converge on realistic attack scenarios and identify the appropriate placement of the WP3 tools within the STAR pilots, initially addressing the WP3 context and paving the way for further actions to be taken in the WP6 context.

To validate the effectiveness of the Security & Data Governance solution and demonstrate the end-to-end usage of the system, a questionnaire was meticulously crafted and circulated among the Pilot partners. This questionnaire sought to capture vital information about their respective manufacturing environments, critical assets of production lines, potential vulnerabilities, and threats that could pose risks to their systems. Additionally, it aimed to identify possible abnormalities that might occur during the manufacturing process. The main goal of this process was to pinpoint realistic attack scenarios that closely align with the unique environments of the pilots."

The Pilot's use cases identified to apply and evaluate the WP3 tools are:

- Philips Pilot: AI cyber-defence and decentralized reliability for industrial data.
- DFKI Pilot: Robot Reconfiguration based on the Dynamic Layout.
- IBER Pilot: Secure and human centred AI systems for agile production operations.

More specifically, in the administered questionnaire the pilots were asked to provide specific information about the chosen use case. The information requested was a description of the environment setup, comprising of the:

- Topology of the manufacturing floor or the production line.
- Assets dependencies.
- Known vulnerabilities of asset

Given the above, the Pilots were asked to use the RAME Star tool to create the digital reflection of the environment.

Along with the above-mentioned information, the Pilot partners were asked to determine:

- Possible adversarial scenarios (current, previously witnessed, or hypothetical) against their production lines.
- A list of critical indicators/measurements/values that need to be monitored to identify potential misbehaviours/abnormalities in the production process.
- Possible integration points (APIs) for realising the tools' pipeline between the pilots and the WP3 tools. (Note that this is an ongoing action in the context of WP6)

All the above, will help the WP3 and WP6 partners to define the relevant security policies that will be synthesised in the context of the SPM tool which will combine the data collected from the various indicators that will reveal potential abnormal behaviours in the production lines.

A template of the questionnaire is available in Appendix A.

The subsections below highlight the information acquired from the questionnaires for each Pilot, with a focus on the identified scenarios, while indicative security policies are given. It is expected that the final set of the security policies will be defined in the context of the WP6 actions.

It must be state that in most cases no previous attacks were identified and reported by the pilot partners, as their environments are detached from the external world. However, the WP3 and WP6 partners worked for the formation of hypothetical scenarios that fit to nature of the monitored production lines.

## 6.1 Philips Pilot

### 6.1.1 Environment setup

The topology of the cherry inspection station is provided, along with a description of the assets and their dependencies, as can be seen in Figure 28 and Table 3 and Table 4. The goal of the manufacturing line is the visual quality inspection, performed by 9 cameras, taking shots from deferent angles to uncover different faults on the production.

The Philips factory in Drachten, is an advanced factory for the mass manufacturing of consumer goods (e.g., shavers, OneBlade, baby bottles, and soothers). Current production lines are often tailored for the mass production of one product or product series in the most efficient way. However, the manufacturing landscape is changing. Due to global shortages, manufacturing assets and components are becoming scarcer, and a shift in market demand requires the production of smaller batches more often. To adhere to these changes, production flexibility, re-use of assets, and a reduction of reconfiguration times are becoming more important for the cost-efficient production of consumer goods. In this context, one of the topics currently investigated within Philips is the painless setting up of the automated AI-based quality inspections that aim to make the reconfiguring of quality control systems faster and easier.

The setup used to inspect the quality of the printed logos on the Philips products or to detect faulty soother products. In this setup, the system comprises a few different assets, including a camera, a robot and an inspection PC. The decision on the quality of the product is made by a quality control algorithm running on the industrial server based on the DNN-based system destined to classify the images captured by the inspection camera. The CNN-based classifier has been trained based on manually collected historical data reflecting all possible flaws generated by the automated manufacturing line. During production time, the captured images are given as input to the pre-trained CNN-based quality inspection system to perform the necessary checks on the quality of the product. Based on the decision taken by the latter app, the product will continue its journey to the factory's assembly lines, or it will be discarded. One can easily understand that the quality inspection system is a mission-critical asset in the production workflow to ensure the high quality of the end products and avoid the undesirable assembly or delivery of faulty parts. This is also translated into increased production cost and the waste of essential time due to unnecessary machine engagement, slowing down the production rates.

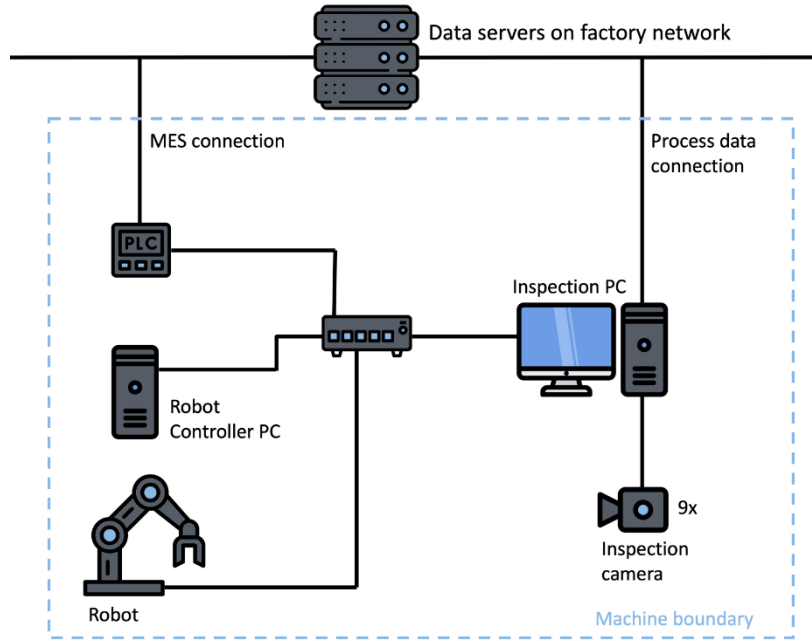


Figure 28 Topology of the cherry inspection station

Table 3 Assets of the cherry inspection station

Asset_ID	Name	Short description	Technical Details	Asset Category
<u>PCL.PADAS.1</u>	Industrial PC	A computer that provides a user interface & runs the required software	Version CPE Type Model No.	Proprietary Hardware
<u>PCL.PADAS.2</u>	Industrial PC OS	Operating system installed on computer	Version: Windows 10 Service pack: Enterprise N	Proprietary Software
<u>PCL.PADAS.3</u>	AI Algorithm for quality control	AI algorithm in python	Python 3.7 and libraries to run AI algorithm	
<u>PCL.PADAS.4</u>	Camera with telecentric lens	Camera that makes the quality inspection image when triggered by the printer PLC	Camera: Basler ACA3800 -10gm  Lens: Opto Engineering Telecentric Lens TC12192	Proprietary Hardware
<u>PCL.PADAS.5</u>	Firmware camera	Firmware to control the camera	Pylon provided by Basler	Proprietary Software
<u>PCL.PADAS.6</u>	Programmable LED controller + lighting	LED controller that controls the lighting needed for the quality inspection image	CCS PD3-5024-4-EI	Proprietary Hardware

<u>PCL.PADAS. 7</u>	LED controller protocol/software	The LED controller is able to receive encoded bytes	Python socket module TCP/IPv4 protocol UTF-8	
<u>PCL.PADAS. 8</u>	Printer PLC	PLC that provides a trigger to the camera to make an inspection image	Connected with simple on/off signal	Proprietary Hardware
<u>PCL.PADAS. 9</u>	Database	Storage of quality inspection images	Local storage on computer	
<u>PCL.PADAS. 10</u>	External server	For remote access	-	
<u>PCL.PADAS. 11</u>	Other computer software	To run the system	Pypylon by Basler, Python	
<u>PCL.INSAS.01</u>	Camera with lens (9x)	Camera that makes the quality inspection image when triggered by the printer PLC	HARD VCXG-51M NR.11165952-BAUMER  Sony IMX264 sensor	Hardware
<u>PCL.INSAS.02</u>	Firmware Camera	Firmware to control the camera	Unknown	Software

*Table 4 Relationships of the assets of the cherry inspection station*

Asset_ID_X	Relation	Asset_ID_Y	Details
<u>Printer PLC (8)</u>	Is_connected_to	<u>Camera (4)</u>	On/off signal / power
<u>Camera (4)</u>	Is_connected_to	<u>Computer (1)</u>	Connected via Ethernet, GigE vision
<u>Windows 10 OS (2)</u>	Is_installed_on	<u>Computer (1)</u>	
<u>AI Algorithm (3)</u>	Is_running_on	<u>Computer (1)</u>	
<u>Firmware camera (5)</u>	Is_installed_on	<u>Camera (4)</u>	Pylon by Basler
<u>Other Computer SW (11)</u>	Is_running_on	<u>Computer</u>	Pypylon by Basler, Python
<u>Computer (1)</u>	Is_connected_to	<u>External server (10)</u>	Connected via Ethernet
<u>LED Controller (6)</u>	Is_connected_to	<u>Computer (1)</u>	Connected via Ethernet, running according to specs (7)
<u>Quality control images &amp; results (9)</u>	Is_stored_on	<u>Computer (1)</u>	Current situation: images stored in folders; results stored in log files
<u>External server (10)</u>	Is_connected_to	<u>Computer (1)</u>	

A digital representation of the environment was created using OLISTIC tools (Figure 29). This digital representation materialises the environment illustrated in Figure 28. Figure 29 OLISTIC digital representation of the Philips environment, considering the interconnections reported in Table 4. Table 3 Assets of the cherry inspection station.

