

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

D3.5 – Security and Data Governance Infrastructure-Initial version

Dissemination level	PU -Public
Type of Document	Report
Contractual date of delivery	31/05/2022
Deliverable Leader	GFT
Status - version, date	Final – v1.0, 15/06/2022
WP / Task responsible	WP3
Keywords:	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 first 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. 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 (offered 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.

The document begins with an explanation of the Security and Data Governance Infrastructure with all the modules developed by WP3 partners. This is a detailed summary of the technologies involved during the development of this task.

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 Policy Manager. A PoC of the Validation scenario is proposed, with a reference to input and outputs, also reported as scripts in Appendix A.

The answers to a survey aimed at collecting use cases' necessities in the field of security policies are also summarised and the filled-in questionnaires are reported integrally as Appendixes (Appendix B).

Deliverable Leader:	GFT
Contributors:	GFT, INTRA, UBI
Reviewers:	SUPSI, DFKI
Approved by:	INTRA

Document History			
Version	Date	Contributor(s)	Description
0.1	28/03/2022	GFT	Initial version and ToC
0.2	16/05/2022	UBI, INTRA	Contributions
0.3-0.5	31/05/2022	SUPSI, DFKI	Reviewed Final Version
0.6	01/06/2022	GFT, INTRA, UBI	Integration
1.0	15/06/2022	INTRA	QA and creation of the final submitted version

Table of Contents

EXECUTIVE SUMMARY	2
TABLE OF CONTENTS.....	4
TABLE OF FIGURES.....	6
LIST OF TABLES.....	7
DEFINITIONS, ACRONYMS AND ABBREVIATIONS.....	8
1 INTRODUCTION	9
1.1 OVERVIEW AND PURPOSE	9
1.2 RELATIONSHIP TO OTHER DELIVERABLES.....	9
1.3 DELIVERABLE STRUCTURE.....	9
2 THE SECURITY AND DATA GOVERNANCE INFRASTRUCTURE ARCHITECTURE.....	11
2.1 THE COMPONENTS.....	14
2.1.1 <i>Distributed Ledger Services for Data Reliability</i>	14
2.1.2 <i>Runtime Monitoring System</i>	14
2.1.3 <i>AI Cyber Defence Module</i>	15
2.1.4 <i>OLISTIC</i>	15
2.1.5 <i>Security Policy Manager</i>	15
3 RUNTIME MONITORING SYSTEM.....	22
3.1 ARCHITECTURE	22
3.2 COMPONENT DIAGRAM AND API IDENTIFICATION	24
3.2.1 <i>Interface Specification</i>	24
3.3 PROBE AVAILABILITY AND USAGE.....	25
3.4 GUI.....	27
4 OLISTIC	29
4.1 ARCHITECTURE	29
4.2 STAR RISK ASSESSMENT METHODS AND MODELS	31
4.2.1 <i>Asset Modelling & Visualization</i>	31
4.2.2 <i>Conceptual OLISTIC Risk Assessment meta-model</i>	34
4.2.3 <i>Risk, Vulnerability and Threat modelling</i>	35
4.3 INPUT.....	40
4.4 OUTPUT.....	40
4.5 GUI.....	40
5 STAR SECURITY POLICY MANAGER.....	42
5.1 ARCHITECTURE	42
5.2 INPUT.....	43
5.3 OUTPUT.....	45
5.4 SERVICE DISTRIBUTION AND CONFIGURATION.....	45
6 RELEVANT SECURITY POLICIES ASSESSMENT FOR USE CASES	46
6.1 USE CASES ANALYSIS.....	46
6.1.1 <i>1.Human Behaviour Prediction and Safe Zone Detection for Routing (DFKI)</i>	46
6.1.2 <i>2.Human Centred AI for Agile Manufacturing 4.0 (IBER)</i>	47
6.1.3 <i>3.Pilot Human-Robot Collaboration for Quality Management (PHILIPS)</i>	47
6.2 USE CASES SURVEYS RESULTS.....	47
6.2.1 <i>Human Behaviour Prediction and Safe Zone Detection for Routing</i>	48

- 6.2.2 *Human Centred AI for Agile Manufacturing 4.0*.....49
- 6.2.3 *Pilot Human-Robot Collaboration for Quality Management*.....49
- 6.3 FINE TUNING OF SECURITY POLICIES.....50
 - 6.3.1 *Pilots’ assets*.....50
 - 6.3.2 *Security policies*55
- 6.4 DISCUSSION ON SURVEYS RESULTS56
- 7 POC INTEGRATED ARCHITECTURE & VALIDATION SCENARIO..... 58**
 - 7.1 INTEGRATED ARCHITECTURE.....58
 - 7.2 COMMON DATA MODELS59
 - 7.2.1 *Observations*.....59
 - 7.2.2 *Security-focused Configuration Management (SecCM) Data Model*.....60
 - 7.3 INTEGRATION & VALIDATION.....63
 - 7.3.1 *Validation Scenario and Components Integration*.....63
- 8 CONCLUSIONS 65**
- REFERENCES..... 66**
- APPENDIX A COMPONENT SCRIPTS..... 67**
 - A.1 COMPONENT INPUTS & OUTPUTS.....67
- APPENDIX B QUESTIONNAIRES..... 69**
 - B.1 SECURITY POLICIES NEEDS ASSESSMENT - HUMAN BEHAVIOR PREDICTION AND SAFE ZONE DETECTION FOR ROUTING69
 - B.1.1. Do you have already figured out a list of security policies and related rules?*.....71
 - B.2 SECURITY POLICIES NEEDS ASSESSMENT - HUMAN CENTRED AI FOR AGILE MANUFACTURING 4.0.....72
 - B.3 SECURITY POLICIES NEEDS ASSESSMENT - PILOT HUMAN-ROBOT COLLABORATION FOR QUALITY MANAGEMENT75

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 RBAC BASED EXAMPLE OF POLICY17

FIGURE 5 GENERIC USE CASE SECURITY POLICY, BASED ON IMAGE SIZE VISUALIZATION PERMISSION.....19

FIGURE 6 GENERIC USE CASE SECURITY POLICY, ACTION ALLOWED.....20

FIGURE 7 GENERIC USE CASE SECURITY POLICY, ACTION DENIED.....21

FIGURE 8 RMS DATA FLOW DIAGRAM.....22

FIGURE 9 RMS COMPONENT DIAGRAM.....24

FIGURE 10 PROBE DATA STORAGE SEQUENCE DIAGRAM25

FIGURE 11 RMS INFRASTRUCTURE FLOW.....26

FIGURE 12 RMS DATA COLLECTION, TRANSFORMATION & FILTERING EXAMPLE27

FIGURE 13 KIBANA DISCOVERY VIEW DASHBOARD.....27

FIGURE 14 KIBANA DASHBOARD VIEW.....28

FIGURE 15 OLISTIC INTERNAL COMPONENT ARCHITECTURE.....29

FIGURE 16 ASSETS OLISTIC GUI AND ADDITIONAL OPERATION OPTIONS32

FIGURE 17 EXAMPLE OF INTERDEPENDENCY GRAPHS IN OLISTIC.....32

FIGURE 18 SOFTWARE ASSET TEMPLATE.....33

FIGURE 19 HARDWARE ASSET TEMPLATE34

FIGURE 20 DATA ASSETS TEMPLATE.....34

FIGURE 21 OLISTIC RISK ASSESSMENT META-MODEL35

FIGURE 22 VULNERABILITY MODELLING TEMPLATE37

FIGURE 23 THREAT TEMPLATE38

FIGURE 24 RISK APPETITE TEMPLATE.....38

FIGURE 25 ATTACK SCENARIO TEMPLATE.....40

FIGURE 26 ATTACK SCENARIOS OVERVIEW40

FIGURE 27 OLISTIC DASHBOARD41

FIGURE 28 OLISTIC RISK ASSESSMENT MANAGEMENT ENVIRONMENT.....41

FIGURE 29: SSPM HIGH LEVEL ARCHITECTURE.....42

FIGURE 30 DIFFERENT MODELS FOR LOADING BASE DOCUMENTS INTO OPA,44

FIGURE 31 POLICIES MODEL DOCUMENT (EXAMPLE)44

FIGURE 32 OLISTIC ASSET CARTOGRAPHY FOR HUMAN BEHAVIOR PREDICTION AND SAFE ZONE DETECTION FOR ROUTING.....51

FIGURE 33 OLISTIC CARTOGRAPHY HUMAN CENTRED AI FOR AGILE MANUFACTURING 4.0.....53

FIGURE 34 OLISTIC CARTOGRAPHY HUMAN-ROBOT COLLABORATION FOR QUALITY MANAGEMENT.....54

FIGURE 35 EXAMPLE OF POLICY REPRESENTATION57

FIGURE 36 STAR SECURITY AND DATA GOVERNANCE INTEGRATED ARCHITECTURE58

FIGURE 37 OBSERVATION ENTITY STRUCTURE.....60

FIGURE 38 SecCM ENTITY STRUCTURE.....61

FIGURE 39 ASSET ENTITY STRUCTURE62

FIGURE 40 ATTACK SCENARIOS ENTITY STRUCTURE63

FIGURE 41 RMS-MONITORED SYSTEM OBSERVATION DATA (PROBE MONITORING DATA).....67

FIGURE 42 AI CYBER DEFENCE OBSERVATION.....67

FIGURE 43 SECURITY POLICIES MANAGER CONFIGURATIONS.....68

FIGURE 44 SECURITY POLICIES MANAGER RESULTS.....68

List of Tables

TABLE 1 RMS INTERFACE SPECIFICATION.....	24
TABLE 2 CONTINUOUS INTERVAL OF OLISTIC'S RISK CLASSIFICATION	36
TABLE 3 SUMMARIZED ANSWERS FROM THE PILOT HUMAN BEHAVIOUR PREDICTION AND SAFE ZONE DETECTION FOR ROUTING.....	48
TABLE 4 SUMMARIZED ANSWERS FROM THE PILOT HUMAN CENTERED AI FOR AGILE MANUFACTURING 4.0.....	49
TABLE 5 SUMMARIZED ANSWERS FROM PILOT HUMAN-ROBOT COLLABORATION FOR QUALITY MANAGEMENT	49
TABLE 6 HUMAN BEHAVIOR PREDICTION AND SAFE ZONE DETECTION FOR ROUTING PILOT'S ASSETS.....	50
TABLE 7 HUMAN CENTRED AI FOR AGILE MANUFACTURING 4.0 PILOT'S ASSETS	52
TABLE 8 HUMAN-ROBOT COLLABORATION FOR QUALITY MANAGEMENT PILOT'S ASSETS.....	53
TABLE 9 PERSPECTIVES ON TRANSFORMING CYBERSECURITY, MCKINSEY AND COMPANY, 2019.....	55
TABLE 10 COMPONENT INTEGRATION DEPENDENCIES AND FLOWS	64
TABLE 11 LIST OF ASSETS.....	69
TABLE 12 LIST OF ASSETS INTERDEPENDENCIES	70
TABLE 13 LIST OF ASSETS.....	72
TABLE 14 LIST OF ASSETS INTERDEPENDENCIES	73

Definitions, Acronyms and Abbreviations

Acronym/ Abbreviation	Title
ABAC	Attribute-based Access Control
API	Application Programming Interface
BFT	Byzantine Fault Tolerance (or Tolerant)
BTC	Bitcoin
CA	Certificate Authority
CE	Circular Economy
CFT	Crash Fault Tolerance (or Tolerant)
CLI	Command-Line Interface
CRUD	Create Read Update Delete
DLSDR	STAR Distributed Ledger Services for Data Reliability
DLT	Distributed Ledger Technology
DoA	Description of Action
DPT	Data Provenance and Traceability
EDM	Ecosystem Data Manager
GUI	Graphical User Interface
HLF	HyperLedger Fabric
JSON	JavaScript Object Notation
MSP	Membership Service Provider
MVP	Minimum Viable Product
P2P	Peer-to-Peer
PKI	Public Key Infrastructure
PoS	Proof of Stake
PoW	Proof of Work
RBAC	Role Based Access Control
SC	Smart Contract
TLS	Transport Layer Security
UBAC	User Based Access Control
URI	Universal Resource Identifier
URL	Universal Resource Locator
UUID	Universally Unique Identifier
WP	Work Package
XACML	eXtensible Access Control Markup Language
XSD	XML Schema Definition

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 the deliverable D3.5 “Security and Data Governance Infrastructure-Initial version” is to describe the advancements made in the development of the Security and Data Governance Infrastructure with inputs from GFT, UBI and INTRA, namely the focus of the report would be:

- The description of the modules of the Data Governance Infrastructure: the Runtime Monitoring System, the Explainable AI (XAI), the risk assessment module Olistic, the Star Security Policy Manager (SSPM);
- The initial 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 initial 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, the components of the Security Policy Manager and high-level interactions with other 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. The outcome of these actions is the documentation of information flows, which are giving also as input in the developments of WP2 for the definition of the overall STAR project architecture.

1.2 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.
- D6.3 provides input for the integrated STAR platform.

1.3 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 produced by the task partners;
- **Section 3** deepens in the description of the Runtime Monitoring System 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** and **Section 6** describe the Security Policy Manager and how the security policies can be implemented in the future;
- **Section 7** describes data models implemented in the Security and Data Governance Infrastructure.

2 The Security and Data Governance Infrastructure architecture

As detailed in D 2.6 “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).

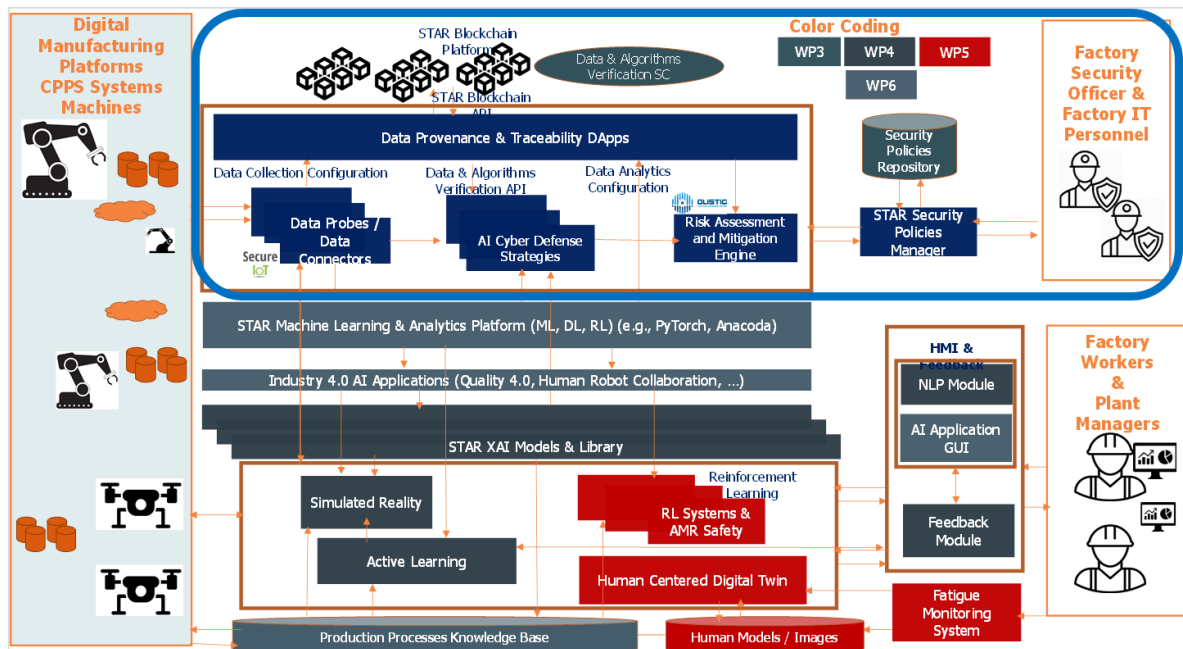


Figure 1 STAR Functional Modules and Logical View of the Architecture [D2.6]

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 (offered 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.