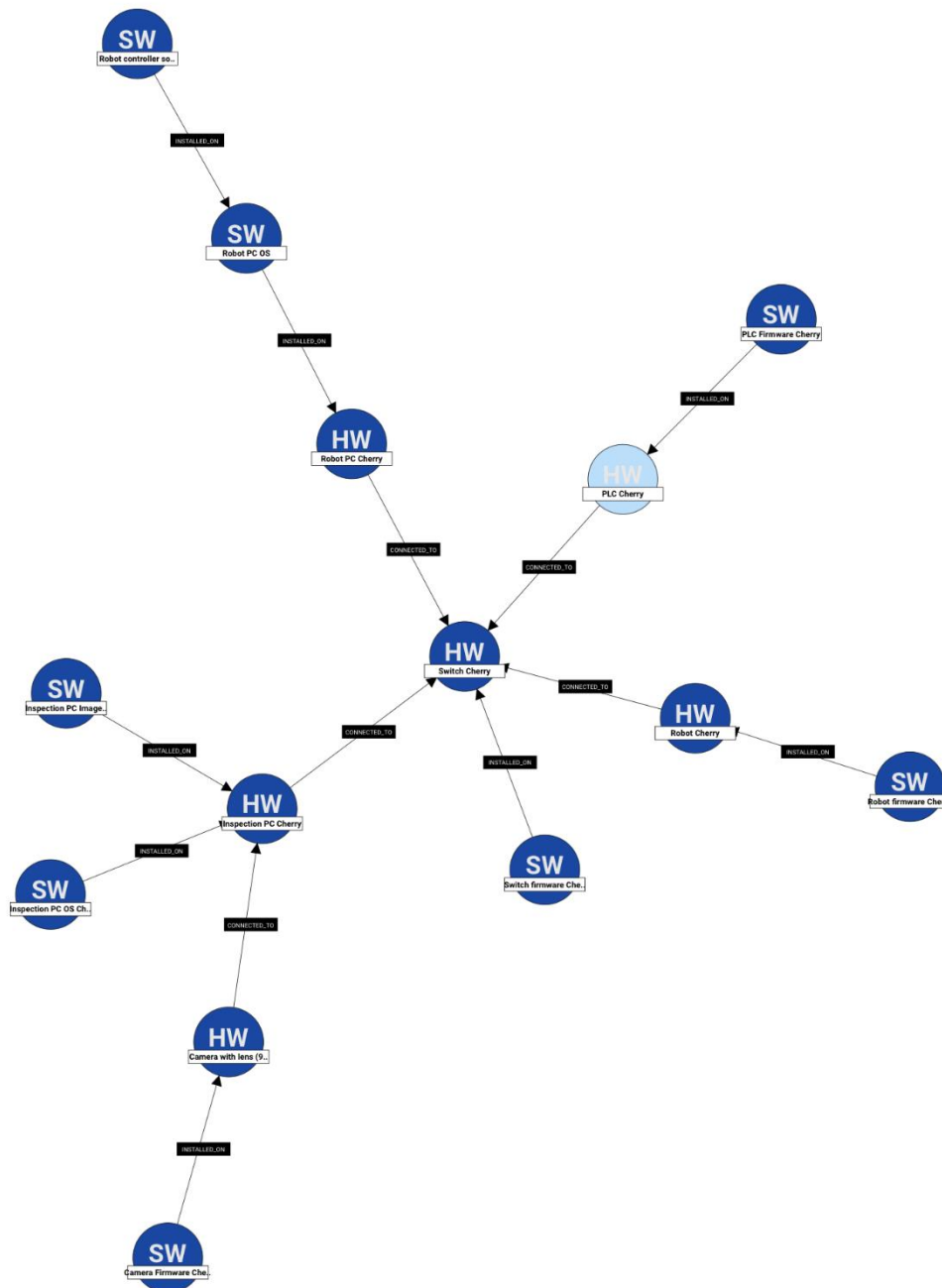


Figure 29 OLISTIC digital representation of the Philips environment

### 6.1.2 Current status and previously witnessed security faults

Based on the bilateral discussion between the WP3 partners and PCL, only one previously faced attack was identified. In the past an attacker was able to fire a Denial of service (DoS) attack against the IP camera of the visual inspection system, setting the production line out of order.

### 6.1.3 Hypothetical scenarios

Given the analysis the technical partners have identified two hypothetical, but plausible, scenarios that document and focus on different abnormal behaviours in the PCL’s production environment.

#### 6.1.3.1 Scenario 1

**Description:** The computer mentioned under nr. *PCL.INSAS.03* plays a central role in the cherry inspection machine. If an adversary is able to make a connection with the computer, he can reach all vital systems of the machine. In practice, this is possible if an adversary who is working in the area gets access to the machine. The goal is to disrupt the normal operation of the machine.

**Abnormal behaviours:**

- The production rate determined by robot handling is known and always within specific boundaries. Any deviation of this performance metric can indicate something is wrong.
- The evaluation time of the images by the AI is also known within boundaries. Deviation of this performance metric can indicate something is wrong.

#### 6.1.3.2 Scenario 2

**Description:** The PLC mentioned under nr. *PCL.INSAS.07* handles all operational logic within the inspection machine. If an adversary is able to make a connection with the PLC, he can change the behaviour dramatically. In practice, this is possible if an adversary who is working in the area gets access to the machine. The goal is again to disrupt the normal operation of the machine.

**Abnormal behaviours:**

The production rate is known as it is the sequence of motions within the machines operation. Any change in the behaviour of the machine will indicate it is tampered with.

### 6.1.4 Critical Assets

Based on the hypothetic scenarios and the environment analysis, Philips has identified a set of critical assets:

- **Industrial PC:** It is a critical resource as it hosts the algorithms that perform the quality inspection.
- **Switch:** It is a critical resource as it provides all internal communication in the machine.
- **Camera:** It is a critical resource as it produces the images based on which the visual inspection process takes place. A potential malfunction on the camera can cause financial loss.
- **PLC:** It is a critical resource as it provides all logic for operating the machine.
- **Robot controller:** It is a critical resource as it manages robot behaviour.
- **Robot:** It is critical resource because it physically manipulates the products.

### 6.1.5 Properties to be monitored

Based on the analysis performed, the WP3 technical partners and the PCL has identified a set of key indicators that can be monitored in the production environment and could reveal potential malfunctions or attacks against the above-mentioned critical assets.

- **Images generation rate:** e.g., 50 pictures per minute;
- **Device resources:** e.g., Average CPU/GPU/RAM utilization;
- **Machine Cycle:** e.g., Cycle per hour of the Cherry inspection machine.
- **Device availability:** e.g., ping an exposed endpoint to identify availability.

Asset	How to access data	Values/Properties to monitor	Normal values and abnormal thresholds
Camera	Push data to RMS provided API.	Images generation rate	The images generation rate should be fixed: 50 images per minute. Any discrepancy is an indication of abnormal behaviour.
Average CPU/GPU utilization	Push data to RMS provided API.	CPU/GPU utilization percentage	The CPU/GPU utilization should not exceed 80% per hour
Cherry inspection machine	Push data to RMS provided API.	Machine cycle	
Robot	Ping an exposed Robot API and Push data to RMS provided API.	Availability	Robot should always be available and respond to a ping

### 6.1.6 Scope of AI Cyber Defence tool in the Philips pilot.

The focal point in the context of the PCL pilot is the visual quality inspection setup. Considering the threat landscape, the AI-based visual inspection system, the WP3 partners will focus on the detection of poison and evasion attacks that may threaten the model that empowers the aforementioned system. Thus, apart from monitoring the various indicators that can reveal abnormal events, we will consider that an adversary can gain access to the infrastructure (it could be also acting as an insider) so that to inject perturbed images (instances) in order to lead the DNN-based quality inspection model to misclassifications.

The model is trained based on historical datasets, compiled under human supervision. The trained model is then placed manually in the file system of the industrial server, and the CNN-based quality inspection app puts the model in the inspection workflow.

Thus, the AI Cyber defence tool, which is documented in D3.3 and D3.4, will be focused on detecting the following AI-based attack variations:

- **Evasion Attacks:** We consider adversaries, under the notion of ML model evasion attacks, who can craft adversarial data which can lead a machine learning model to identify the contents of the data incorrectly. In this context, the adversary tries to

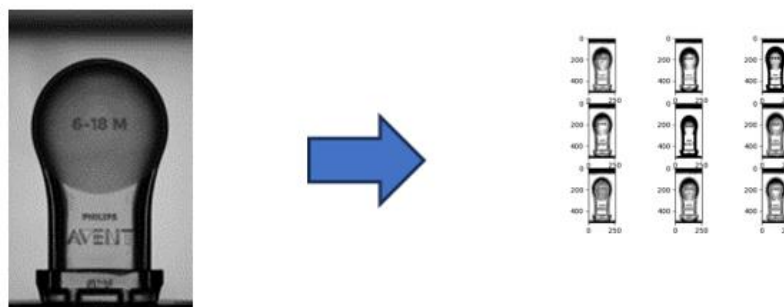
manipulate systems and data and evade the model during the inference mode. For instance, the adversary may exploit a vulnerability on the visual inspection camera and compromise the integrity of the captured data by manipulating the operational behaviour of this business resource. Under this adversarial approach, the corresponding business processes can look fine, but may have been altered to benefit the adversaries' goals.

- **Poisoning attacks:** In addition, we consider an adversary who may attempt to poison the target model and craft adversarial data to feed the DNN model and obtain the intended result. This approach will enable the adversary to create “vulnerable” trained models using data that may not be easily detectable during the training phase.

In this context, the attacks and defences will be generated and evaluated in the context of the PCL pilot. Our aim will be to identify the model and strategy that will be interposed in the dataflows of the training or the testing phases to try to sanitize the data pipelines by filtering out malicious instances or by detecting the injection of adversarial examples in the process.

Thus, in this context UBITECH team will be focused on analysing the image datasets of Philips for creating malicious instance and evaluating defences. The image below shows how the soother images are utilised in order to create adversarial instances and evade the AI-based quality inspection model.

The interested reader can refer to D3.4 for getting more details on the experimental results of the AI Cyber Defence tool of STAR when using the datasets from the Philips pilot.



*Figure 30 Generation of adversarial examples (deepfool attack) for the soother dataset.*

### 6.1.7 Scope of Runtime Monitoring System tool in the Philips pilot.

The scope of RMS in the Philips pilot is to collect the identified monitored properties, using the methods specified in section 6.1.5 above, which are received in a proprietary format. Then the data are filtered based on the abnormal thresholds specified and transformed using the Observations data model which are pushed to the Kafka bus to be consumed by the Security Policies Manager.

### 6.1.8 Scope of Security Policies Manager in the Philips pilot.

Analysing the information provided by the pilot on the environment and the previously faced faults, few initial policies have been created for Scenario 1. Policies to be applied in Scenario 2 will be identified in the context of WP6. Also, initial rules identified for Scenario 1 will be further refined in WP6. Most appropriate threats and vulnerability to create an attack scenario for each policy will be identified among the ones available in OLISTIC (if no appropriate threat or vulnerability is available, custom ones will be created).

### 6.1.8.1 Policies for Scenario 1

- A. **IF** (camera generation rate > 50 images/min) **THEN** potential adversarial attack ongoing.
- B. **IF** (CPU load >90%) **THEN** potential DoS attack ongoing.
- C. **IF** (EVASION\_ATTACK detected with 80% confidence) **THEN** potential evasion attack ongoing.

In Table 5 each rule is associated with a Threat, a Vulnerability and an Assets, these three elements are used to create an attack scenario on OLISTIC.

*Table 5 Attack Scenarios for Policies for Philip's Scenario 1*

Rule	Threat	Vulnerability	Asset
A	Evade ML Model	Evade ML Model Vulnerability	Camera
B	Denial of ML service	System Resource Vulnerability	Industrial PC
C	Evade ML Model	Evade ML Model Vulnerability	TDB

## 6.2 DKFI Pilot

### 6.2.1 Environment setup

The purpose of the pilot is to design an optimal path for the robotino so that to navigate through the manufacturing floor by avoiding obstacles and keeping workers safe. In this context, the key focus is on manufacturing customized products in agile production systems, allowing for greater efficiency, speed, and maintenance monitoring. The automotive industry's increasing demand for cockpit customization has led to changes in the plastic parts and components market, requiring faster reconfiguration of production processes to meet small and large series production needs (customized mass production). The topology of the production line is provided, along with a description of the assets and their dependencies.

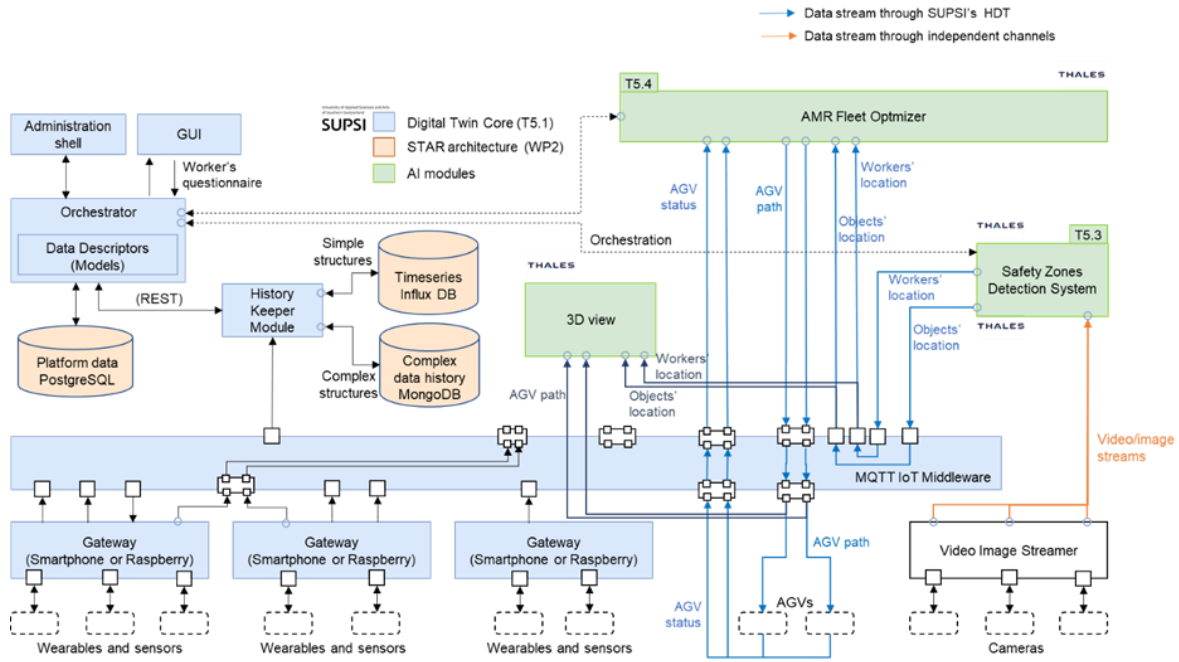


Figure 31 DFKI Environment Setup

The following table and figure describe the assets of DFKI’s pilot.

Table 6 Assets of DFKI's pilot

Asset_ID	Name	Short description	Technical Details	Asset Category
DFKI.Camera.1	Cameras	Two stationary ceiling Cameras	M3066-V	Hardware
DFKI.Camera.2	Axis Device Manager	Two connect to Cameras		Software
DFKI.Camera.3	Relay	30 minutes Relay to connect to network		Hardware
DFKI.Robotino.4	Robotino	An omni-directional mobile robot Manufacturer:Festo	Model 3	Hardware
DFKI.Robotino OS.5	Ubuntu	Robotinos Operating System	16.04	Operating System
DFKI.ROS.6	ROS	Open Source Robot Operating System, ROS Kinetic version	Runs in Ubuntu 16.04	Robot Operating System
DFKI.ros_mqtt.7	ros_mqtt	Interface between ROS1 and mqtt	Runs in ROS	Software Interface
ID6	Apple Watch	Wearable IMU sensors	iOS	Hardware

<i>ID7</i>	iPhone	Wearable IMU sensor and processor	iOS	Hardware
<i>DFKI.Server.8</i>	Server	As the edge server		HW
<i>DFKI.Server_Ubuntu.9</i>	ubuntu	OS	20.0	SW
<i>Safety_Zones_Detection_Systems</i>	Video Analytics component	Detect and locate human and object present in the factory	Docker/ Ubuntu18.04	SW
<i>AMR_Fleet_Optimizer</i>	AMR Controller	Compute dynamic path that avoid obstacles	Python code	SW

*Table 7 Relationships of the DFKI's assets*

<b>Asset_ID_X</b>	<b>Relation</b>	<b>Asset_ID_Y</b>	<b>Details</b>
<i>DFKI.Camera_1</i>	Is_connected_to	<i>Safety_Zones_Detection_Systems</i>	Safety zones Detection systems grabs video streams in order to analyse the scene (human and object detection)
<i>DFKI.Robotino.4</i>	Is_used_by	<i>DFKI.ROS_3</i>	Robotino is controlled via ROS packages to achieve desired behaviour
<i>DFKI.RobotinoOS.5</i>	Is_installed_on	<i>DFKI.Robotino.4</i>	Ubuntu 16.04 is installed in local PC of Robotino.
<i>DFKI.ROS.6</i>	Is_installed_on	<i>DFKI.RobotinoOS.5</i>	ROS Kinetic runs on Ubuntu 16.04, which enables control of Robotino and its components
<i>DFKI.ros_mqtt.5</i>	Is_connected_to	<i>DFKI.ROS.6</i>	It is an interface between ROS messages and mqtt type of messages
<i>DFKI.ros_mqtt.5</i>	Is_connected to	Thales.MQTTIot Middleware_ID	Reads mqtt format and translates into ROS messages
<i>Asset_ID_5</i>	Is_connected_to	Asset_ID_6	Transmitting the sensor data acquired by Apple Watches to iPhone
<i>DFKI.Camera.3</i>	Is_connected_to	<i>DFKI.Camera.1</i>	Relay will connect/disconnect the Cameras from network infrastructure
<i>DFKI.Camera.1</i>	Is_used_by	<i>DFKI.Camera.2</i>	To capture the frames from cameras.
<i>Safety_Zones_Detection_Systems</i>	Is_connected_to	Thales.MQTTIot Middleware_ID	Extract and send human and object positions through MQTT middleware.
<i>AMR Fleet Optimizer</i>	Is_connected_to	Thales.MQTTIot Middleware_ID	Move Commands are sent to robotino

A digital representation of the environment was created using OLISTIC tools (Figure 32). This digital representation materialises the environment illustrated in Figure 31. Figure 29 shows the OLISTIC digital representation of the Philips environment, considering the interconnections reported in Table 3. Assets of the cherry inspection station.