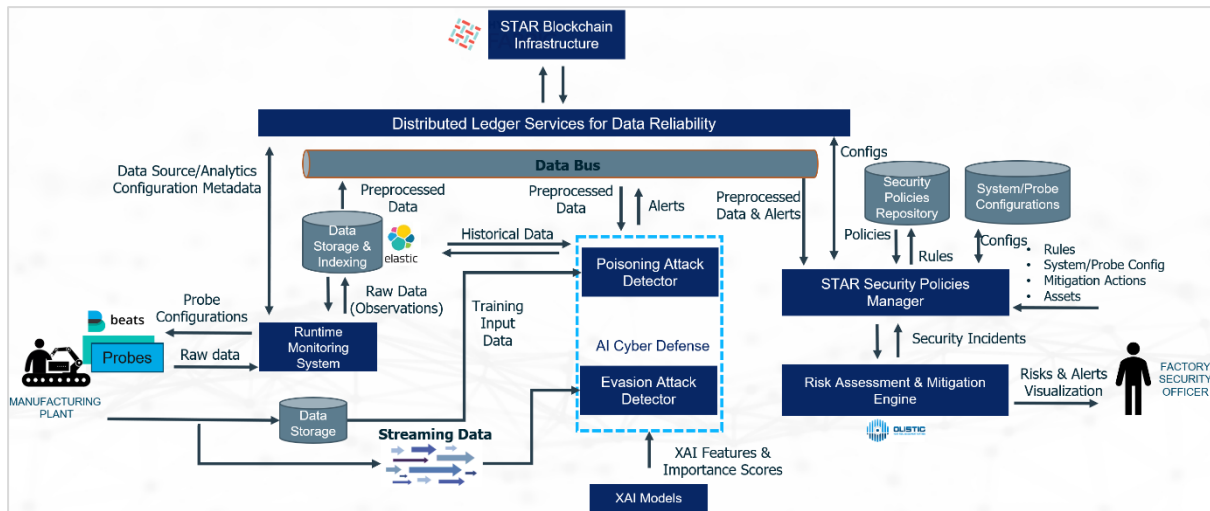


Figure 2 STAR Security and Data Governance for AI Systems in Manufacturing Logical View

The purpose of the AI Security and Data protection layer is to guarantee the operational assurance and credibility of a manufacturing floor. That is, 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 Systems 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 through the deployment of probes which are 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 introduced in the context of D3.1 and D3.3, respectively. Thus, this deliverable describes only the placement and interactions of those components in the frame of the AI Security and Data protection layer. The interested reader can refer to D3.1 and D3.3 for more technical details. 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 this deliverable.

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 D3.1 section 3.3.1). Information about the exposed API, data models and usage can be found in section 4.2.3 of D3.1, and
- For persisting AI algorithm results by utilizing the Analytics Results Publishing (ARP) service (see D3.1 section 3.3.2) using the Observation data structure as described in D3.1 section 4.2.4. Information about the exposed API can be found in section 4.2.4 of D3.1. Samples of the blockchain persisted Analytics' results can be consumed by the Security Policy 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

RMS, depicted in Figure 2 above, 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 Policy 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). 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. 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.

2.1.4 OLISTIC

OLISTIC is the technical component that will complement the Security Policy 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 Policy 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. Overall, OLISTIC will enable the risk management and the identification and visualization of risks through comprehensive and reactive visualization, while it will provide the means to the security officer to manage the life cycle of mitigation actions that will help to eliminate or control risk events that have been detected by the monitoring mechanisms of STAR. More details on the utilized risk assessment method and the tool's architecture are given in Section 4.

2.1.5 Security Policy Manager

The STAR Security Policy 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 Runtime Monitoring System 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)

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

Research activities were carried out about existing standards, methodologies and technologies to manage security policies within AI/Bigdata frameworks. The open-source platform OPA was selected for its flexibility and accessibility compared to other tools.

Open Source

- <https://open-scap.org/>

OPEN-SCAP tool, despite being open-source, is less adaptable to users' needs, it also embodies a security system and follows specific standards.

The language adopted by OPEN-SCAP is Security Content Automation Protocol (SCAP) which is a U.S. standard maintained by the National Institute of Standards and Technology (NIST).

Non-Open Source:

- <https://www.imanage.com/products/security-policy-manager/>
- <https://metacompliance.com/>
- <https://logicgate.com/>
- <https://www.powerdms.com/>

The mentioned tools allow less flexibility and power to adapt to user cases' needs in terms of security policy expressions, furthermore, they are not open source, and it is risky to envisage their adoption in such a multifaceted environment as the STAR project.

Here below are the reasons supporting the decision to adopt OPA in the STAR project framework.

- OPA is declarative, meaning it expresses policy in a high-level, declarative language that promotes safe, performant, fine-grained controls.
- OPA uses a language purpose-built for policy in a world where JSON is pervasive.
- OPA iterates traverse hierarchies and applies 150+ built-ins like string manipulation and JWT decoding to declare the policies to be enforced.
- OPA is Context-aware, it leverages external information to write the needed policies.
- OPA does not create roles that represent complex relationships that years down the road no one will understand. OPA, instead, writes logic that adapts to the world around it and attaches that logic to the systems that need it, this is exactly what is needed in the STAR project.

OPA is integrated into the SSPM as a running service, allowing specification of policies, like coding, through a high-declarative language called REGO and provides simple APIs to offload policy decision-making.

In the following subparagraph, the function of OPA is explained together with examples of policies evaluation.

2.1.5.1 The logic of OPA and the interaction with SSPM

SSPM offload policy decisions to OPA by executing queries. OPA evaluates policies and data to produce query results (sent back to the client). Policies can be loaded dynamically into OPA via the local filesystem.

OPA is a full-featured policy engine providing the building blocks for enabling better control and visibility over policy in the systems. The SSPM software receives data from the Data Bus in the form of JSON Objects. Such objects are then loaded into OPA from the Data Bus using queue mechanisms that operate synchronously or asynchronously for policy evaluation.

Policies consist of multiple rules that can also refer to other rules and are generated accordingly to the needs of the use cases.

An example of this is a simple RBAC model-based policy shown in Figure 4.

```

1 package play
2
3 # sample rbac assignments
4
5 # user-role assignments
6 user_roles := {
7   "alice": ["engineering", "webdev"],
8   "bob": ["hr"]
9 }
10
11 # role-permissions assignments
12 role_permissions := {
13   "engineering": [{"action": "read", "object": "id123"}],
14   "webdev": [{"action": "read", "object": "server123"},
15             {"action": "write", "object": "server123"}],
16   "hr": [{"action": "read", "object": "database456"}]
17 }
18
19
20 default allow = false
21
22 allow {
23   # lookup the list of roles for the user
24   roles := user_roles[input.user]
25   # for each role in that list
26   r := roles[_]
27   # lookup the permissions list for role r
28   permissions := role_permissions[r]
29   # for each permission
30   p := permissions[_]
31   # check if the permission granted to r matches the user's request
32   p == {"action": input.action, "object": input.object}
33
34   # check the image size
35   image := data.image[_]
36   image.names[_] == input.image.name
37   input.image.size < image.size
38 }

```

Figure 4 RBAC based example of policy

Where the allow function is dependent on the two functions above, it returns a positive or negative result accordingly.

It is possible for policies to utilize other policies for evaluating their functions; this is done by using the output of other policy rules. This happens because policies generate JSON data and JSON data can be referenced and used by the policy itself.

Once a policy has been successfully evaluated, a decision is generated, giving information about the evaluation results. In policies, rules are represented as functions, taking inputs, evaluating them, and returning a result.

When a new event is received by RMS and AI Cyber Defence Module (Json object), it is processed and analysed by OPA. Policies are stored in the Policies repository for persistence. If an attack is predicted, the SSPM generates an x-attack instance (alert). json object and it is sent to Olistic for the risk assessment evaluation.

This process allows to automatize and have a constant control of the data, without the need of having a continuous presence of the security officer. This is also creating a system where new results and/or data is constantly monitored and compared to the model and the labels already present in the Database, which means that whenever different results arrive from the RMS and AI Cyber Defence Module, the correct policies can be activated to signal an attack or a problem enforcing security.

A new course-of-action (CoA) object is generated and inserted in the data layer when a new attack is received. It triggers the generation of a policy object by SSPM. This object is inserted into the data layer and will be received by the risk assessment component (Olistic) and, after the evaluation by the Security Manager, the relative actions will take place on the probes and other components that are affected by the attack.

The probe applies the measures in compliance with the received policy.

Exercises have been made to test OPA and the following demos have been carried out on the REGO playground to test a policy evaluation which is suitable to most of the STAR's use cases.

Figures 5, 6 and 7 define a Role Based Access Control (RBAC) model policy for a generic use case scenario. The use case policy allows users to look at objects in database or server, update their stats, and so on. The policy controls which users can perform actions on which resources. The policy implements a classic Role-based Access Control model where users are assigned to roles and roles are granted the ability to perform some action(s) on some type of resource.

This example shows how to:

- Define an RBAC model in Rego that interprets role mappings represented in JSON.
- Iterate/search across JSON data structures (e.g., role mappings).

Specifically, the first run of the policy evaluation result is "allowed", since the user requested to read an image which size was below the set threshold.

In the second run, the permission is denied to the user since the size image is above the image size set threshold.

```

1 package play
2
3 # sample rbac assignments
4
5 # user-role assignments
6 user_roles := {
7   "alice": ["engineering", "webdev"],
8   "bob": ["hr"]
9 }
10
11 # role-permissions assignments
12 role_permissions := {
13   "engineering": [{"action": "read", "object": "id123"}],
14   "webdev":      [{"action": "read", "object": "server123"},
15                  {"action": "write", "object": "server123"}],
16   "hr":          [{"action": "read", "object": "database456"}]
17 }
18
19
20 default allow = false
21
22 allow {
23   # lookup the list of roles for the user
24   roles := user_roles[input.user]
25   # for each role in that list
26   r := roles[_]
27   # lookup the permissions list for role r
28   permissions := role_permissions[r]
29   # for each permission
30   p := permissions[_]
31   # check if the permission granted to r matches the user's request
32   p == {"action": input.action, "object": input.object}
33
34   # check the image size
35   image := data.image[_]
36   image.names[_] == input.image.name
37   input.image.size < image.size
38 }

```

INPUT
1 - {
2 "action": "read",
3 "image": {
4 "name": "quay.io/openshift/origin-jenkins-agent-base:4.4",
5 "size": 725714890
6 },
7 "object": "id123",
8 "type": "dog",
9 "user": "alice"
10 }

DATA
1 - {
2 "image": [
3 {
4 "names": [
5 "quay.io/openshift/origin-jenkins-agent-base:4.4"
6],
7 "size": 725714891
8]
9]
10 }

OUTPUT
1

Figure 5 Generic Use case security policy, based on image size visualization permission

```

1 package play
2
3 # sample rbac assignments
4
5 # user-role assignments
6 user_roles := {
7   "alice": ["engineering", "webdev"],
8   "bob": ["hr"]
9 }
10
11 # role-permissions assignments
12 role_permissions := {
13   "engineering": [{"action": "read", "object": "id123"}],
14   "webdev": [{"action": "read", "object": "server123"},
15             {"action": "write", "object": "server123"}],
16   "hr": [{"action": "read", "object": "database456"}]
17 }
18
19
20 default allow = false
21
22 allow {
23   # lookup the list of roles for the user
24   roles := user_roles[input.user]
25   # for each role in that list
26   r := roles[_]
27   # lookup the permissions list for role r
28   permissions := role_permissions[r]
29   # for each permission
30   p := permissions[_]
31   # check if the permission granted to r matches the user's request
32   p == {"action": input.action, "object": input.object}
33
34   # check the image size
35   image := data.image[_]
36   image.names[_] == input.image.name
37   input.image.size < image.size
38 }

```

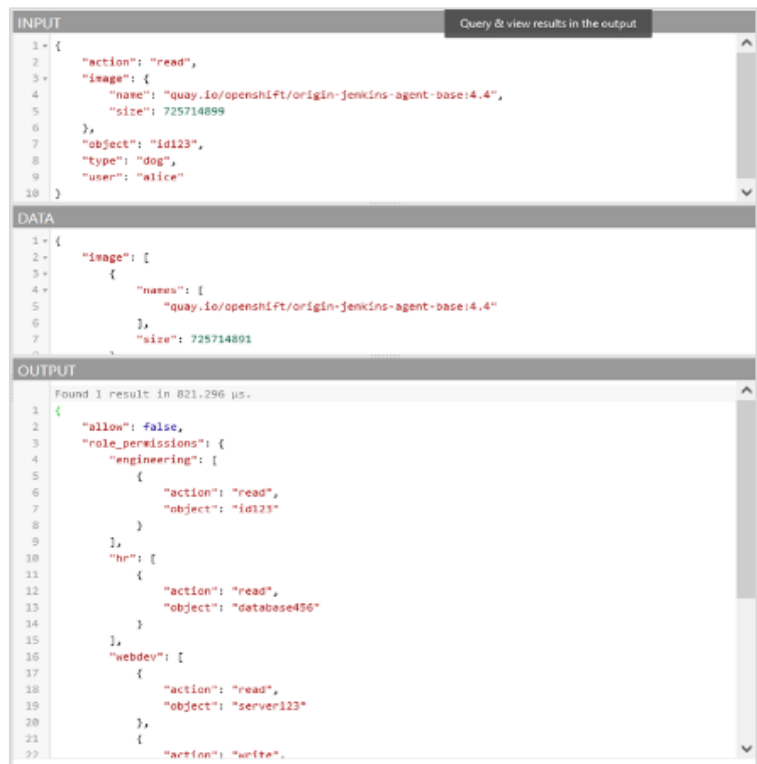
INPUT	
1	{
2	"action": "read",
3	"image": {
4	"name": "quay.io/openshift/origin-jenkins-agent-base:4.4",
5	"size": 725714898
6	},
7	"object": "id123",
8	"type": "dog",
9	"user": "alice"
10	}
DATA	
1	{
2	"image": {
3	"names": [
4	"quay.io/openshift/origin-jenkins-agent-base:4.4"
5],
6	"size": 725714891
7	}
8	}
OUTPUT	
Found 1 result in 651.627 µs.	
1	{
2	"allow": true,
3	"role_permissions": {
4	"engineering": [
5	{
6	"action": "read",
7	"object": "id123"
8	}
9],
10	"hr": [
11	{
12	"action": "read",
13	"object": "database456"
14	}
15],
16	"webdev": [
17	{
18	"action": "read",
19	"object": "server123"
20	},
21	{
22	"action": "write",

Figure 6 Generic Use case security policy, action allowed

```

1 package play
2
3 # sample rbac assignments
4
5 # user-role assignments
6 user_roles := {
7   "alice": ["engineering", "webdev"],
8   "bob": ["hr"]
9 }
10
11 # role-permissions assignments
12 role_permissions := {
13   "engineering": [{"action": "read", "object": "id123"}],
14   "webdev":      [{"action": "read", "object": "server123"},
15                  {"action": "write", "object": "server123"}],
16   "hr":          [{"action": "read", "object": "database456"}]
17 }
18
19
20 default allow = false
21
22 allow {
23   # lookup the list of roles for the user
24   roles := user_roles[input.user]
25   # for each role in that list
26   r := roles[_]
27   # lookup the permissions list for role r
28   permissions := role_permissions[r]
29   # for each permission
30   p := permissions[_]
31   # check if the permission granted to r matches the user's request
32   p == {"action": input.action, "object": input.object}
33
34   # check the image size
35   image := data.image[_]
36   image.names[_] == input.image.name
37   input.image.size < image.size
38 }

```



The screenshot displays a query interface with three main sections: INPUT, DATA, and OUTPUT. The INPUT section shows a request with fields: action: "read", image: { name: "quay.io/openshift/origin-jenkins-agent-base:4.4", size: 725714899 }, object: "id123", type: "dog", user: "alice". The DATA section shows a list of image objects with fields: image: { names: ["quay.io/openshift/origin-jenkins-agent-base:4.4"], size: 725714891 }. The OUTPUT section shows the result: Found 1 result in 821.296 µs. The result is a JSON object: {"allow": false, "role_permissions": {"engineering": [{"action": "read", "object": "id123"}], "hr": [{"action": "read", "object": "database456"}], "webdev": [{"action": "read", "object": "server123"}, {"action": "write", "object": "server123"}]}}

Figure 7 Generic Use case security policy, action denied

OPA showed a good power of expression and generalization to cover a wide spectrum of use cases.

3 Runtime Monitoring System

3.1 Architecture

As mentioned above, the Runtime Monitoring System (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 Policy 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. RMS component has been adapted and extended from previous EU projects like H2020-SecureIoT and H2020-IoTAC projects.

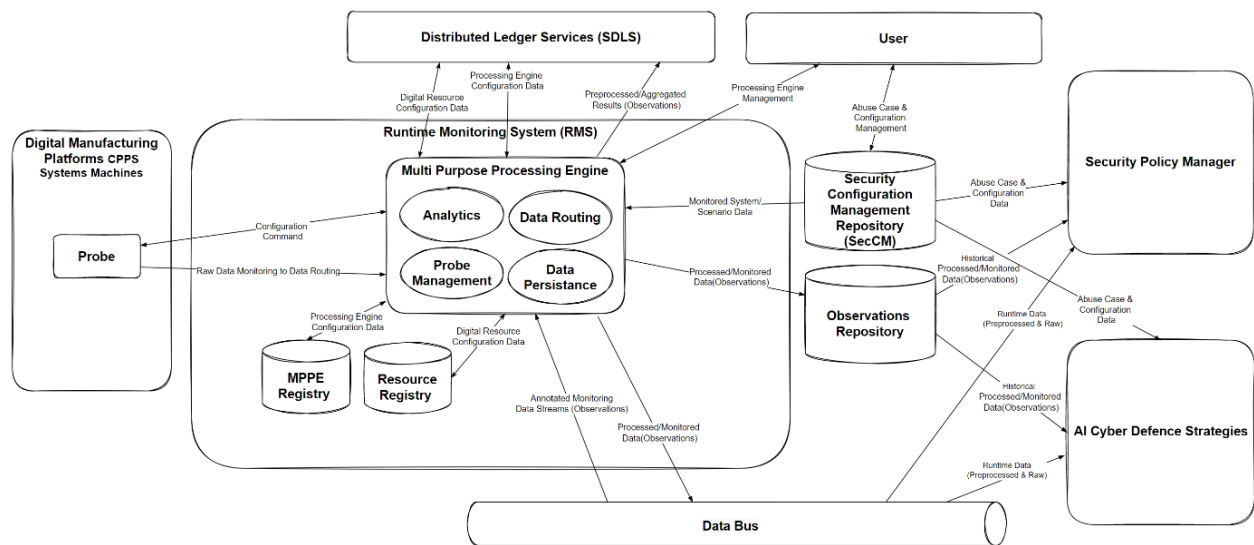


Figure 8 RMS Data Flow Diagram

As shown in Figure 8 above RMS offers the following primitive 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.
- **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 8 above) 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 8 above in the STAR project we have two main RMS external applications which are the Security Policy 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 8 above:

- **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 Runtime Monitoring System 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

Runtime Monitoring System (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 9 below, and described in Section 3.2.1.

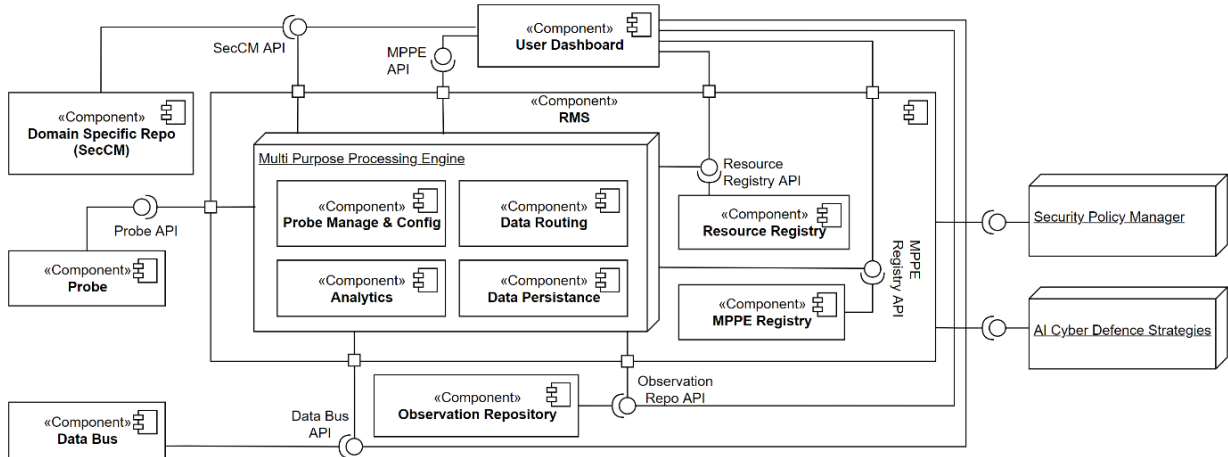


Figure 9 RMS component Diagram

3.2.1 Interface Specification

Table 1 provides a list of the interfaces depicted in Figure 9 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 1 RMS Interface Specification

No	API	Description	Provided	Required
1	Probe API	Probe API enables the control of a Probe by exposing configuration (sending a probe configuration file) and control (start/stop) interfaces.	X	
2	PMC API	Probe Management & Configuration API exposes appropriate endpoints that enables the discoverability, configurability, and management of the deployed probes.	X	
3	MPPE API	Multi-Purpose Processing Engine API exposes appropriate endpoints that enable the discoverability, configurability, and management of deployed processors.	X	
4	MPPE Registry API	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	DR API	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	

No	API	Description	Provided	Required
6	AR API	Automatic Reconfiguration API exposes appropriate endpoints that enable the configuration and control and triggering of the Automatic Reconfiguration component.	X	
7	PR DB API	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	Observation Repo API	Observation Repository API exposes appropriate endpoints that enable the discoverability and usage of captured, pre-processed, and processed data		X
9	Data Bus API	Data Bus API exposes appropriate endpoints that enable the temporary persistence, publishing, subscribing and retrieval of data streams.		X

In Figure 10 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 it is performed 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.

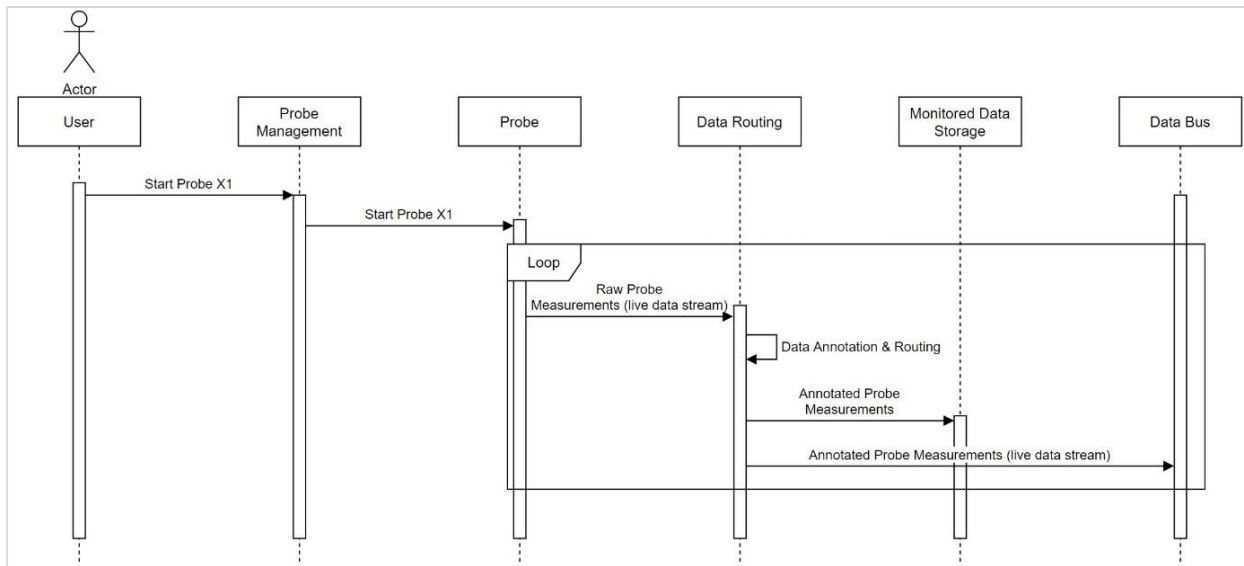


Figure 10 Probe data storage sequence diagram

3.3 Probe availability and Usage

For the RMS component’s implementation, we are using elastic stack¹ 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:** to collect monitored data (i.e., CPU utilization data) by using Elastic MetricBeats deployed to the Manufacturing Plant.
- **Logstash:** Raw monitored Data are transformed and filtered to match the used Data Model (i.e., Observations) and identified rules (i.e., report only values above 90%).

¹ <https://www.elastic.co/elastic-stack/>

- **Kafka & ElasticSearch:** the collected pre-processed data are published to the Data Bus (Kafka) to be accessed by the Security Policies Manager & ElasticSearch for permeate persistence, visualization and monitoring.
 - Security Policies Manager retrieves the pre-processed data by the Data Bus (Kafka) 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 11 below.

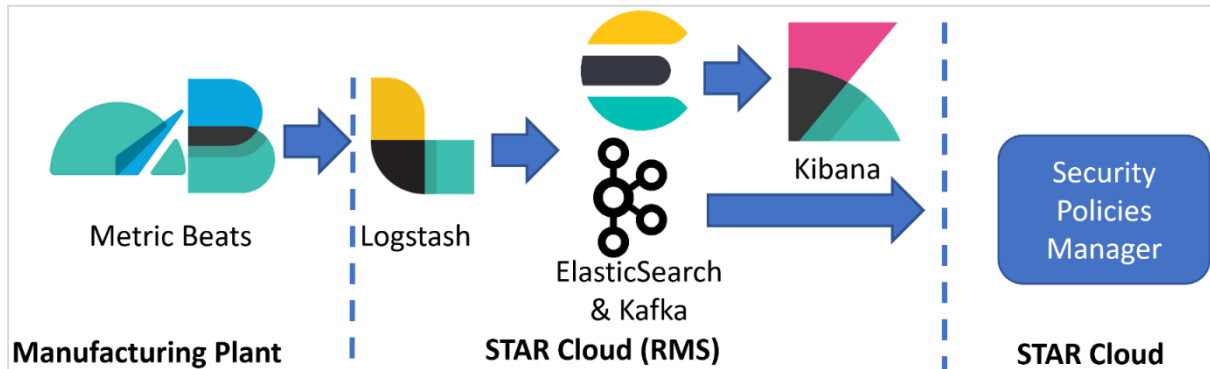


Figure 11 RMS infrastructure flow

As mentioned above we are using Elastic Beats² 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 is several different Elastic Beats supported for retrieving various type and format of data. The main options are offered by Elastic³ but there are also third-party ones offered by the Elastic community⁴. The initial options investigated in STAR to be used for collecting monitored data are the:

- **Filebeat**⁵ which tails and ships log files.
- **Metricbeat**⁶ which fetches sets of metrics from the operating system and services
- **Packetbeat**⁷ which monitors the network and applications by sniffing packets

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 12 below.

² <https://www.elastic.co/guide/en/beats/libbeat/current/index.html>

³ <https://github.com/elastic/beats>

⁴ <https://www.elastic.co/guide/en/beats/libbeat/master/community-beats.html>

⁵ <https://github.com/elastic/beats/tree/master/filebeat>

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

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

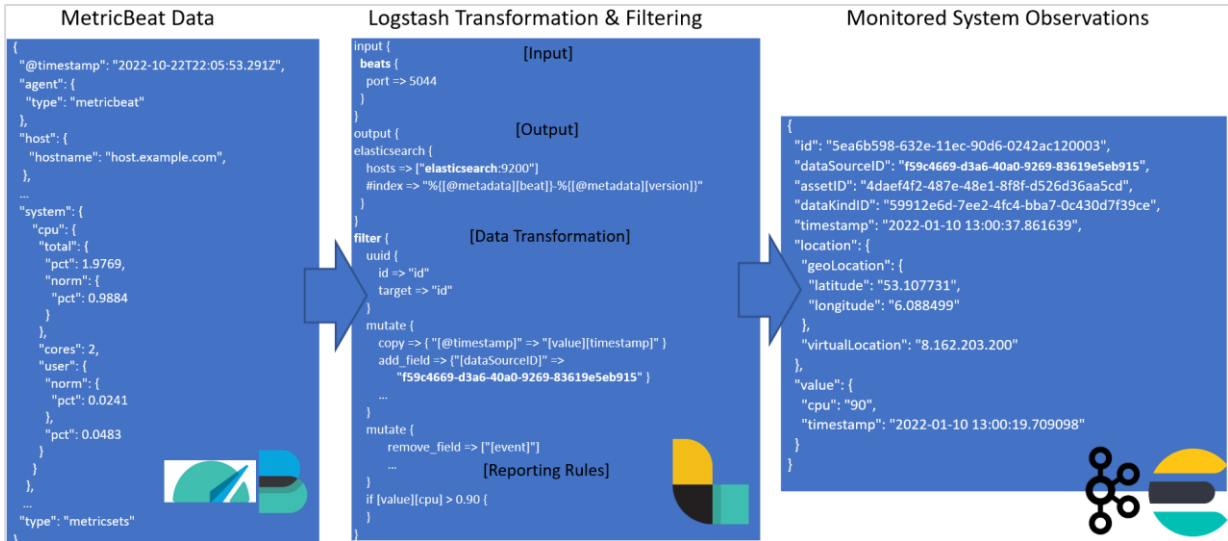


Figure 12 RMS data collection, Transformation & Filtering example

3.4 GUI

RMS will offer a monitoring dashboard for depicting the collected monitoring data. The dashboard will be based on Elastic Kibana⁸ technology and will offer querying functionalities over the Observation repository (which is based on Elastic Search⁹) and appropriate widgets to report to the security expert the monitored values. This dashboard will be an intermediate to the Risk Assessment and Mitigation dashboard (which will be offered by Olistic tool). In Figure 13 and Figure 14 below, we can see an examples of a system utilization monitoring by providing the data discovery view and the dashboard view respectively.