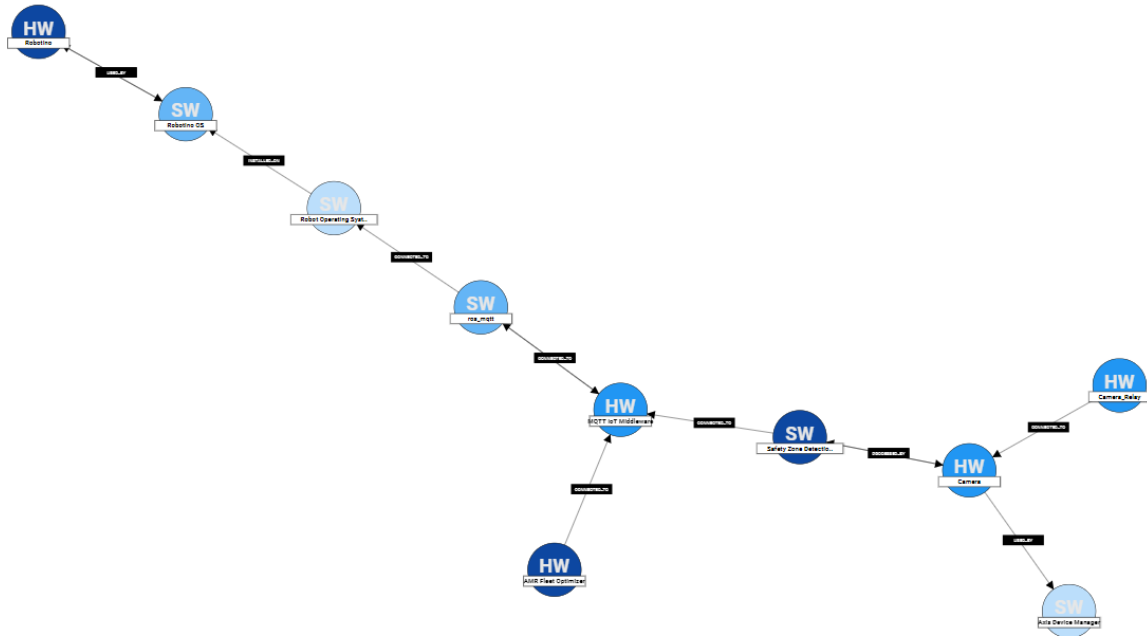


Figure 32 OLISTIC digital representation of DFKI's environment

### 6.2.2 Current status and previously witnessed security faults

The DFKI working team worked with the WP3 partners in order to collect information regarding past or potential incidents that might affect the setup of the pilot. It has to be stated that no past security and production faults were identified as the topology is an isolated environment used for experimental purposes. However, DFKI recognised some vulnerabilities that could potentially affect the systems of the above-mentioned topology (Table 8).

Table 8 DFKI identified potential vulnerabilities

Vulnerability ID	Affected Assets	Source	Description	
CVE-2022-48217		<a href="http://www.cvedetails.com">www.cvedetails.com</a>	** DISPUTED ** The tf_remapper_node component 1.1.1 for Robot Operating System (ROS) allows attackers, who control the source code of a different node in the same ROS application, to change a robot's behavior. This occurs because a topic name depends on the attacker-controlled old_tf_topic_name and/or new_tf_topic_name parameter. NOTE: the vendor's position is "it is the responsibility of the programmer to make sure that only known and required	<a href="https://github.com/tadr-project/tf_remapper_cpp/issues/1">https://github.com/tadr-project/tf_remapper_cpp/issues/1</a>

			parameters are set and unexpected parameters are not."	
CVE-2022-35612	<a href="#"><i>DFKI.ros_mqtt_5</i></a> , <a href="#"><i>Thales.MQTTIoTMiddleware_ID</i></a>	<a href="#">CVE-2022-35612</a>	A cross-site scripting (XSS) vulnerability in MQTTRoute v3.3 and below allows attackers to execute arbitrary web scripts or HTML via a crafted payload injected into the dashboard name text field.	
CVE-2020-10279	DFKI.ROS_3, DFKI.Robotino_2	<a href="#">CVE-2020-10279</a>	UBports Ubuntu Touch 16.04 allows the screen-unlock passcode to be used for a privileged shell via Sudo. This passcode is only four digits, far below typical length/complexity for a user account's password. NOTE: a third party states "The described attack cannot be executed as demonstrated.	
CVE-2023-1006	<a href="#"><i>DFKI.Robotino_2</i></a> , <a href="#"><i>DFKI.RobotinoOS_3</i></a>	<a href="#">CVE-2023-1006</a>	A vulnerability was found in SourceCodester Medical Certificate Generator App 1.0. It has been classified as problematic. This affects an unknown part of the component New Record Handler. The manipulation of the argument lastname with the input "><script>prompt(1)</script>" leads to cross site scripting. It is possible to initiate the attack remotely. The associated identifier of this vulnerability is VDB-221739.	

### 6.2.3 Hypothetical scenarios

Given the analysis the technical partners have identified an hypothetical scenario that document and focus on different abnormal behaviours in the DFKI’s production environment.

#### 6.2.3.1 Scenario

**Description:** During the analysis of DFKI’s environment, it became clear that the Robotino was the asset that had to be protected and it was a vital source for sensory data that could reveal possible abnormalities on the operational profile of the Robot.

Thus, Robotino has been reported as the critical asset, while the following indicators can be used in order to monitor the behaviour of the Robot while it is maneuvering throughout the manufacturing floor in order to achieve the operational goal of the pilot. Here is a list of the indicators that can be monitored in the context of the DFKI pilot.

#### 6.2.4 Properties to be monitored

Based on the analysis performed, the WP3 technical partners and DFKI have identified a set of key indicators that can be monitored in the production environment and could reveal potential malfunctions or attacks against the production line’s critical assets.

- **Robotino speed:** e.g., The robot speed should not exceed 10 [KM/H];
- **Robotino battery status:** e.g., The battery cannot be drained below 5%;
- **Robotino bumper sensor:** e.g., the bumper should not be activated.

*Table 9 Assets of DFKI's pilot*

Asset	How to access data	Values/Properties to monitor	Normal values and abnormal thresholds
Robotino Speed	Push data to WP3 Kafka bus	Robot Speed	The robot speed should not exceed 10 [KM/H]
Battery Status	Push data to WP3 Kafka bus	Voltage values of Battery	The battery cannot be drained below 5%
Proximity or bumper sensor (on/off)	Push data to WP3 Kafka bus	Bumper Sensor	the bumper should not be activated.

### 6.2.5 Scope of AI Cyber Defence tool in the DFKI pilot.

The AI Cyber Defence tool will not be used in the DFKI Pilot.

### 6.2.6 Scope of Runtime Monitoring System tool in the DFKI pilot.

The scope of RMS in the DFKI pilot is to collect the identified monitored properties, which are pushed from DFKI services to the Kafka bus in a proprietary format. Then the data are filtered based on the abnormal thresholds specified and transformed using the Observations data model which are pushed back to the Kafka bus to be consumed by the Security Policies Manager.

### 6.2.7 Scope of Security Policies Manager in the DFKI pilot.

Since Properties to be monitored and parameters that will be sent to the SSPM are still not finalized, it was not possible to create definitive security rules. A set of initial rules, to be refined in the context of WP6, is given below. At the moment it is clear that the robotino is the main asset to be monitored, however it is still unclear which would be the threats associated to its misbehaviour. This will be furtherly investigated in WP6.

1. **IF** (robotino\_speed > 10 km/h) **THEN** *threat TDB*
2. **IF** (robotino\_battery **IS** drained) **THEN** *threat TDB*

To each rule a Threat, a Vulnerability and an Assets will be associated to create an attack scenario on OLISTIC (as described in 5.2.2 and 5.2.3).

## 6.3 IBER Pilot

### 6.3.1 Environment setup

IBER-OLEFF has implemented an agile production unit with vertical and horizontal integration, namely, to incorporate in a production cell of polymeric components an integrated set of equipment and accessories that should add new functions to the usual ones. One of the challenges had to do with full quality control of parts right after the injection process or before entering the assembly process. Another challenge related to the industrialization of plastic components for automotive interior air vents, including parts with appearance requirements, in ONE-SHOT and ZERO-DEFECT typology, creating a highly customizable product, considering a production unit where several production processes coexist simultaneously, such as injection, decoration, and assembly.

The high-level topology of the manufacturing floor is provided, along with a description of the assets and their dependencies.