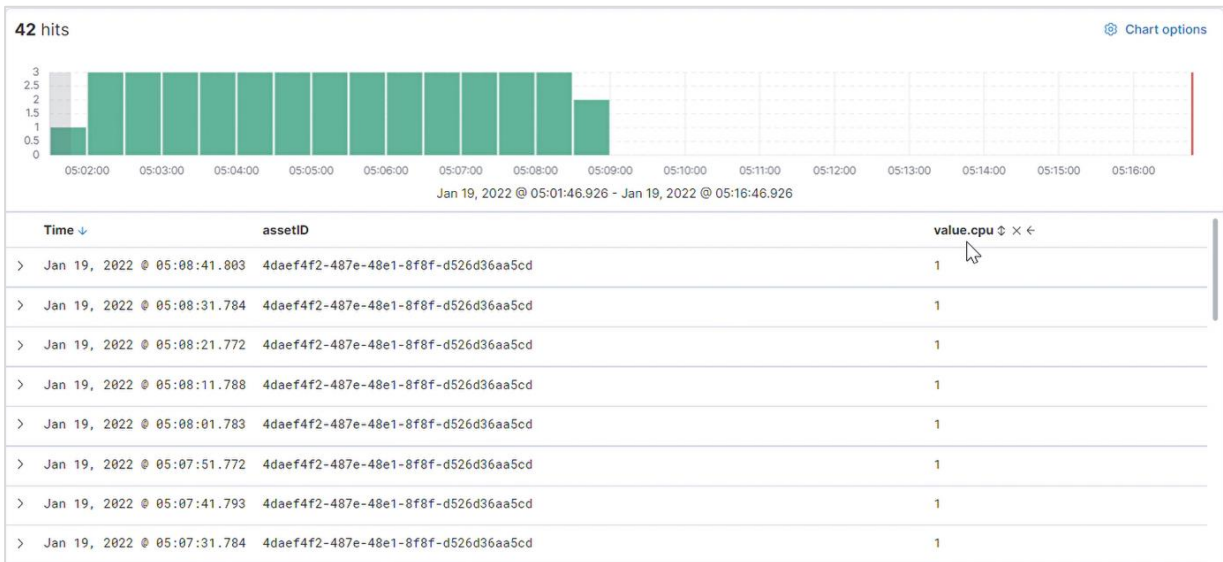


Figure 13 Kibana discovery view dashboard

⁸ <https://www.elastic.co/kibana/>

⁹ <https://www.elastic.co/elasticsearch/>

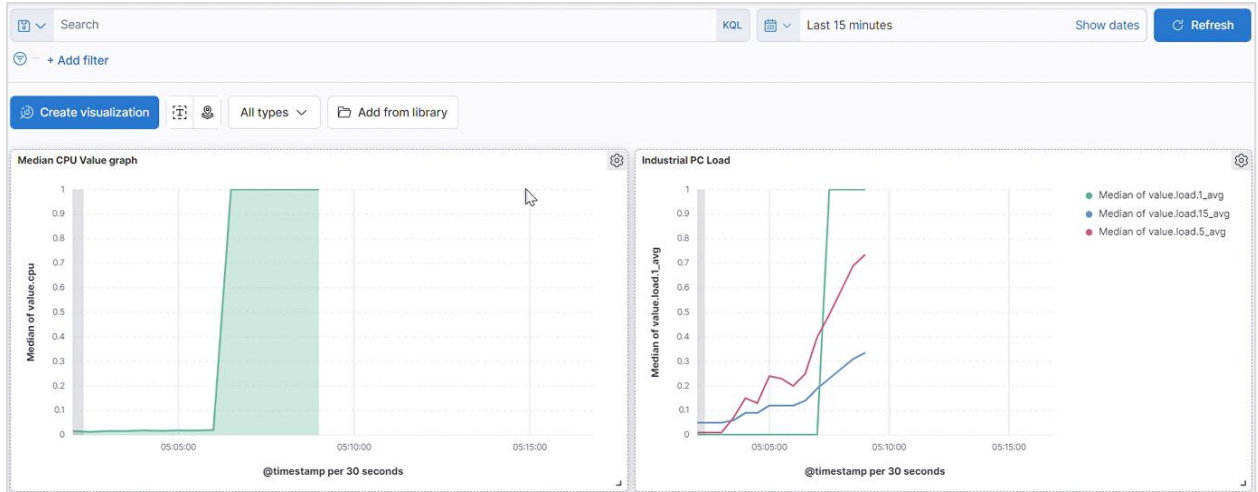


Figure 14 Kibana dashboard view

4 OLISTIC

4.1 Architecture

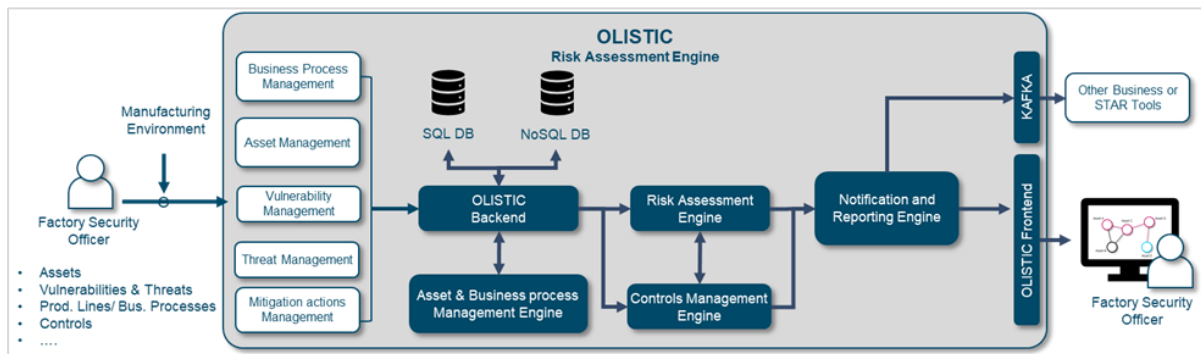


Figure 15 OLISTIC Internal Component Architecture

OLISTIC will be used as one of the dashboards of the AI Security and Data Governance layer of STAR and will enable 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 15 on the left side of the figure 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. 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 interfaces and the API calls that enable the interaction between the frontend and the backend of OLISTIC will be documented in the context of the deliverables of WP6 and in D3.6.

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, MySQL – 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.

As noted, OLISTIC uses both relational and non-relational databases. The former is used to store structured data (e.g., credentials, organizational profile), while the NoSQL is used to store semi-structured data that change frequently (e.g., Vulnerability reports).

The Asset and Business process management engine allows the factory security officer to declare the organizational assets taking part in the production lines that need to be monitored. This declaration is serialized in a strict format which is addressed as “Asset Cartography”. The cartography is then enriched with potential vulnerabilities and threats that are relevant to the individual assets. The cartography along with the linked information will be intuitively visualized by a graph. Such a graph can be seen in Figure 17.

The Risk Assessment Engine undertakes the risk quantification upon triggering events, or on demand by the security officer. Thus, this component is needed to perform the appropriate steps required for the conduction of a risk assessment for the whole organization or even for a specific business service/production line. During the assessment, the tool traverses the asset cartography to map the assets with identified vulnerabilities and threats.

The KAFKA component is the technology that enables the storing and sharing of the risk assessment output. Its purpose is to facilitate the sharing of the risk assessment output with other components of STAR.

The **Notification and Reporting Engine** is responsible to provide push notifications to the security analyst regarding any type of messages that are published in the pub/sub queue. Since the risk assessment may involve time-consuming operations (e.g., the conduction of a vulnerability assessment, the calculation of risks, the processing of data sources) every time 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.

4.2 STAR Risk Assessment methods and models

4.2.1 Asset Modelling & Visualization

The risk assessment will consider possible vulnerabilities and attacks (Threats) that can be exploited or target a set of assets involved in the production lines being monitored. In this regard, OLISTIC makes use of interdependency graphs as the enabler for creating asset cartographies. Based on this, a manufacturing environment is modelled as chains of assets and along with interdependencies which are replicated in a digitally reflected environment that will enable the security officer to manage the assessment process and take advantage of the sophisticated capabilities offered by the STAR AI Security and Data protection layer.

The cyber-physical asset cartography undertakes this digital representation of the monitored environment. This component receives input from the security officer (or the manager of the production line) regarding the assets comprising the monitored ecosystem, including, but not limited to, the hardware/software assets, type of assets, Operating Systems and applications, and other capabilities of assets and services (or business processes). In the scope of STAR, we consider that the analyst has the necessary domain and infrastructure knowledge to instantiate the digital reflection of the deployed environment through the structured process offered by OLISTIC.

Given the above, the Asset modelling and Visualisation component capitalises on the Interdependency Graphs that will be used to create a digital representation of the monitored environments, considering the relationship types among the assets. STAR will use the interdependency types *IsConnectedTo*, *IsUsedBy*, *IsProcessedBy*, *isLocatedIn*, *isStoredOn* and *IsInstalledOn* to annotate the relation among assets.

These relations are not only used to denote connections among tangible ICT assets, but also intangible ones, such as data. The *IsConnectedTo* and *IsInstalledOn* represent the network and systemic inter-dependencies, the *IsUsedBy* and *isLocatedIn* represent physical inter-dependencies, the *IsProcessedBy* and *isStoredOn* represent logical inter-dependencies.

From the implementation perspective, the graph model is based on Neo4j to offer a scalable graph to visualize, analyse and navigate through the assets that a factory needs to manage along with other various interconnected entities and properties.

OLISTIC offers a GUI based on which the security officer can manage the infrastructure assets. As is highlighted in Figure 16, the analyst has a list of assets, she can create new assets, visualise the deployment (see Figure 17), while several other actions are given (Edit, Clone, Customization, Footprint, Deactivate, Delete). OLISTIC supports different kind of templates for the representation of assets. These templates are documented in the following subsection.

Based on the templates, Section 6.3 of this deliverable offers an instantiation of the STAR environments of the use cases for the creation of the respective interdependency graphs.

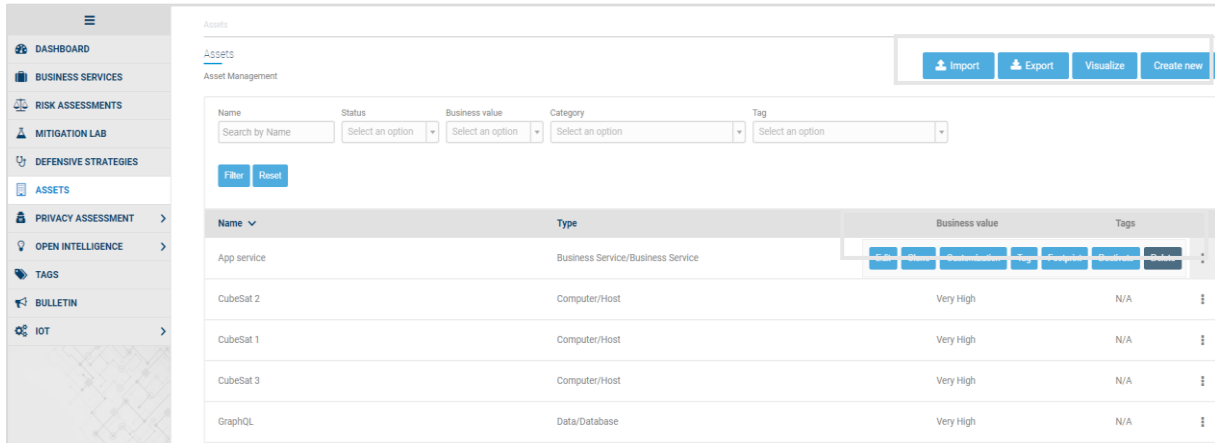


Figure 16 ASSETS OLISTIC GUI and additional operation options

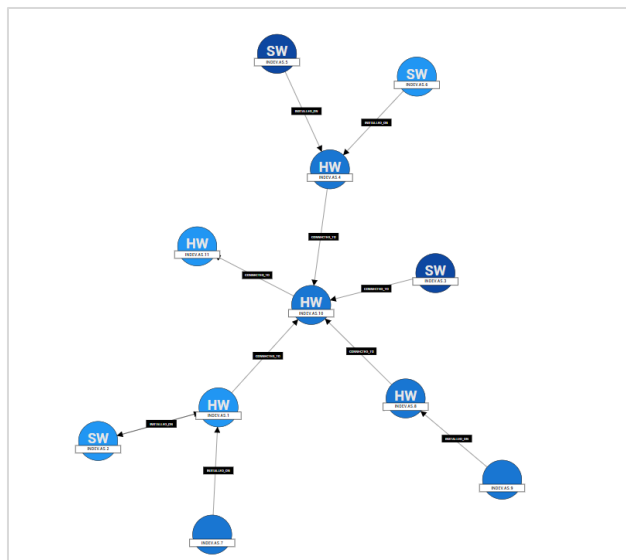


Figure 17 Example of interdependency graphs in OLISTIC

4.2.1.1 Asset templates

In principle, three different asset templates have been designed to provide a structured way to the security officer to insert the necessary information. More specifically, the assets templates refer to the software, the hardware and the data assets.

The **software asset template**, as depicted in Figure 18, the officer is requested to add the following aspects:

- Name
- Business value (Very low, Low, Medium, High, Very high) (Used as an indicator for denoting the criticality of the asset and the prioritisation of mitigation actions)
- Run privilege
- Vendor (optional)
- Product (optional)
- Version (optional)
- Relationships with other assets
- Arbitrary values in the form of key-value format

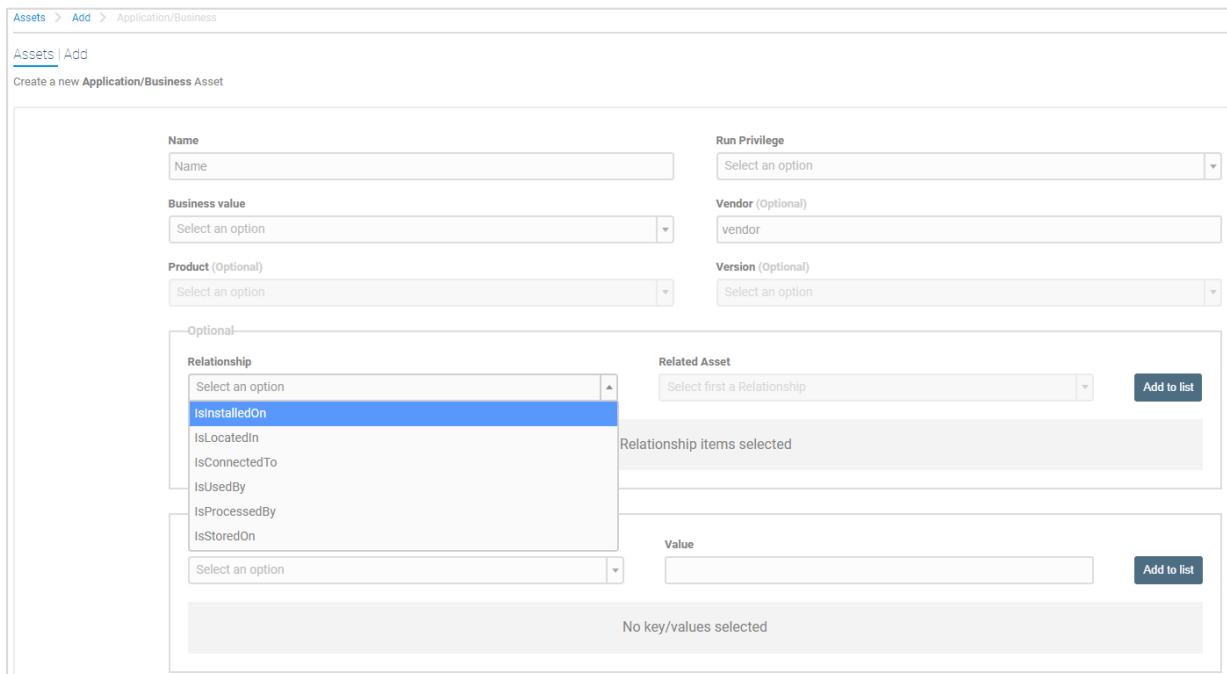


Figure 18 Software asset template

The **hardware asset template**, as depicted in Figure 19, the analyst is requested to add the following aspects:

- Name
- Business value (Very low, Low, Medium, High, Very high) (Used as an indicator for denoting the criticality of the asset and the prioritisation of mitigation actions)
- Vendor (optional)
- Product (optional)
- Version (optional)
- Relationships with other assets
- Arbitrary values in the form of key-value format