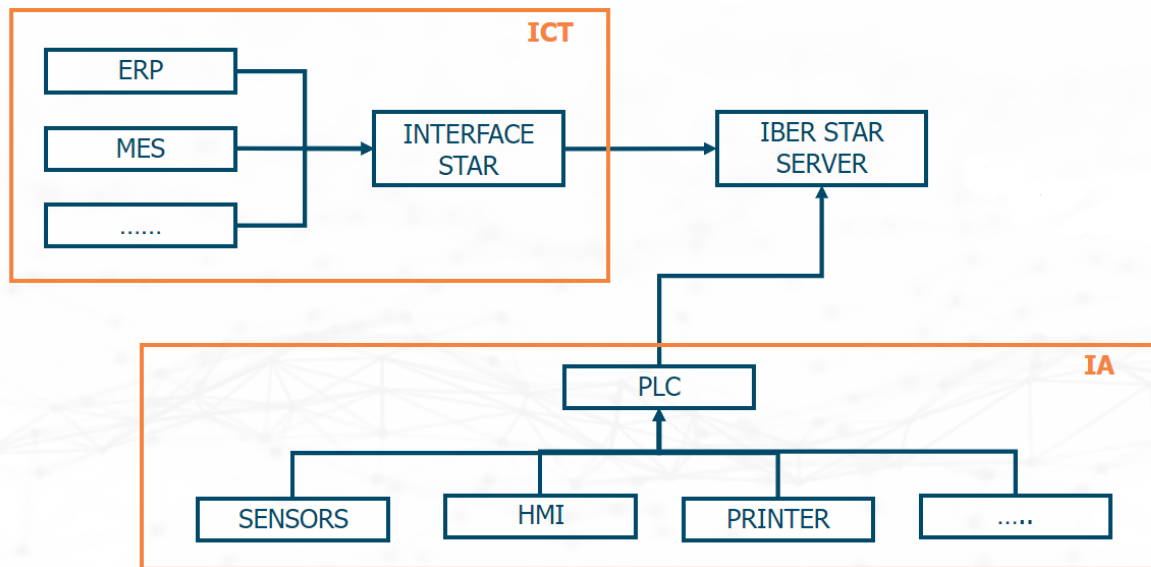


Figure 33 IBER's Environment Setup

Table 10 Assets of IBER's pilot

Asset_ID	Name	Short description	Technical Details	Asset Category
<u>IBER.AS.1</u>	Industrial PLC	A PLC that provides a user interface & runs the required code	WAGO 750-8101 CPU - Cortex A8; 600 MHz	Proprietary Hardware
<u>IBER.AS.2</u>	Industrial PLC OS	Operating system installed on PLC	Real-time Linux 3.18 (with RT-Preempt patch)	Proprietary Software
<u>IBER.AS.3</u>	AI Algorithm	AI algorithm running inside the camera		Proprietary Software
<u>IBER.AS.4</u>	IFM Camera	Camera that makes the quality inspection image when triggered by the PLC	Camera: IFM O2D220 640*480px	Proprietary Hardware
<u>IBER.AS.5</u>	IFM Firmware	Firmware to control the camera	O2D2xx FW1073	Proprietary Software
<u>IBER.AS.6</u>	Sensor /Actuator	Sensors the provides information to the PLC.	Connected with simple on/off signal	Proprietary Hardware
<u>IBER.AS.7</u>	Printer	PLC provides information to printed, to print a new label	Label information	Proprietary Hardware
<u>IBER.AS.8</u>	STAR DB	Storage all information about the	Local storage on server	

		sensors, and production		
<u>IBER.AS. 9</u>	STAR Server	Host of Database	Windows virtualized on ESXI 6.7	

*Table 11 Relationships of IBER's assets*

Asset_ID_X	Relation	Asset_ID_Y	Details
<u>Industrial PLC (1)</u>	Is_connected_to	<u>STAR DB (8)</u>	Sending all information about sensors, cameras, connected via Ethernet
<u>IFM Camera (4)</u>	Is_connected_to	<u>Industrial PLC (1)</u>	Connected via Ethernet
<u>Industrial PLC OS (2)</u>	Is_installed_on	<u>Industrial PLC (1)</u>	
<u>AI Algorithm (3)</u>	Is_installed_on	<u>Camera (4)</u>	
<u>Firmware camera (5)</u>	Is_installed_on	<u>Camera (4)</u>	
<u>Printer (7)</u>	Is_used_by	<u>Industrial PLC (1)</u>	Printing labels
Sensor /Actuator (6)	Is_processed_by	<u>Industrial PLC (1)</u>	I/O
STAR DB (8)	Is_installed_on	STAR Server (9)	

A digital representation of the environment was created using OLISTIC tools (Figure 34). This digital representation materialises the environment illustrated in Figure 33Figure 29Figure 29 OLISTIC digital representation of the Philips environment, considering the interconnections reported in Table 11Table 3 Assets of the cherry inspection station.

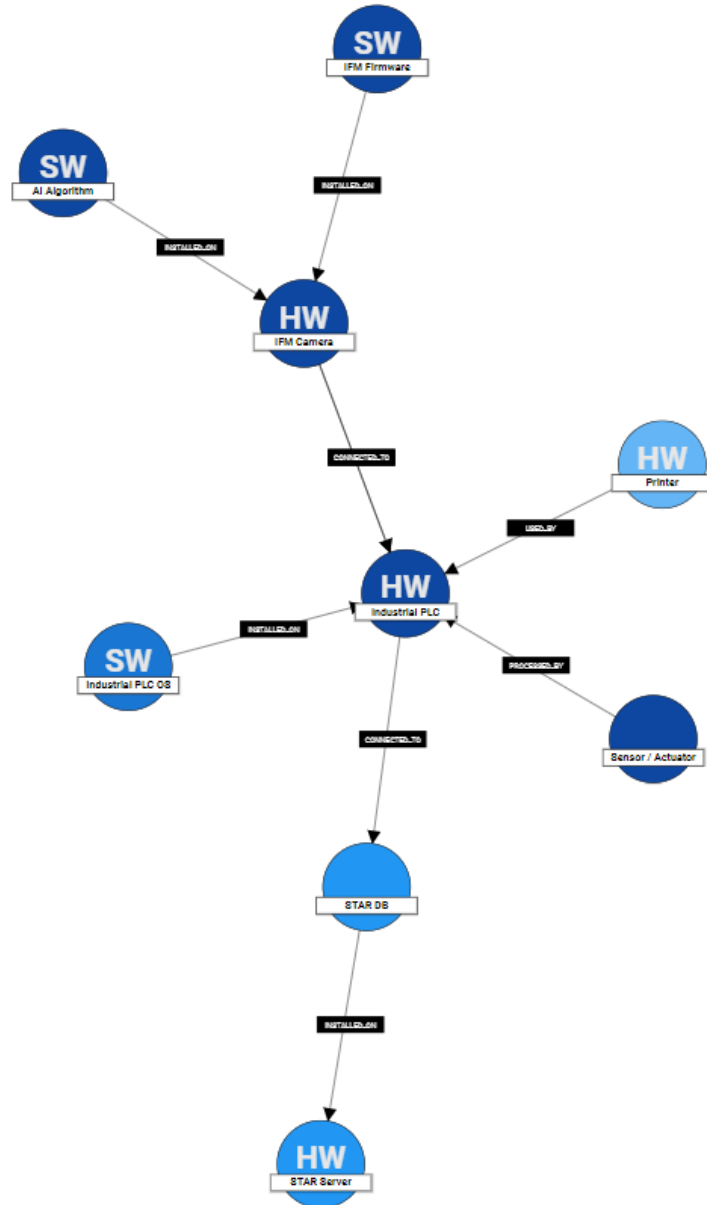


Figure 34 OLISTIC digital representation of IBER's environment

### 6.3.2 Current status and previously witnessed security faults

No security fault was ever witnessed at IBER-OLEFF, however a few potential threats and vulnerabilities were identified that could potentially affect the described environment.

Table 12 IBER identified potential vulnerabilities

Vulnerability ID	Affected Assets	Source	Description
CVE-2021-34569	<u>IBER.AS.1</u>	<a href="https://cert.vde.com/en/advisories/VDE-2020-036/">https://cert.vde.com/en/advisories/VDE-2020-036/</a>	In WAGO I/O-Check Service in multiple products an attacker can send a specially crafted packet containing OS commands to crash the diagnostic tool and write memory. <u>Our Solution:</u> Do not directly connect the device to the internet.

CVE-2019-1068	<u>IBER.AS.8</u>	<a href="https://www.cvedetails.com/cve/CVE-2019-1068/">https://www.cvedetails.com/cve/CVE-2019-1068/</a>	A remote code execution vulnerability exists in Microsoft SQL Server when it incorrectly handles processing of internal functions, aka 'Microsoft SQL Server Remote Code Execution Vulnerability'.
CVE-2020-3992			OpenSLP as used in VMware ESXi (7.0 before ESXi_7.0.1-0.0.16850804, 6.7 before ESXi670-202010401-SG, 6.5 before ESXi650-202010401-SG) has a use-after-free issue. A malicious actor residing in the management network who has access to port 427 on an ESXi machine may be able to trigger a use-after-free in the OpenSLP service resulting in remote code execution.

*Table 13 IBER identified potential threats*

Threats ID	Affected Assets	Source	Description
Denial Of Service attack	<u>IBER.AS.5</u>	Previously simulated attack	The attacker was able to fire a Denial of service attack against the IP of STAR server, setting the production line not saving any historic.
Power Failure	All Assets	Previously faced	The attacker was able to shut down the electrical circuitry, setting the production line, and all systems out of order.
Factory reset on devices		Risk Assessment	The attacker was able to physically factory reset the Cameras, making the imagens not being processed.
Physical connections change	All Assets	Risk Assessment	

### 6.3.3 Environment description

Each zone of assembly production line has a PLC (Programmable Logic Controller). The PLC is responsible for monitoring several sensors (e.g.: Laser Sensors, Capacitive Sensors) and receives the Camera input. The PLC is programmed with a code that not only performs regular checks to ensure that every sensor is properly connected, but also has validations that only count a good piece if there is positive feedback for every sensor.

The code is designed to trigger an alarm if any sensor fails the validation check, indicating that there is a problem with the sensor or its connection. This allows the maintenance team to quickly identify and address any issue, minimizing downtime and reducing the risk of product defects.