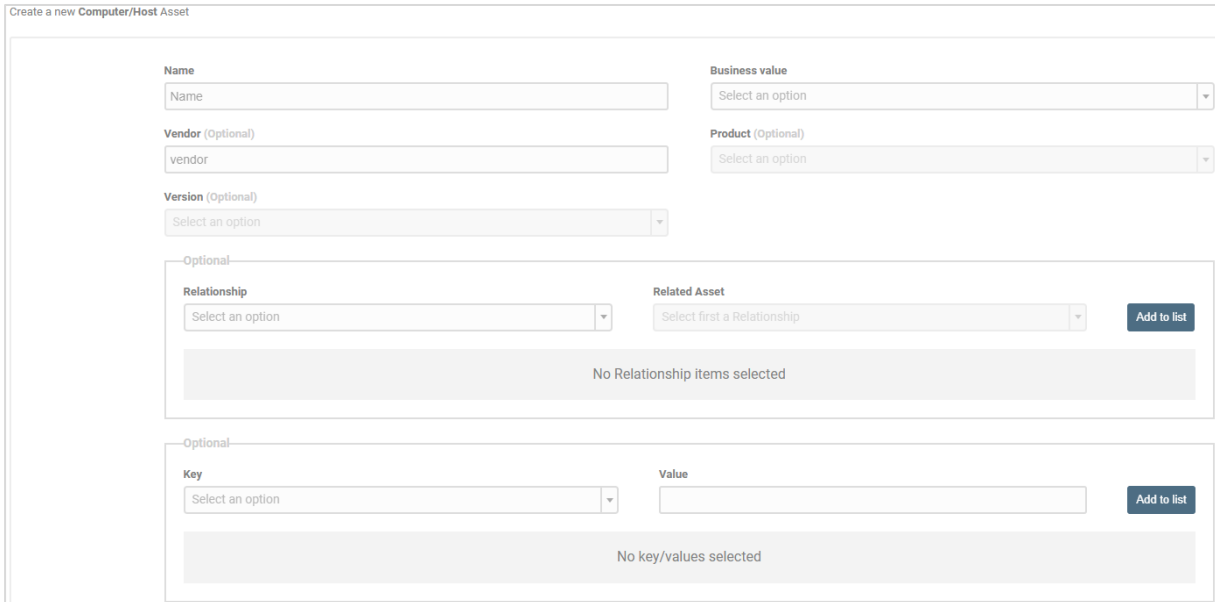


Figure 19 Hardware asset template

The **data assets template** presented in Figure 20 incorporates the following aspects:

- Name
- Business value (Very low, Low, Medium, High, Very high) (Used as an indicator for denoting the criticality of the asset and the prioritisation of mitigation actions)
- Relationships with other assets
- Arbitrary values in the form of key-value format

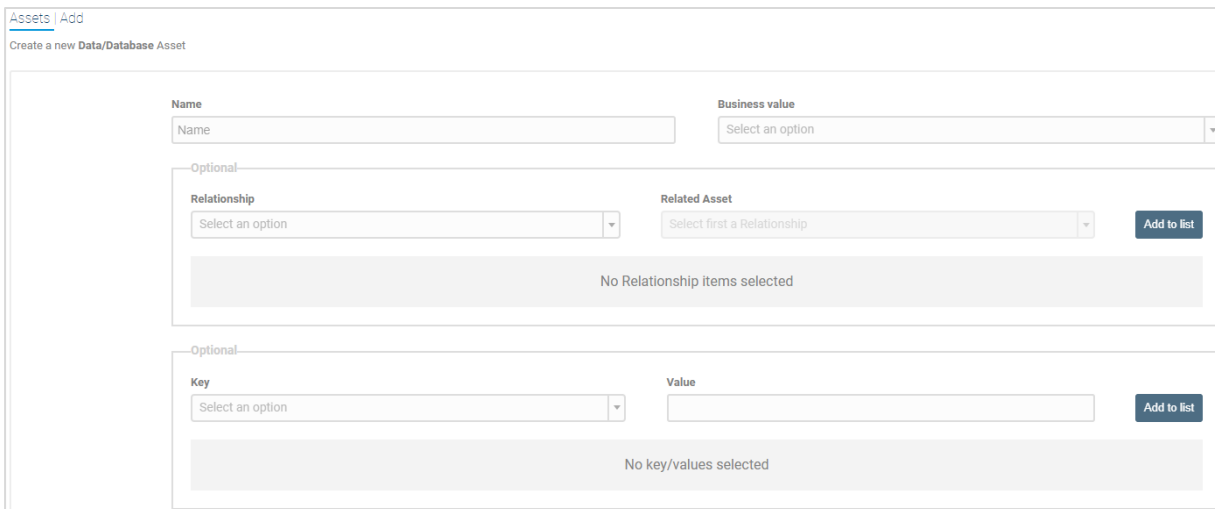


Figure 20 Data assets template

4.2.2 Conceptual OLISTIC Risk Assessment meta-model

In Figure 21, we provide the conceptual view of the risk assessment meta-model which is used to drive the risk assessment workflow in the context of STAR, as well as to document the risk quantification approach used in OLISTIC. More specifically, beyond the interdependencies that have been analysed above, it should be clarified that each asset may have one or more vulnerabilities and is related to specific dependencies. This dependency is shown intuitively in Figure 21. Each vulnerability can be exploited using one or more threats.

A successful attack may lead to an asset exploitation that is accompanied by a specific impact. The impact conceptualizes the damage and is classified in confidentiality, integrity and availability. An exploitation may be prevented in case of the enforcement of control elements. Control elements mitigate a vulnerability or a threat. Finally, a production line contains many assets which may be cyber and physical. To infer the risk per asset, information regarding its vulnerabilities and impact level upon exploitation must be combined. The identification of such control elements constitutes the optimal defence strategy (Mitigation Strategy) tailored to the calculated cyber-risks. In the context of STAR, these controls come as mitigation measures that the security officer must consider and based on her/his domain knowledge, believes that can mitigate a risk.

Overall, the defined meta-model can sufficiently consider the relations among assets, threats, vulnerabilities, attack types and controls. More specifically, the utilisation of the interdependency graphs enables the security officer to consider the relations among the assets of the factory dependencies among different actors, personnel, data and processes. Hence, the risk quantification enables the security officer to perceive how vulnerable or attacked assets may affect the manufacturing process and support the decision making when it comes to the consideration of mitigation measures that can regulate the risk of critical assets in the production line.

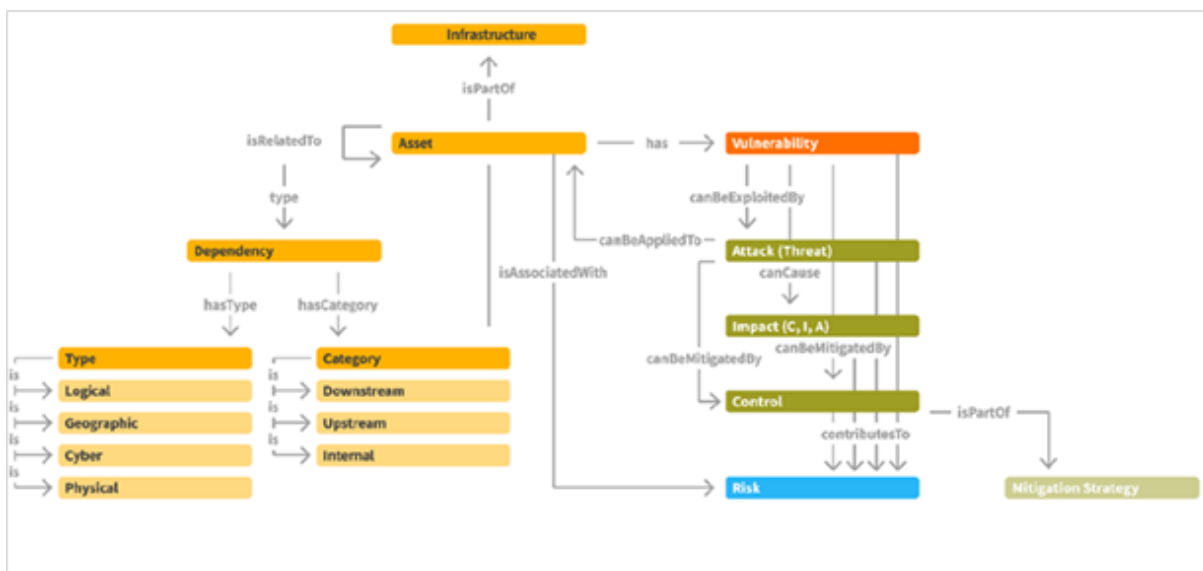


Figure 21 OLISTIC Risk Assessment Meta-model

4.2.3 Risk, Vulnerability and Threat modelling

The risk assessment aggregates the inputs of the Vulnerability, Threat and Impact analysis with the aim to perform the final risk assessment. The mapping of all available assets in the interdependency graph, as well as the identified vulnerabilities and impact, based on the graphical representation, will be used to calculate the overall risk. OLISTIC offers a methodology for the quantification of risks, based on the use of the CVSS scoring system, which is used for the quantification of the impact of a vulnerability. Thus, the **Risk Level (RL)** of an asset, represents how dangerous a threat is to the specific asset. More specifically, RL quantifies the risk of an asset taking into consideration all the associated vulnerabilities ignoring the assets’ dependencies and relationships. The RL can be calculated as a multiplication of the imposed Threat level, Vulnerability and Impact Levels (VL and IL, respectively) as follows:

$$RL = TL \times VL \times IL$$

The RL, as well as the TL, VL, and IL are values form the interval [0.0, 1.0], with 0 indicating the lowest level (value) and 1.0 denoting the most severe risk case. However, to make the perception of risk easier and more flexible, OLISTIC maps the continuous [0.0, 1.0] interval into a 5-tier risk classification, as shown in the following table.

Table 2 Continuous interval of OLISTIC's risk classification

Value Range	Qualitative Value
[0.80, 1.00]	Very high (VH)
[0.60, 0.80)	High (H)
[0.4, 0.60)	Medium (M)
[0.20, 0.40)	Low (L)
[0.0, 0.20)	Very Low (VL)

To perform all the above-mentioned calculations, OLISTIC implements the methodology and the risk quantification formula and provides the necessary modelling techniques to do so, as reported in the following sections.

4.2.3.1 Vulnerability modelling

A core aspect of the cyber-risk quantification operation is the vulnerability modelling. OLISTIC is based on the CVSS scoring system¹⁰ to define already known technical vulnerabilities or even zero-day, custom and system-specific vulnerabilities and quantify the generated risk. In this release of the Risk Assessment Framework, OLISTIC supports CVSS v2.0 and it is expected that OLISTIC will be extended with CVSS v3.1 at the 2nd release of the AI Security and Data governance layer. This development will be documented in D3.6. Each Vulnerability is represented by a) the **Base Metrics** that are represented by the Access Vector (AV), the Access Complexity (AC) and the Authentication (Auth), b) the **Impact Metrics** that are represented by the Confidentiality Impact, Integrity Impact and Availability impact. Figure 22 presents OLISTIC's UI for the Vulnerability modelling.

¹⁰ <https://www.first.org/cvss/>

ID CVE-2020-11899	Base Score 4.80
Exploitability Subscore 6.50	Impact Subscore 4.90
Access Vector ADJACENT_NETWORK - Adjacent Network	Authentication NONE - None
Access complexity LOW - Low	Confidentiality impact NONE - None
Integrity impact PARTIAL - Partial	Availability impact PARTIAL - Partial
Security protection Leads to gaining ADMINISTRATIVE access	Privacy impact (Optional) VH - Very High
Library (Optional) Select an option	Description (Optional) Description
Choose applicable asset categories (Optional) Select one or more asset categories	

Figure 22 Vulnerability modelling template

OLISTIC takes advantage of online sources to keep its internal knowledge base synced with the latest vulnerabilities that are being reported by the community. More specifically, the NVD is the main source of vulnerabilities, but it can be easily extended, if needed, as the implementation is based on interoperable data models. In addition, the OLISTIC operator can define arbitrary vulnerabilities based on her domain and infrastructure knowledge by completing the template given in Figure 22. In this way, the operator can work with a higher degree of freedom through the definition of custom vulnerabilities that can be adjusted to the nature of the threat landscape of each use case.

4.2.3.2 Threat modelling

The probability of the occurrence of a threat is a factor which may vary based on several factors, such as the nature of an infrastructure per se, the accessibility provided to the targeted assets, and its definition is a subjective matter which is usually undertaken by the security officer of the infrastructure. The officer may base the decision on his/her domain knowledge, the history of logged events and valuable information coming from online repositories and other experts’ opinions. In the context of the STAR approach, this likelihood is expressed using a semi-quantitative, five-tier scale. A Threat Level (TL) based on this likelihood is assigned to each threat.

Given the above, OLISTIC offers the corresponding template for the definition of the TL of threats. The definition of threats is based on the template given in Figure 23.

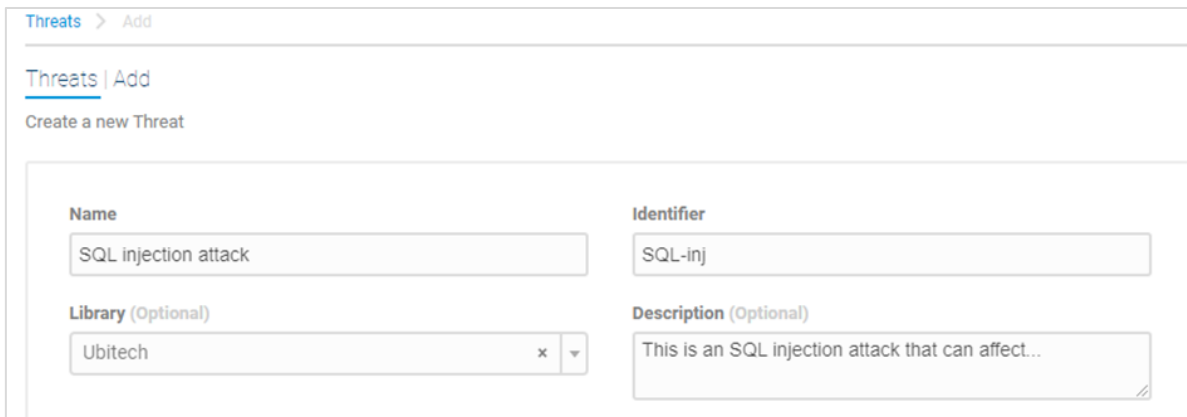
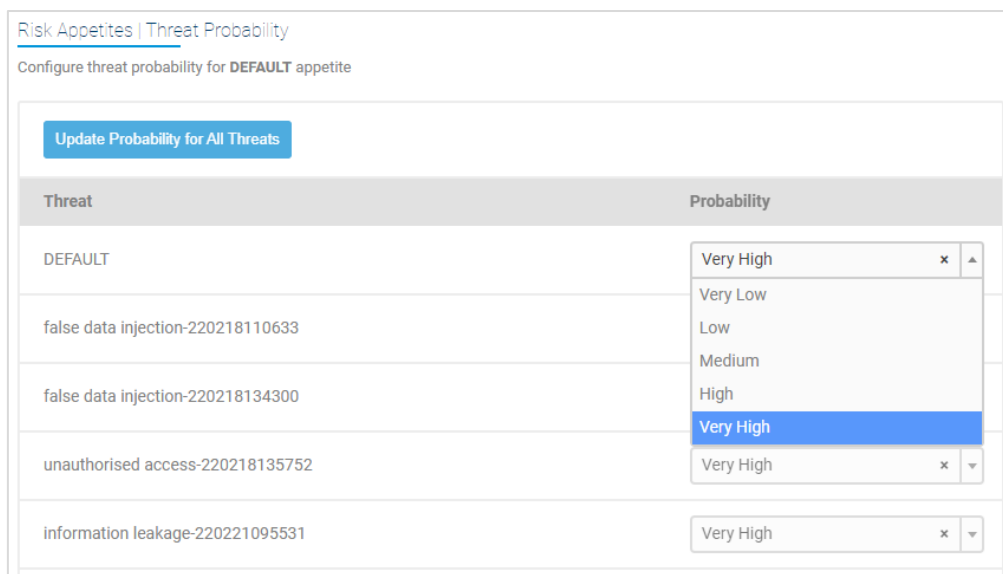


Figure 23 Threat template

However, the TL, i.e., the probability of a threat to occur is adjusted by the risk appetite template. An example is given in Figure 24, where the operator can adjust the TL choosing among the 5-tier values from “very low” (VL) to “very high” (VH). OLISTIC API offers CRUD operations for the adjustment of the risk appetite in case of external entities or other STAR tools need to manage the TL. Mainly, the STAR Security Policies manager will take advantage of the OLISTIC API to manage the threat level accordingly.



Threat	Probability
DEFAULT	Very High
false data injection-220218110633	Very Low
false data injection-220218134300	Low
unauthorised access-220218135752	Medium
information leakage-220221095531	High
	Very High

Figure 24 Risk Appetite template

4.2.3.2.1 Common Attack Patterns against STAR architecture

In order to be able to defend an ICT infrastructure, a security officer needs to be aware of the vulnerable points of it. In order to have a comprehensive view of possible ways to be attacked and to minimise the imposed cyber risk, defenders need to be aware of the attack patterns that adversaries may employ against the infrastructure.

In this regard, the cyber security community has defined a Common Attack Pattern Enumeration and Classification dictionary and classification taxonomy (MITRE - Common Attack Pattern Enumeration and Classification (CAPEC) - <https://capec.mitre.org>) for documenting the possible attack tactics. An attack pattern is a description of the common attributes and approaches employed by adversaries to exploit known weaknesses in cyber-enabled assets.

Each attack pattern captures knowledge about how specific parts of an attack are designed and executed and gives guidance on ways to mitigate the attack's effectiveness. Attack patterns help those developing applications or administrating cyber-enabled capabilities to better understand the specific elements of an attack and how to stop them from succeeding. Following this "know your enemy" strategy, a defender can increase the robustness of the deployed defensive mechanisms, but more importantly, is in position to identify the imposed risks and have proper planning for mitigating them.

In the context of STAR, CAPEC will be used as the framework for representing potential threats against the manufacturing systems. The security officer will use CAPEC to capitalise on a globally accepted approach for expressing common attack patterns that can be utilised against specific critical assets of the manufacturing environment.

4.2.3.2.2 ATLAS MITRE - Adversarial Threat Landscape for Artificial-Intelligence Systems for the STAR architecture

Following the reasoning of the CAPEC framework for having a description of the common attack patterns used in legacy ICT systems, in STAR we make use of the ATLAS Mitre for the description of the AI-based threats that may threaten the AI systems taking part in the manufacturing process.

Thus, as reported in D3.2, ATLAS is modelled after the MITRE ATT&CK® framework and its tactics and techniques are complementary to those in ATT&CK. It enables researchers to navigate the landscape of threats to machine learning systems. There are a growing number of vulnerabilities in ML, and its use increases the attack surface of existing systems. Atlas was developed to raise awareness of these threats and present them in a way familiar to security researchers. The goal of ATLAS is to connect the research with the actual mitigations actions that the industry should be prepared to take, due to the attention that attacks on ML systems have started to attract. Therefore, a threat taxonomy has been created, to provide a reference point to cybersecurity actors regarding the types of attacks that exist, based on real-world examples.

In the context of STAR, the AI Cyber Defence tool produces alerts based on the identification of poisoning or evasion attacks against AI-enabled manufacturing systems. Thus, ATLAS MITRE will be used as the framework to describe the existence of such threats in the context of the risk assessment framework.

4.2.3.3 Attack scenario modelling

The **Attack Scenario** is the key aspect of the cyber-risk quantification process as it combines the a) threat level, b) the asset and c) the corresponding vulnerability. In fact, this notion correlates the necessary information to define that a threat occurred against an existing asset of the deployment. Based on the existence of an attack scenario, the risk assessment process will quantify the risk based on the risk formula given at the beginning of this section.

Figure 25 offers an instance of the attack scenario template, while Figure 26 illustrates the main UI for the management of all the attack scenarios.

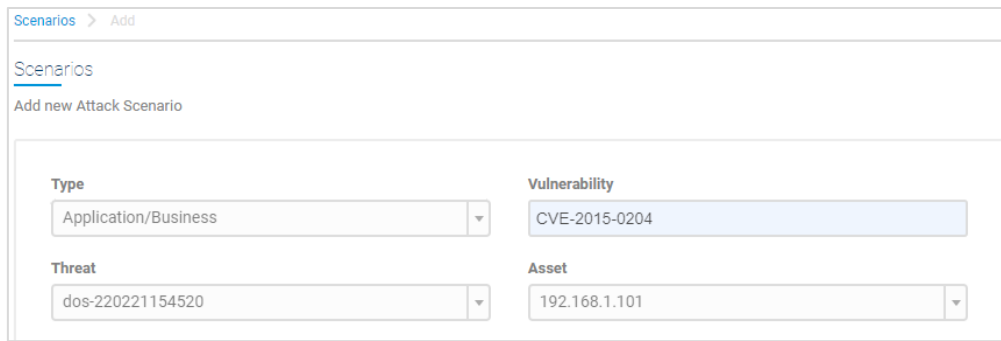
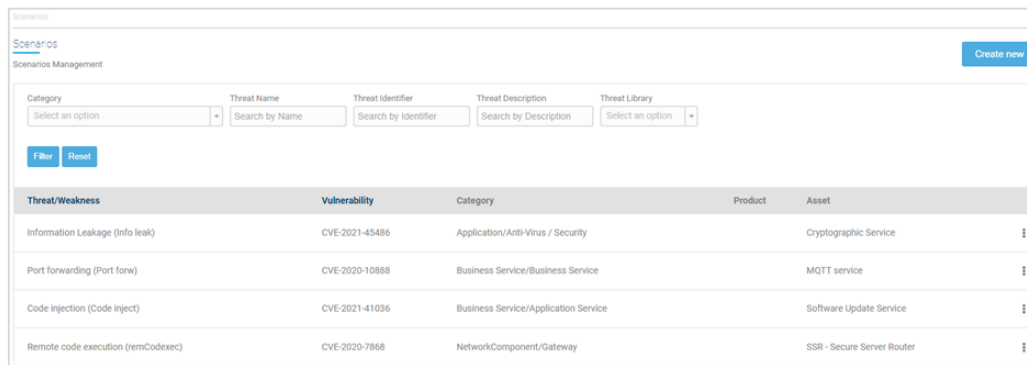


Figure 25 Attack Scenario template



Threat/Weakness	Vulnerability	Category	Product	Asset
Information Leakage (Info leak)	CVE-2021-45486	Application/Anti-Virus / Security		Cryptographic Service
Port forwarding (Port forw)	CVE-2020-10888	Business Service/Business Service		MQTT service
Code injection (Code inject)	CVE-2021-41036	Business Service/Application Service		Software Update Service
Remote code execution (remCodexec)	CVE-2020-7868	NetworkComponent/Gateway		SSR - Secure Server Router

Figure 26 Attack scenarios overview

4.3 Input

As denoted also in the general architecture of Figure 2, OLISTIC receives from the SPM 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 is a vital piece of information that the risk assessment process will need and coincides with the attack scenario model which was described in section 4.2.3.3. In addition, the data model of the attack scenario can be seen also in Figure 40.

4.4 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 be been put 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.

4.5 GUI

The previous subsections have provided several images that correspond to the data templates used in the GUI of OLISITC to enable mainly data entry. However, OLISTIC comes with a complete GUI 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. Given the current version of OLISITC, Figure 27 offer an instance of the dashboard of OLISTIC that offers an overview to the security officer, while Figure 28 illustrates the environment where new risk assessments can be triggered and already executed ones can be processed and investigated.

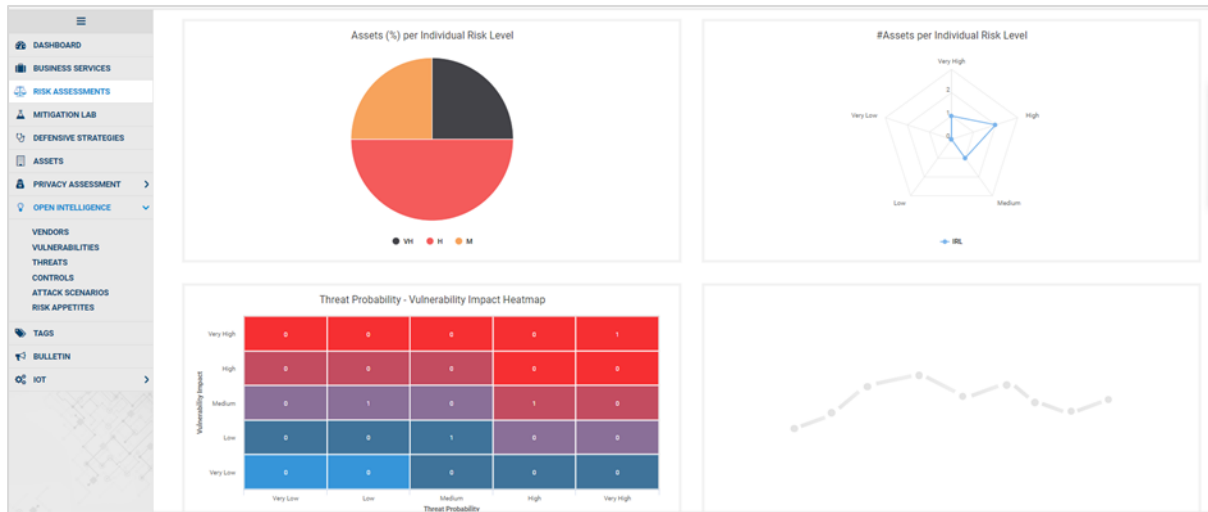


Figure 27 OLISTIC Dashboard

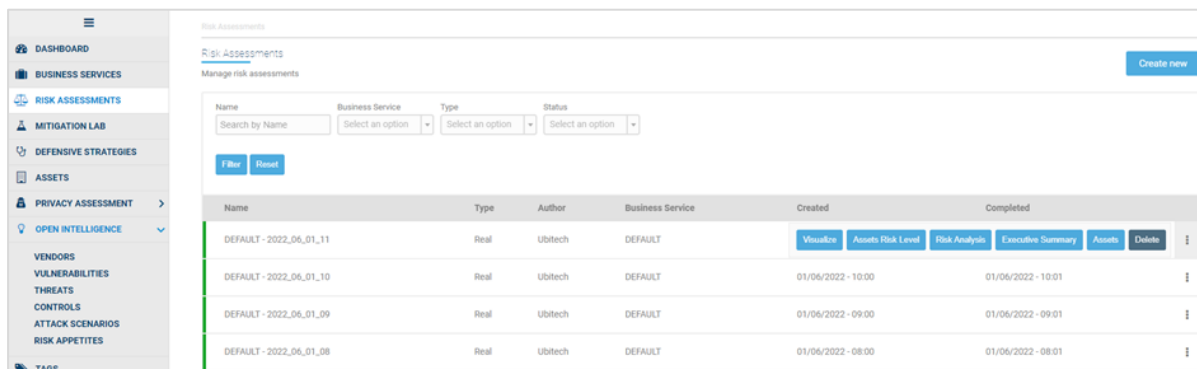


Figure 28 OLISTIC Risk Assessment management environment

5 Star Security Policy Manager

5.1 Architecture

The SSPM is implemented as a Python application that wraps OPA as an external service and acts as a middle-man regarding the other components by gathering the input from RMS and the XAI component, evaluating the defined policies on the given input, and returning the output. The input is collected, using a Kafka consumer, via the specific topics published by the RMS and the AI Cyber Defence components on the Data Bus. OPA manages both input and output in JSON format.

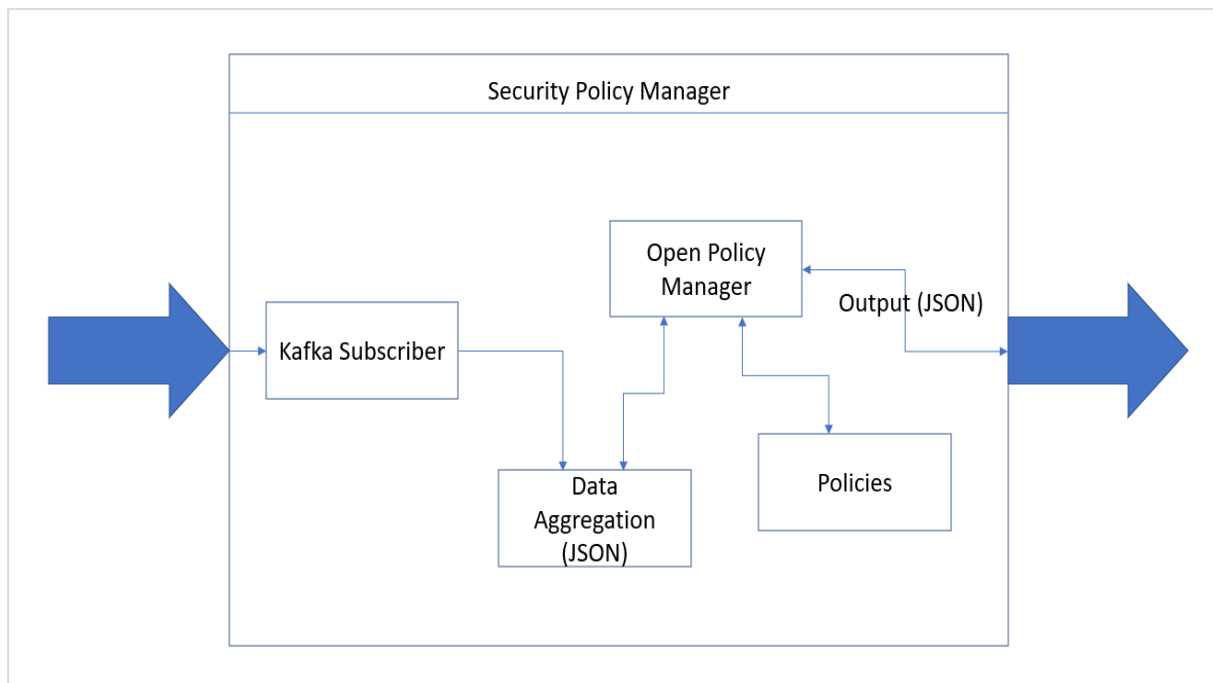


Figure 29: SSPM high level architecture

Figure 29 depicts a scheme of the architecture with the SSPM containing OPA as its main service, a module that acts as a data preparation on the input received that is used during policy evaluation, and the policies that will be evaluated depending on the input received.

These features are wrapped in a Python application that converses with OPA through Queries passing the input data and telling the service which policy to evaluate.

Policies can also be created and updated, and once a policy decision has been made, a JSON object is generated and passed as output to the external services.

SSPM supports the logic for multiple kinds of policies, evaluating the input received through the Kafka queue from the RMS and the AI Cyber Defence Star’s components. These policies 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 Input

The SSPM connects to external services (Data Bus) through a KAFKA queue, by subscribing the specific topics exposed by RMS and AI Cyber Defence systems, using a consumer. If any data is present, the system collects and aggregates them, considering about the time of the specific events, into a single JSON object so it can be managed by OPA. Once the JSON Object is ready, OPA will evaluate the input together with the policies defined.

OPA proceeds through the Policy evaluation searching for the policy on the Database and passing it the JSON object it received from the aggregator component. Depending on the kind of policy evaluated other policies can be called or Data from the Database can be checked, in case of a poisoning attack for example. Once the policy has been fully evaluated a JSON object is created and published on the output, that can be a normal REST API or, if necessary, a KAFKA queue through its publisher.