27	Actuator 2	0
28	Machine State	2
29	Alam Number	0

*Figure 35 Alarm number for each PLC, example*

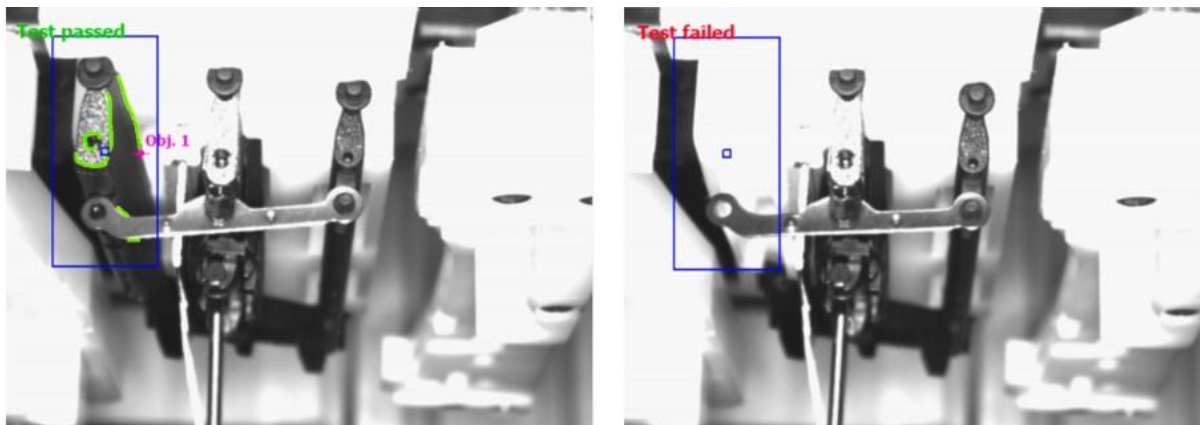
When assembly production is starting, the first part that passes through the line is a known good part, and if the PLC detects any sensor not responding with the correct value, an alarm is triggered, which automatically stops the assembly line.

Because the PLC code is designed to validate every sensor and AI camera, and to trigger an alarm if any sensor fails the check, therefore it would be difficult for an attacker to target a specific sensor without triggering an alarm. A remote attacker can't compromise the communication between sensors and the PLC since they are physically connected, and the sensors aren't connected to ethernet.



*Figure 36 AI camera*

But if an attacker were able to compromise the AI camera validation, after the “known good part”, and changes the parameters used to validate a good product, the camera would validate the part to PLC as a good part, not providing the expected response. And it would not trigger any alarm because the camera would be still responding.



*Figure 37 Camera tests on real components*

A DoS attack on our production system that relies on a PLC would cause disruptions in the system's availability, potentially rendering it inaccessible for a period of time. However, since production is not taking place without the PLC working, the impact of such an attack may be less severe than if production were ongoing because the downtime is controlled.

### 6.3.4 Hypothetical scenarios

Given the analysis the technical partners have identified two hypothetical scenarios that document and focus on different abnormal behaviours in the IBER's production environment.

#### 6.3.4.1 Scenario 1

##### Description

An attacker gains remote access to the IFM camera and changes the parameters used to validate a good product, potentially incorrectly validating the corrected assembly of parts.

##### Abnormal behaviours

- The system detected that the parameters used to validate a good image were changed, and that the camera was accessed from an unknown location.
- The system detected that the assembly of parts was not functioning correctly, indicating that the manufacturing process may have been compromised.
- The camera is not responding, it could mean that it has been compromised and it is saving a new configuration.

#### 6.3.4.2 Scenario 2

##### Description

The PLC responsible for inserting production quality values into the database is not connected to the Ethernet network and is therefore unable to insert the data into the database.

This could be due to a number of factors, such as a disconnected ethernet cable, a malfunctioning network interface, intentional disconnection for maintenance, security reasons or an attack on the industrial network.

Without a connection to the server, the PLC is unable to transmit the production values to the database STAR server. This means that the data is only on the PLC and cannot be accessed or analysed. Thus, data cannot be used for monitoring, reporting, or analysis purposes.

##### Abnormal behaviours:

- If there is no planned maintenance, neither have any issues related to the switches, and one of the assembly lines is not introducing data on the server, that means, something is wrong.
- If the production line is functioning, but no data is saved on the server, there is an issue.
- This could lead to inaccurate or incomplete production data.

### 6.3.5 Critical Assets

Based on the hypothetical scenarios and the environment analysis, IBER has identified a set of critical assets:

- **IMF Camera:** It is a critical resource as it produces the images on which the visual inspection process takes place and hosts the algorithms that perform the quality inspection. A potential malfunction on the camera can stop production or cause financial loss since NOK parts are being evaluated as OK parts.

- **Industrial PLC (PLC):** It is a critical resource where all sensors and cameras are connected and processed. If PLC is not working, there is no production system.
- **PLC:** it is a critical resource where all sensors and cameras are connected and processed. If PLC is working, but can't store the information in database, because network is not working, no values will be saved.
- **Database:** it is a critical resource where all data is stored.

### 6.3.6 Properties to be monitored

Based on the analysis performed, the WP3 technical partners and IBER has identified a set of key indicators that can be monitored in the production environment and could reveal potential malfunctions or attacks against the above-mentioned critical assets.

- **Production rate:** X items per hour
- **Device availability:** e.g., ping an exposed endpoint to identify availability;
- **Device resources:** e.g., Average CPU/GPU/RAM utilization;

Asset	How to access data	Values/Properties to monitor	Normal values and abnormal thresholds
Device availability	Ping an exposed Robot API and Push data to RMS provided API.	Availability of network interface.	Device connected to network should response to a ping.
Production rate	Polling an exposed API	OK and NOK production rate	The NOK production rate should not exceed X pieces per hour.
Average CPU utilization	Push data to RMS provided API.	CPU/GPU utilization percentage	The CPU/GPU utilization should not exceed 80% per hour

### 6.3.7 Scope of AI Cyber Defence tool in the IBER pilot.

As was the case also for the Philips pilot, the focal point in the context of the IBER pilot is the visual quality inspection setup. Considering the threat landscape, the AI-based visual inspection system, the WP3 partners will focus on the detection of poison and evasion attacks that may threaten the model that empowers the aforementioned system. The concepts of evasion and poisoning attacks was introduced in section 6.1.6 in the context of the Philip’s pilot. The same approach will be used also for the IBER pilot.

Thus, in this context UBITECH team will be focused on analysing the image datasets of IBER for creating malicious instance and evaluating defences. The image below shows how the soother images are utilised in order to create adversarial instances and evade the AI-based quality inspection model.

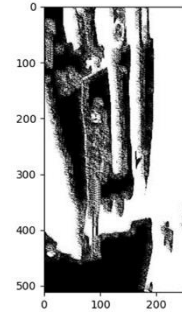
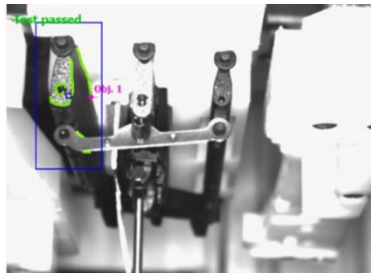


Figure 38 Generation of adversarial examples (deepfool attack) for the IBER dataset

### 6.3.8 Scope of Runtime Monitoring System tool in the IBER pilot.

The scope of RMS in the IBER pilot is to collect the identified monitored properties, using the methods specified in the table below, which are received in a proprietary format. Then the data are filtered based on the abnormal thresholds specified and transformed using the Observations data model which are pushed to the Kafka bus to be consumed by the Security Policies Manager.

Asset	Beat	Values/Properties to monitor	Normal values and abnormal thresholds
All assets with network interface	<a href="#">Heartbeat</a>	On / Off	ON – If equipment is turned on  OFF – Means a network problem, or any not allowed change in system.
STAR Server	<a href="#">Metricbeat</a>	CPU / Memory Usage	
PLC / STAR	<a href="#">Packetbeat</a>	All packets	

### 6.3.9 Scope of Security Policies Manager in the IBER pilot.

From the information collected from the Pilot, few initial policies have been created for both scenarios. In the context of WP6, additional security rules will be defined, and the policies will be further refined and enriched, in particular the threats and vulnerabilities associated to each rule to create attack scenarios in OLISTIC in case a policy is violated.

1. **IF** (monitored device is OFF) **THEN** potential DoS attack ongoing
2. **IF** (STAR server CPU usage > 50%) **THEN** potential DoS attack ongoing
3. **IF** (NOK production rate > X pieces/hour) **THEN** (*threat to be defined*)
4. **IF** device is not online **THEN** potential DoS attack ongoing

In Table 14 each rule is associated with a Threat, a Vulnerability and an Assets, these three elements are used to create an attack scenario on OLISTIC when a policy is violated.

Table 14 Attack Scenarios for Policies for IBER's Scenario 1

Rule	Threat	Vulnerability	Asset
1	DoS Attack	TBD based on specific asset monitored	TDB
2	DoS Attack	CVE-2019-1068	IBER.AS.5

3	TBD	TBD based on identified threat	Production rate
4	DoS Attack	TBD based on specific asset monitored	TBD

## 6.4 Discussion on survey results

This section has described the final layout of the tools developed in WP3, based on the insights gathered from the surveys sent to the Pilots. The collaboration between technical partners from WP3 and use case partners from WP6 proved essential in assessing the applicability of the WP3 tools within the Security & Data Governance solution of STAR. The questionnaire circulated among the pilot partners provided valuable information about their manufacturing environments, critical assets, vulnerabilities, and potential adversarial scenarios.