The process for the definition of the active rules in SSPM is up to the Security Manager that will have available a GUI where it is possible to define, update and save them. It will communicate with OPA using standard CRUD operations; actually, will use only Read and Update operations because an empty rule will always be present to avoid the OPA start-up error. This user interface will be web based and hosted by the Olistic module where usually the Security Manager operates and it will consist in a simple form, empty at the beginning (if there are no rules defined, only the empty one), where it will be possible to write the rules following the REGO language that is the OPA’s native query language.

Rego was inspired by Data log, which is a well understood, decades old query language. Rego extends Data log to support structured document models such as JSON. Rego queries are assertions on data stored in OPA. These queries can be used to define policies that enumerate instances of data that violate the expected state of the system. The beauty of this language is that it is easy to read and write and this should simplify the definition and management work for the Security Manager.

OPA queries for both base and virtual documents via its API. Therefore, queries for just data return the combination of base and virtual documents located under that path.

Since base documents come from outside of OPA, their location under data is controlled by the Python application wrapping it. On the other hand, the location of virtual documents under data is controlled by policies themselves using the package directive in the language.

Documents are pushed or pulled into OPA synchronously when the SSPM queries for policy decisions, these documents are regarded as “Input” by OPA.

Policies can access these inputs under the input global variable. Built-in function return values can be assigned to local variables and surfaced in virtual documents. Data loaded synchronously is kept outside of data to avoid naming conflicts.

The following table summarizes the different models for loading base documents into OPA, how they can be referenced inside of policies, and the actual mechanism(s) for loading.

Model	How to access in Rego	How to integrate with OPA
Asynchronous Push	The <code>data</code> global variable	Invoke OPA's API(s), e.g., <code>PUT /v1/data</code>
Asynchronous Pull	The <code>data</code> global variable	Configure OPA's <code>Bundle</code> feature
Synchronous Push	The <code>input</code> global variable	Provide data in policy query, e.g., inside the body of <code>POST /v1/data</code>
Synchronous Pull	The <code>built-in functions</code> , e.g., <code>http.send</code>	N/A

Figure 30 Different models for loading base documents into OPA,

Data loaded asynchronously into OPA is cached in-memory so that it can be read efficiently during policy evaluation. Similarly, policies are also cached in-memory to ensure high-performance and high-availability. Data pulled synchronously can also be cached in-memory.

In the following image, we can see an example of a Policy Document structure.

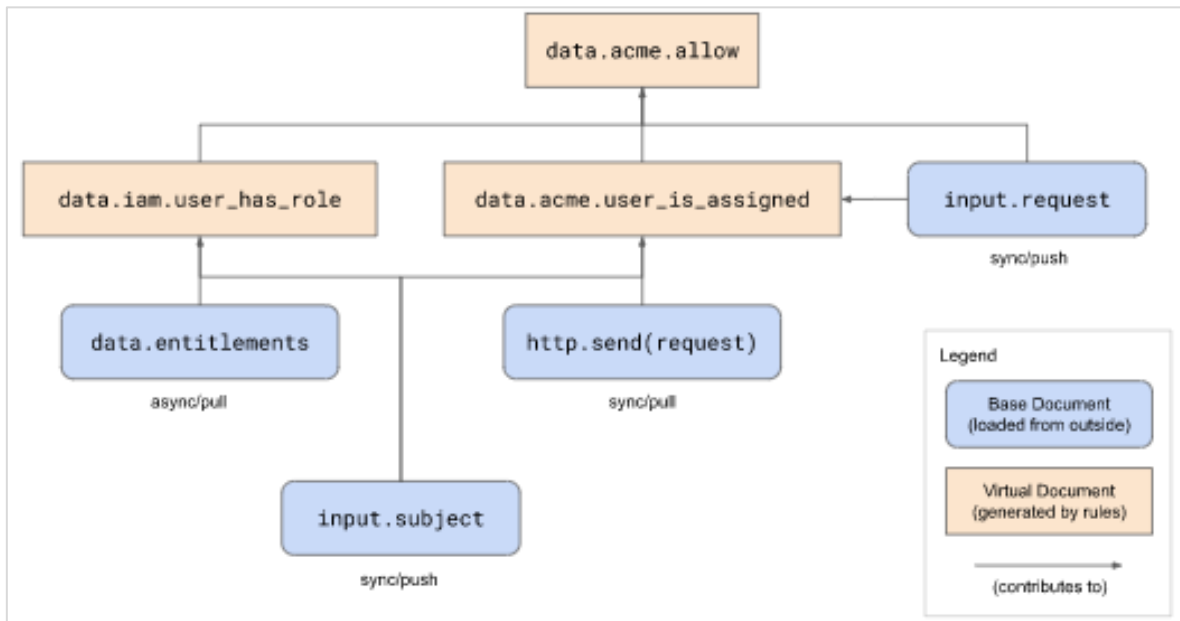


Figure 31 Policies model document (example)

Input: data received through Data Bus from RMS (mainly environmental data type, through probes and sensors) and from AI Cyber Defence Module (informative data type), both as JSON

format, which are then encapsulated in a single object provided to OPA, then evaluated with the policy, and at the end the result is provided to OLISTIC for the final risk evaluation process.

The detailed description of inputs and the related scripts are reported in Chapter 7 Integration and Validation chapter, and Appendix A.

5.3 Output

Output: a decision made by running the policies defined on the input data and then sent to Olistic as a JSON object.

Please refer to Chapter 7 Integration and Validation chapter, and Appendix A for the detailed description and scripts of the output.

5.4 Service distribution and configuration

The SSPM is deployed as a Docker image in the STAR infrastructure.

The Image consists of:

- The SSPM application implemented in python using python3;
- The KAFKA libraries for implementing the KAFKA queue for communicating with the external STAR components;
- The OPA service that is running in background for policy management and execution;
- A database containing all the required data;
- A database containing all the policies

The service can then be accessed externally for queries on the policies as an external service.

Also, the service provides a connection through KAFKA queues for external STAR components to communicate.

6 Relevant security policies assessment for Use Cases

6.1 Use cases analysis

Questionnaires (Annex B) were administered to the 3 pilots to understand the activities of the different UCs and determine which are their necessities in the field of security policies enforcement. Currently, there are not yet security policies enforced related to poisoning or evasion attacks built on purpose for the Use Cases. The aim of this task, the analysis of Use cases scenarios through the questionnaires is to assess the needed policies to protect the industrial processes during specific operations, involving AI systems in manufacturing.

The Questionnaires have been structured to explain the Use Cases scenarios and determine the assets involved. The assets are the potential targets of poisoning and evasion attacks and as so, they are the basis to understand the policies requirements.

The Questionnaires were structured also with the specific objective to establish a connection with the pilot owners' and build a common platform of discussion for the creation of policies, considering the day by day needs and concerns arising during the production.

6.1.1 1.Human Behaviour Prediction and Safe Zone Detection for Routing (DFKI)

This pilot will provide an automatic mobile robotic teaching solution based on a reinforcement learning approach for flexible and modular production lines. To achieve this a simulation of the production line and the mobile robot will be performed. The simulation will get dynamically updated when changes in the real environment occur. The simulated environment will enable the development of a Reinforcement Learning (RL) approach which will find the best (or even multiple good alternatives) solutions for the path control. The outcome of the simulation process will support the human operator. Furthermore, the pilot will define dynamic safety zones for mobile robots by using intention recognition algorithms to allow a close and safe collaboration between humans and mobile robots in a dynamic environment.

The pilot is composed by these UCs:

- UC1 Human intention recognition

First use case plans to detect the human activities and predict their next actions, which then will be combined with robot navigation to create a safer environment.

- UC2 Robot reconfiguration based on the dynamic layout.

The second use case is to dynamically update the navigation route of the mobile robot, by considering human and/or other (non-)moving objects in the environment. This use case will also enable easier reconfiguration of the robot in case the layout of the environment (including the production stations) changes. The layout is actively monitored by the cameras, and humans, as well as the objects in the layout, are detected. In case of any change, the new coordinates of the stations, where the robot should navigate to, are updated.

- UC3 Dynamic path planning using both first and second use cases.

As the third use case, these above mentioned two use cases are going to be combined to have a safe environment for the workers and the hardware equipment. The newly received

coordinates of the stations will be used to set the robot's destinations. The speed of the robot and the objects in the layout will also be considered to create a collision-free navigation path for the robot.

6.1.2 2.Human Centred AI for Agile Manufacturing 4.0 (IBER)

The IBER's final goal within the STAR project is to develop a solution for intelligent integration of processes and products to achieve solutions and systems that will allow the production of complex parts with the highest quality and minimal resources. In order to achieve that, IBER has defined four use cases listed above.

- UC1 Production Processes Simulations for Accelerated Decisions and Safe Processes
- UC2 Production Planning Optimization
- UC3 Employee Training for Reduction of Human Errors
- UC4 Agile Production Management System Data Integrity and Reliability

6.1.3 3.Pilot Human-Robot Collaboration for Quality Management (PHILIPS)

The objective of the pilot and related UCs is:

Enable lower volume production and decrease lead-times

Need for flexibility and reconfigurability of production assets

Need for increased re-use of manufacturing assets

Develop the manufacturing system of the future by integrating innovative technologies in existing production processes and by creating meaningful innovation while designing a new innovative production platform

The different UCs are:

- UC1 Easy reconfiguration for automated part handling

Employ AI for automated part detection, recognition, and localization for different parts, used the same hardware setup (Partnered with UPRC)

- UC2 Human supervised learning for visual quality inspections

Employ active learning for setting up automated quality inspection systems. (Partnered with JSI)

- UC3 Safe collaboration between human and robot

Employ the HDT / Fatigue monitoring tool on the active learning component from UC2 to recognize fatigue during labelling and be able to act upon it. (Partnered with JSI/SUPSI)

6.2 Use cases surveys results

As mentioned above, the scope of the surveys is to collect all the useful information about the UCs related to STAR Pilots, identifying the specific information to assess the needs in terms of Security Policies implementation.

Here below the answers received from the pilots for each UCs are summarized for a quick understanding and preliminary evaluations on policies needs, based on the declared assets and data flows identified.

6.2.1 Human Behaviour Prediction and Safe Zone Detection for Routing

Table 3 Summarized answers from the pilot Human Behaviour Prediction and Safe Zone Detection for Routing

Pilot	UC	Sources of cyber attacks	Input data		Sensors used	Data sets	Data collection	Security Policy manager	security policies
Human Behaviour Prediction and Safe Zone Detection for Routing	1	-	IMU sensor data+ camera video stream		IMU sensor data+ camera video stream	-	not continue	No connection from sensors to security systems in pilot	no
	2	computer running the AMR Fleet controller	Camera video stream		Camera video stream	A simulation is used for learning, not a data set	not continue		no
	3	-	MU sensor data + camera video stream + Camera video stream		MU sensor data+ camera video stream + Camera video stream	-	not continue		-

6.2.2 Human Centred AI for Agile Manufacturing 4.0

Table 4 Summarized answers from the pilot Human Centered AI for Agile Manufacturing 4.0

Pilot	UC	Sources of cyber attacks	Input data	Sensors used	Data sets	Data collection	Security Policy manager	security policies
Human Centred AI for Agile Manufacturing 4.0	1	Database; End-points; Com. devices; Users	TBD	Vision smart sensors, position sensors (hall effect), position transducers, photoelectric sensors, electromechanical sensors, inductive sensors and capacitive sensors	TBD	Stops during sensors stand-by	Rafael Almeida	ISMS that cover this pilot, based on best practices of ISO 27001 and NIST
	2							
	3							
	4							

6.2.3 Pilot Human-Robot Collaboration for Quality Management

Table 5 Summarized answers from pilot Human-Robot Collaboration for Quality Management

Pilot	UC	Sources of cyber attacks	Input data	Sensors used	Data sets	Data collection	Security Policy manager	security policies
Pilot Human-Robot Collaboration for Quality Management	1	TBD	9 CAD drawings supplemented by pictures of products	3D camera	CAD drawings	Stops during sensors stand-by	No insights	No
	2	Poisoned data Evasion attacks	dataset of labelled images	no sensor	datasets from IBER	Stops during sensors stand-by		Adversarial Robustness Toolbox vs poisoning and data poisoning. Evasion monitored by Fast Generalized Subset Scan. Poisoning prevented by spectral signatures or activation clustering.
	3	Human Digital Twin Core Infrastructure running on STAR servers	Dynamic (e.g., heart rate) and quasi-static (e.g., age) data of workers	Empatica E4, Polar H10	Experiment datasets containing anonymised personal data	Data collection is performed only during the experiments and requires that both sensor and		No

Pilot	UC	Sources of cyber attacks	Input data	Sensors used	Data sets	Data collection	Security Policy manager	security policies
						the sensors and gateways are manually activated		

6.3 Fine tuning of security policies

Security policies protect the organisation’s assets by identifying all possible threats and providing solutions to avoid data breach or leaks, economic loss and damage to physical infrastructure. Namely, STAR project security policies are meant to secure manufacturing processes against poisoning and evasion attacks, deriving from the AI systems implied.

6.3.1 Pilots’ assets

The first activity to allow a proper protection of the company’s assets is the identification of the assets themselves.

Given the assets reported in the tables below by the responsible use case partner, we proceed with the instantiation of this environment in the OLISTIC. The following asset cartographies are given in each pilot respectively. Assets have been generated as the digital reflection of the manufacturing floors and will enable the risk management operations of OLISTIC. This asset cartography is expected to be updated, as the STAR demonstrators become more mature, and more assets can be engaged in the manufacturing processes of the pilots.

The following tables have been obtained from the surveys and were used to generate the respective asset cartographies.

- Pilot: Human Behaviour Prediction and Safe Zone Detection for Routing

Table 6 Human Behavior Prediction and Safe Zone Detection for Routing pilot’s assets

Asset_ID	Name	Short description	Details	Asset Category
<u>1</u>	<u>ROS</u>	Robot OS		
<u>2</u>	Ubuntu	OS		
<u>3</u>	Camera	Video streaming from testbed to THALESgroup.		
<u>4</u>	IMU Sensor	Shimmer3		
<u>5</u>	Smart Watch	Apple Watch		
<u>6</u>	Smart Phone	iPhone 12 Mini		
<u>7</u>	Smart Object detection & localisation (VCA modules)			
<u>8</u>	Smart Human detection &			

Asset_ID	Name	Short description	Details	Asset Category
	localisation (VCA modules)			
9	AMR Fleet Controller	AI (RL) controller that send commands to AMRs depending on the current need, and current AMRs status, and current factory workers occupancy	Python code relying on PyTorch, receiving information indirectly about workers through a heatmap produced by video analytics (Safety zone detection)	Software