Despite most pilot partners reporting no previous attacks due to their isolated environments, the formulation of hypothetical – but plausible- scenarios allowed for a comprehensive understanding of potential abnormal behaviours in the production lines. This information plays a crucial role in defining relevant security policies within the SPM tool. To be noted, the described scenarios are hypothetical, they are based on previously faced attacks and potential threats that could disrupt the normal pilot’s environment behaviour but have never happened directly to the Pilots.

By monitoring key indicators such as image generation rate, device resources, and production rate, the pilots are well-equipped to detect and prevent potential malfunctions or attacks. The findings from this deliverable have paved the way for defining the final set of security policies, which will be further refined during WP6 actions.

The outcomes of the collaborative activity with the Pilots serve as a steppingstone for further advancements in WP6. While we have made significant progress in defining rules and priorities during this stage, it's important to note that WP3- work with the partners is an ongoing process. The final version of the tools and policies remains the ultimate objective of WP6. The dedication to continuous improvement ensures that the final versions of the tools will be comprehensive, efficient, and aligned with the needs of the Pilots.

## 7 Conclusions

This Deliverable marks the final milestone in the development of the Security and Data Governance Infrastructure within the STAR project. Building on the initial version showcased in the 3.5 deliverable, the current deliverable focuses on the adaptation of general findings to real industrial environments through the integration of information from the Use Cases. The collaboration between WP3 and WP6 partners has been instrumental in achieving this goal, with a primary focus on delivering practical and useful tools to enhance the management of technology in the use cases.

The AI Security and Data protection layer ensures the operational assurance and credibility of a manufacturing floor. It aims to empower Factory Security Officers by providing them with the tools to effectively govern and regulate the operational behaviour within the manufacturing environment. In the context of STAR, this objective is accomplished by seamlessly integrating individual components into a unified flow, delivering crucial indications and alerts to the security officer. These notifications accurately reflect the security and operational status of the monitored deployment, enabling proactive management and response to potential threats and incidents.

The acquired information from the Pilot partners has played a crucial role in defining relevant security policies within the context of the SPM tool. This integration enables the SPM to combine data from various indicators, effectively identifying potential abnormal behaviours in the production lines. Although no previous attacks were reported by the pilot partners due to their detached environments, the formulation of hypothetical scenarios tailored to the nature of the monitored production lines has ensured that the system can handle real-world challenges.

This deliverable provides a comprehensive account of the detailed designs, implementation, integration, and evaluation of all modules, emphasizing their successful deployment across all use cases and pilots. The document highlights the seamless interaction between various components of the SPM, detailing how they exchange information and work in synergy to create a robust and effective Security and Data Governance Infrastructure.

The efforts made by task 3.5 partners in the development of this infrastructure have led to a refined and tailored solution, ready for deployment in real-world industrial environments. The collaboration and synergy between WP3 and WP6 have been instrumental in ensuring the applicability and effectiveness of the WP3 tools, culminating in the delivery of a solution that provides better technology management for the use cases. The STAR project has now a comprehensive and reliable Security and Data Governance Infrastructure, built with practicality, efficiency, and security at its core to meet end users' (pilot's) necessities.

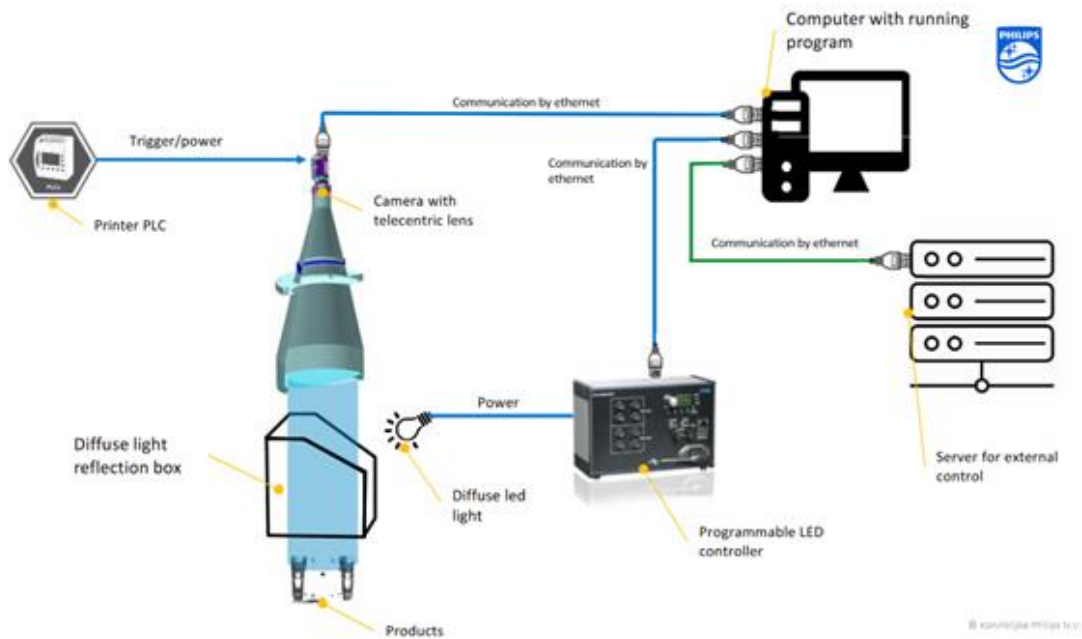
## References

Page	Link to document
16	<a href="https://www.openpolicyagent.org/docs/latest/">https://www.openpolicyagent.org/docs/latest/</a>
20	<a href="https://www.elastic.co/elastic-stack/">https://www.elastic.co/elastic-stack/</a>
21	<a href="https://www.elastic.co/guide/en/beats/libbeat/current/index.html">https://www.elastic.co/guide/en/beats/libbeat/current/index.html</a>
21	<a href="https://github.com/elastic/beats">https://github.com/elastic/beats</a>
21	<a href="https://www.elastic.co/guide/en/beats/libbeat/master/community-beats.html">https://www.elastic.co/guide/en/beats/libbeat/master/community-beats.html</a>
21	<a href="https://github.com/elastic/beats/tree/master/filebeat">https://github.com/elastic/beats/tree/master/filebeat</a>
21	<a href="https://github.com/elastic/beats/tree/master/metricbeat">https://github.com/elastic/beats/tree/master/metricbeat</a>
21	<a href="https://github.com/elastic/beats/tree/master/packetbeat">https://github.com/elastic/beats/tree/master/packetbeat</a>
22	<a href="https://www.elastic.co/kibana/">https://www.elastic.co/kibana/</a>
22	<a href="https://www.elastic.co/elasticsearch/">https://www.elastic.co/elasticsearch/</a>
27	<a href="https://www.first.org/cvss/">https://www.first.org/cvss/</a>
29	<a href="https://nvd.nist.gov/">https://nvd.nist.gov/</a>
29	<a href="https://nvd.nist.gov/General/News/retire-cvss-v2">https://nvd.nist.gov/General/News/retire-cvss-v2</a>
35	<a href="https://www.openpolicyagent.org/docs/latest/policy-language/">https://www.openpolicyagent.org/docs/latest/policy-language/</a>

# Appendix A - WP3 tools in pilot template

## 1. Environment setup

Topology of the manufacturing floor or the production line



### Assets

[List here all the software and Hardware assets participating in the production line]

Asset_ID	Name	Short description	Technical Details	Asset Category	Equipment
<i>PCL.AS.1</i>	Industrial PC	A computer that provides a user interface & runs the required software	Version CPE Type Model No.	Proprietary Hardware	10 OR 11?

## 2. Asset dependencies

The risk assessment engine is able to represent the interdependencies that may exist among the identified assets. This subsection aims to **document the interdependencies of the assets**. The result will be an interdependency graph as show in the figure below.

**Visit the RAME tool of STAR here and add assets** so that to create the digital reflection of your environment and your manufacturing floor here: <https://star-rame.euprojects.net/asset>

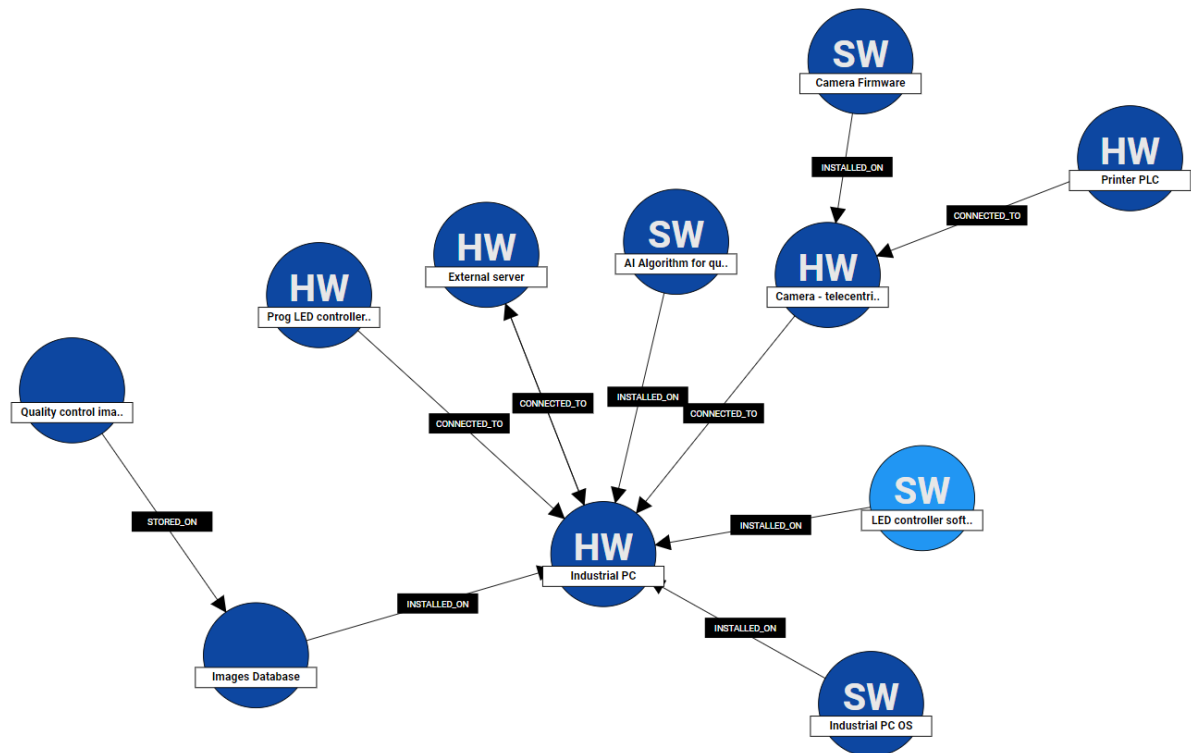
Consider the following relations among the assets:

- Is\_installed\_on
- Is\_connected\_to
- Is\_used\_by
- Is\_located\_in
- Is\_stored\_on
- Is\_Processed\_by

Fill in the following table with the assets and relations that you have created in the RAME environment.

Asset_ID_X	Relation	Asset_ID_Y	Details
<i>Printer PLC (8)</i>	Is_connected_to	<i>Camera (4)</i>	On/off signal / power

Copy and paste here a screen shot of the environment that you generated in the RAME.



### 3. Known vulnerabilities of assets

In the following table please fill in known vulnerabilities that may affect the assets of your manufacturing floor. Fill the table considering:

- **Your domain and infrastructure knowledge:** Based on your experience and previous digital security issues that you may have faced in your organisation.
- **Online repositories:** Such repositories offer search tools so that to find vulnerabilities that may affect your software and hardware assets in your

manufacturing floor. (e.g., [www.cvedetails.com](http://www.cvedetails.com), <https://nvd.nist.gov/vuln/search>)

Vulnerability ID	Affected Assets	Source	Description
e.g. CVE-2017-15361	<i>PCLAS.5</i>	<a href="http://www.cvedetails.com">www.cvedetails.com</a>	The Infineon RSA library 1.02.013 in Infineon Trusted Platform Module (TPM) firmware, such as versions before 0000000000000422 - 4.34, before 000000000000062b - 6.43, and before 00000000000008521 - 133.33, mishandles RSA key generation.
e.g. CVE-2017-18361	<i>PCLAS.5</i>	<a href="https://nvd.nist.gov/vuln/search">https://nvd.nist.gov/vuln/search</a>	In Pylons Colander through 1.6, the URL validator allows an attacker to potentially cause an infinite loop thereby causing a denial of service via an unclosed parenthesis.

### Other threats to be reported.

If there are vulnerabilities/threats that you could not find in the catalogues above, please fill in this table with a general description of such vulnerability.

Vulnerability/Threats ID	Affected Assets	Source	Description
e.g Denial Of Service attack	<i>PCLAS.5</i>	Previously faced attack	The attacker was able to fire a Denial of service attack against the IP camera of the visual inspection system, setting the production line out of order.

## 4. Scenarios

Current status and previously witnessed security and production faults

Have you ever faced cyber security incidents in your manufacturing floor that affected your production?

### 1. Case A

Description:  
 Critical Assets/sensors/processes affected:  
 Vulnerabilities or threat detected:  
 Abnormal behaviors:

### 2. Case B

Description:  
 Critical Assets affected:  
 Vulnerabilities or threat detected:  
 Abnormal behaviors:

### 3. Case X

Description:

Critical Assets affected:

Vulnerabilities or threat detected:

Abnormal behaviors:

## 5. Definition use case scenario

Considering the previously reported incidents, we **need to come up with a storyline of a demonstration scenario to contextualise UCx**. The aim is to **build a scenario as close as possible to the pilots' needs** but showing off the core functionalities of all WP3 tools.

### Scenario Example

**Description:** [*Add below a description of a cyber security scenario that can be plausible in your environment and against the specific production line of the use case. Create this scenario, based on your experience for previously witnessed incidents in your organization or based on possible scenarios that you may think of.*]

*e.g. AI-assisted visual quality inspection is undertaken by a fixed system on PCL's manufacturing floor. An adversary has gained remote access to the infrastructure and is able to interact with critical systems and change the operational behaviour of the production line. In this regard, the following attack scenarios can be executed by the attacker. The adversary tries to evade the inference process of the AI-assisted visual quality inspection system by injecting adversarial images among the normal images sent to the visual inspection system. The adversary tries to affect the availability of critical systems/sensors performing DoS attack trying to drain the system's available resources. The goal is to disrupt the normal operation of the production line.*

**Critical Assets:** [*List here the critical assets that need to be protected by cyber-attacks. Consider that if those assets get compromised, the production process may be affected significantly or safety incidents may occur.*]

- **Camera:** It is a critical resource as it produces the images based on which the visual inspection process takes place. A potential malfunction on the camera can cause financial loss
- **Industrial PC:** It is a critical resource as it hosts the algorithms that perform the quality inspection.

**Vulnerable assets:** [*List the assets that be the target of an attacker and may be used in the context of an attack.*]

- **e.g. Database: MySQL:** A SQL injection vulnerability exists when MySQL is being used as the application database. An attacker can issue SQL commands to the MySQL database through the vulnerable "id" parameter.
- **e.g. Camera:** The camera has the port 22 open but no authentication is required to login in the camera user interface.

**Abnormal behaviours:** [*Describe here the rationale behind the identification of an abnormal situation in your systems that may indicate the presence of an attacker*]

- The production rate is known, and the rate of images sent to the visual inspection system for quality checking is fixed. Any increment in the production rate indicating the injection of image samples in the process.

**Properties to be monitored:** *[List here indicators that you use to monitor the behavior of your production line and systems and their nominal values (normal operation thresholds).]*

- Production rate: e.g. X items per hour
- Device resources: e.g. Average CPU utilization per hour 80% and 70% of RAM
- Device status: e.g. Motor RPMs (between 2000-2500), Motor Temperature (between 70-90 °C), Robot speed (between 0-10KMPH),
- [Add here your indicator...]
- [Add here your indicator...]
- [Add here your indicator...]
- 

Asset	How to access data	Values/Properties to monitor	Normal values and abnormal thresholds
e.g. Camera	Log, API, database,	Images generation rate	The images generation rate should be fixed: 50 images per minute. Any discrepancy is an indication of abnormal behaviour.
e.g. Robot Speed	Polling an exposed API (HTTPS)	Robot Speed	The robot speed should not exceed 10KMPH (0-10)

**Additional properties and conditions:** *[Add here in the form of free text additional information that you think that may be relevant to the scenario, the conditions under which a cyber incident may occur or how your systems may be affected.]*

## Scenario

### Description:

- [Add here the description of the scenario]

### Critical Assets:

- [Add here an asset...]
- [Add here an asset...]

### Abnormal behaviors:

- [Add here description of an abnormal behavior...]

### Properties to be monitored:

- [Add here your indicators/properties...]
- [Add here your indicators/properties...]

Asset	How to access data	Values/Properties to monitor	Normal values and abnormal thresholds
	e.g.: Log, API, ..		


**Additional properties and conditions:**

- [Add here additional information...]

## 6. WP3 tool-specifics

### AI Cyber defence

- Define the Datasets to be used.
- Model used for training – Is there any AI model that you already use to empower your system?

### RMS

RMS can access data either by:

- Deploying small pieces of software called beats (see table below) that will push data to the security infrastructure
- By polling for data thru an exposed API that will be provided by the pilot
- By accepting data from the pilot (pilot system to push data to the security framework)
  - E.g. to:
    - Push data to an exposed API from the security infrastructure
    - Push data to Kafka data bus provided by the security infrastructure

Beat	Description
<a href="#">Auditbeat</a>	Collect your Linux audit framework data and monitor the integrity of your files.
<a href="#">Filebeat</a>	Tails and ships log files
<a href="#">Functionbeat</a>	Read and ships events from serverless infrastructure.
<a href="#">Heartbeat</a>	Ping remote services for availability
<a href="#">Metricbeat</a>	Fetches sets of metrics from the operating system and services
<a href="#">Packetbeat</a>	Monitors the network and applications by sniffing packets
<a href="#">Winlogbeat</a>	Fetches and ships Windows Event logs
<a href="#">Osquerybeat</a>	Runs Osquery and manages interaction with it.

Asset	Beat	Values/Properties to monitor	Normal values and abnormal thresholds

## 7. Star Blockchain

Define AI algorithm configurations (in the form of Processor Manifest (PM)) and critical results (in the form of Observations) to be stored and checked

## 8. Security policies manager

Policies that regulate the behaviour of the manufacturing floor.

**Example rules based on Scenario example described in 2.2.1:**

- **E.g.: IF** (CPU load >90%) **THEN** potential DoS attack ongoing

Following the given examples, create a set of IF-THEN statements and RULES.

- ....
- ...