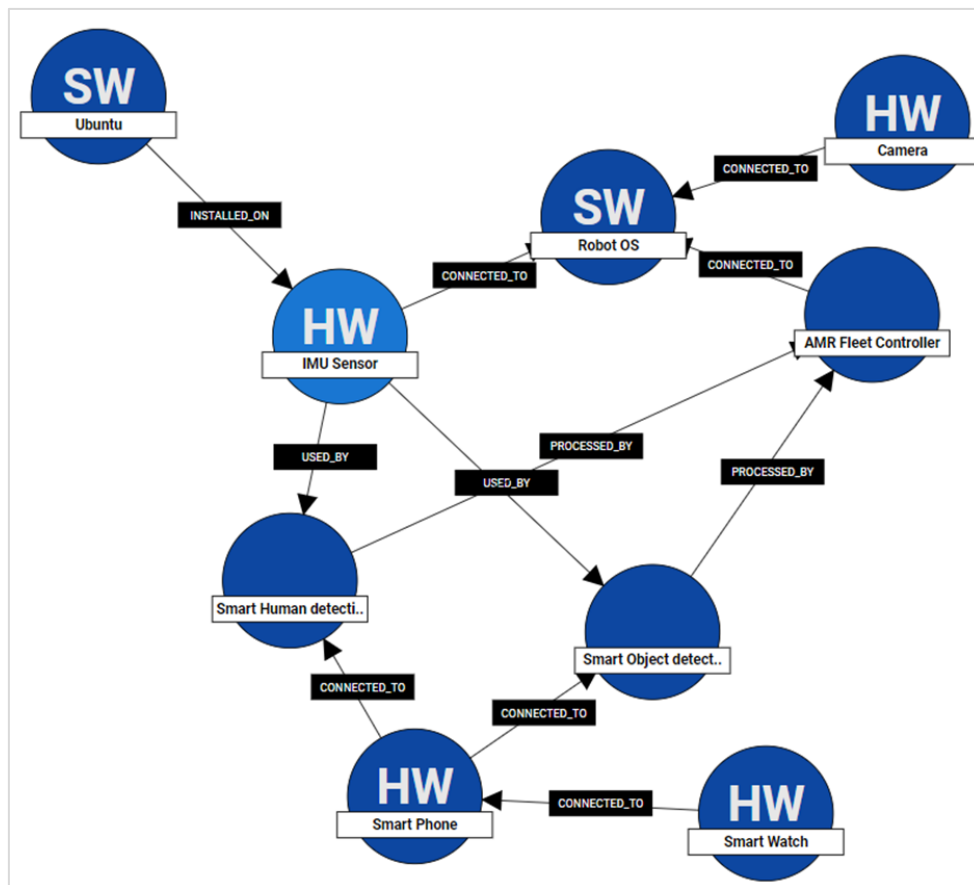


Figure 32 OLISTIC Asset Cartography for Human Behavior Prediction and Safe Zone Detection for Routing

- Human Centred AI for Agile Manufacturing 4.0

Table 7 Human Centred AI for Agile Manufacturing 4.0 pilot's assets

Asset_ID	Name	Short description	Details	Asset Category
<u>1</u>	PLC (Linux)	PLC that controls the functions of an equipment	Linux distribution	Proprietary Hardware
<u>2</u>	PLC (Windows)	PLC that controls the functions of an equipment	Windows CE	Proprietary Hardware
<u>3</u>	Smart camera	Firmware to control the camera		Proprietary Software and Hardware
<u>4</u>	Printer	Product labelling system	Connected in the network	Proprietary Software and Hardware
<u>5</u>	Database	Storage of quality inspection images	Local storage on server (MS SQL)	
<u>6</u>	Server	Sever for remote access	-	
7	HMI interface	Display on equipment for human-machine interface	Windows CE	Proprietary Software and Hardware
8	Switch	Communication device	Component where the device connects	Network communication
9	MES Software	Views from Manufacturing Execution System, real time production information Database	Views from SQL database	Database
10	ERP Software	Views from ERP, Management, logistic and production Information	Views from SQL Database	Database
11	BDE	Human Interface terminal in work center	Proprietary Hardware	End- Point
12	Terminal PC	Human Interface terminal in work center	Computer	End-Point
13	Barcode Reader	Barcode Reader with a screen to human interface	Proprietary Hardware	End-Point
14	Wirelessnetwork	To connect barcode Readers	Component where the device connects	Network communication

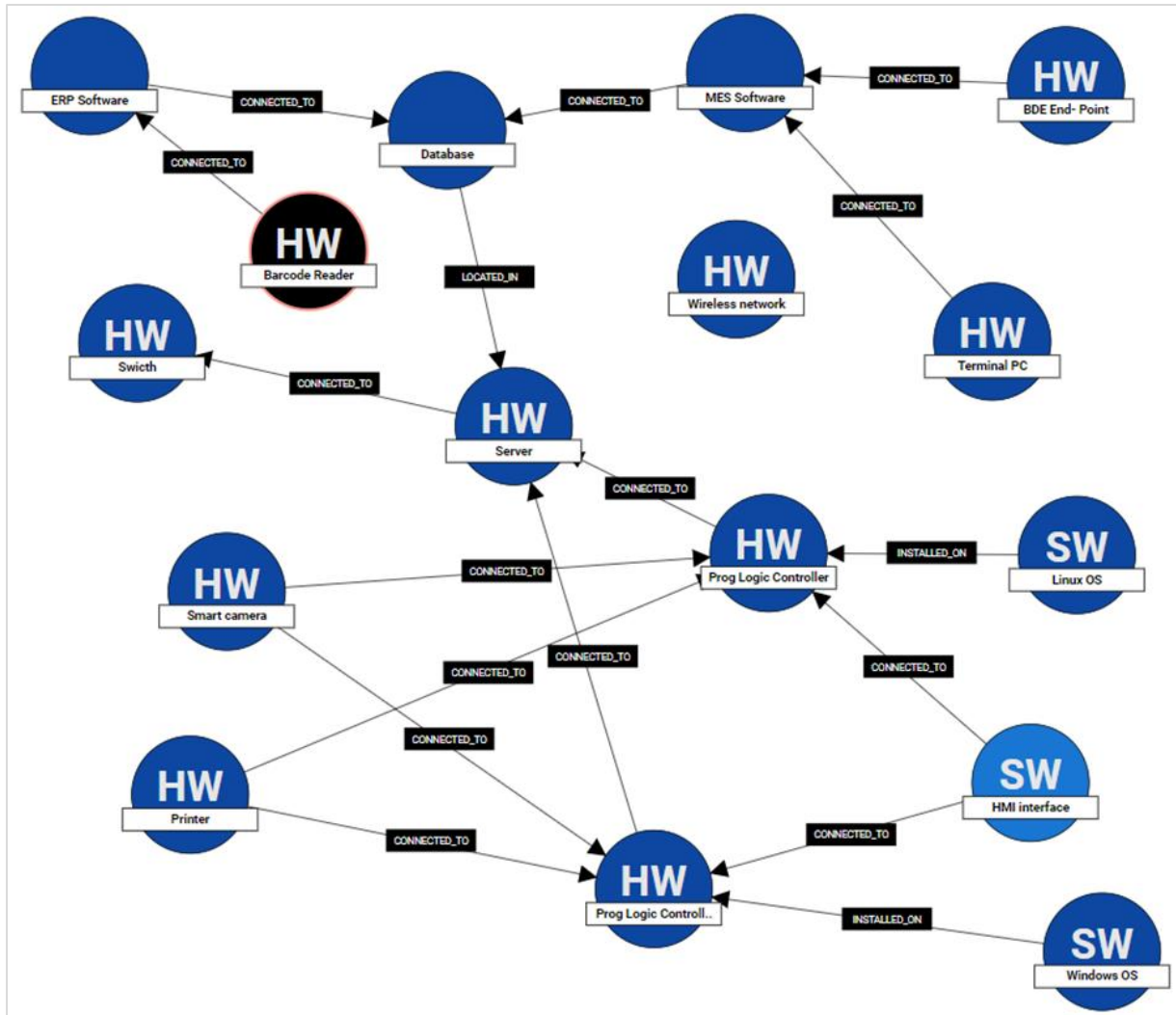


Figure 33 OLISTIC Cartography Human Centred AI for Agile Manufacturing 4.0

- Human-Robot Collaboration for Quality Management

Table 8 Human-Robot Collaboration for Quality Management pilot's assets

Asset_ID	Name	Short description	Details	Asset Category
<u>1</u>	Industrial PC	A computer that provides a user interface & runs the required software		Proprietary Hardware
<u>2</u>	Industrial PC OS	Operating system installed on computer	Windows Enterprise N 10	Proprietary Software
<u>3</u>	AI Algorithm for quality control	AI algorithm in python	Python 3.7 and libraries to run AI algorithm	
<u>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	Proprietary Hardware

Asset_ID	Name	Short description	Details	Asset Category
			Telecentric Lens TC12192	
5	Firmware camera	Firmware to control the camera	Pylon provided by Basler	Proprietary Software
6	Programmable LED controller + lighting	LED controller that controls the lighting needed for the quality inspection image	CCS PD3-5024-4-EI	Proprietary Hardware
7	LED controller protocol/software	The LED controller can receive encoded bytes	Python socket module TCP/IPv4 protocol UTF-8	
8	Printer PLC	PLC that provides a trigger to the camera to inspect image	Connected with simple on/off signal	Proprietary Hardware
9	Database	Storage of quality inspection images	Local storage on computer	
10	External server	For remote access	-	
11	Other computer software	To run the system	Pypylon by Basler, Python	

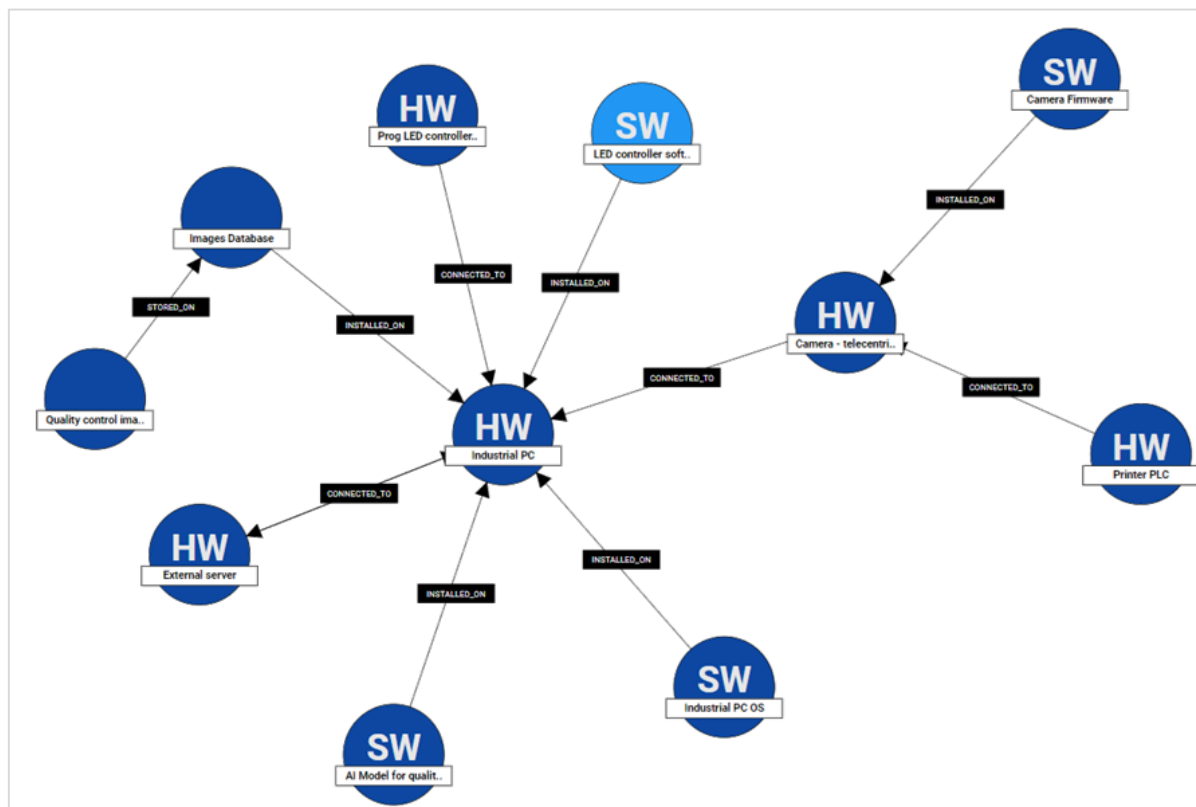


Figure 34 OLISTIC Cartography Human-Robot Collaboration for Quality Management

The assets are both physical and digital, a condition which adds complexity and require a holistic approach when establishing security policies. Even though not every component produces the data with the same relevance, each part of the infrastructure can become a hole in which attacks can find a way into way more important components of the network.

However, the protection of every single part of the infrastructure is complex and requires a lot of work. To be effective, the first activity should be the prioritization of assets and risks by criticality.

6.3.2 Security policies

Security and Data Governance for AI Systems integrates Cyber-defence and data reliability techniques within existing platforms of the partners for IoT security. The platform once deployed will facilitate the implementation of security and data protection policies for industrial data and AI algorithms, leveraging on Star’s cyber-security results. However, at this stage of the Security and Data Governance platform deployment, the draft of the security policies remains generic, allowing flexibility to any required policy, to be implemented, once the use cases layout will be validated.

As an example of the type of assets, threats and controls object of policies, we hereby (see table 9) report the European Agency for Network and Information Security ways to attack, and therefore protect, the company from people’s wrong use of data and facilities:

Table 9 Perspectives on transforming cybersecurity, McKinsey and Company, 2019

Assets	Threats	Controls
Data	Data breach Misuse or manipulation of information Corruption of data	Data protection (e.g., encryption) Data-recovery capability Boundary defense
People	Identity theft “Man in the middle” Social engineering Abuse of authorization	Controlled access Account monitoring Security skills and training Background screening Awareness and social control
Infrastructure	Denial of service Manipulation of hardware Botnets Network intrusion, malware	Control of privileged access Monitoring of audit logs Malware defenses Network controls (configuration, ports) Inventory Secure configuration Continuous vulnerability assessment
Applications	Manipulation of software Unauthorized installation of software Misuse of information systems Denial of service	Email, web-browser protections Application-software security Inventory Secure configuration Continuous vulnerability assessment

Starting from the above threats it is possible to delineate several security policies for Star Use Cases. However, as already mentioned, security policies are strongly related to the single case, thus it is necessary to receive further information from pilots regarding outputs on threats, likely to occur in the different UCs scenarios. The further development of use cases preparation will display all information required for the deployment of specific policies.

6.4 Discussion on surveys results

As stated before, and as emerged during the pilots' surveys, there are no security policies already established to govern security of the pilots in terms of protection against poisoning and evasion attacks.

This is probably related to the fact pilots are still under development, thus other sections, more technical, are under attention. This also means that security policies, at this stage of development, are difficult to be defined because not every technology is implemented yet. Security policies are selected and implemented according to the usage of the specific technology and the kind of data produced.

However, in the next months it will be possible, even before the final development of the technical part of the pilots, to suggest and implement a first stage of security policies related to the interaction with STAR innovative technologies (detection of abnormal behaviours of manufacturing production lines, existing vulnerabilities, AI-oriented attacks). Technological security issues will require a specific work with each pilot to define the best security policies.

In fact, security policies are extremely specific to the kind of data produced, the assets and the user involved. A deep interaction with pilots will allow the definition of specific security policies.

In general, STAR security policy activities will be helpful in focusing the issues at the pilots' level and in developing tailored solutions according to the technologies involved.

Summarizing up, according to the use cases preliminary surveys, the policies 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.

Based on the type of assets surveyed in the pilots and involved in the different use cases, it is possible to confirm that the type of features involved in policies would be among the followings:

- Time series on CPU;
- Time series on GPU;
- RAM usage;
- Network card throughput (number of packets exchanged);
- Disk input-output.

The rules to be assigned for the evaluation of the policies during the running of the use cases will be defined by monitoring the ranges of values recorded by the RMS and then passed to the SSPM for the detection of abnormalities.

Here below (Figure 35) is a possible representation of a policy layout as it will appear to the Policy Manager once defined:

Rule
Disable Anonymous Authentication to the Kubelet [ref]

By default, anonymous access to the Kubelet server is enabled. This configuration check ensures that anonymous requests to the Kubelet server are disabled. Edit the Kubelet server configuration file `/etc/kubernetes/kubelet/kubelet-config.json` on the kubelet node(s) and set the below parameter:

```
authentication:
  ...
  anonymous:
    enabled: false
  ...
```

Rationale:	When enabled, requests that are not rejected by other configured authentication methods are treated as anonymous requests. These requests are then served by the Kubelet server. OpenShift Operators should rely on authentication to authorize access and disallow anonymous requests.
Severity:	medium
Rule ID:	xccdf_org.ssgproject.content_rule_kubelet_anonymous_auth
Identifiers and References	References: CIP-003-8 R6 , CIP-004-6 R3 , CIP-007-3 R6.1 , CM-6 , CM-6(1) , SRG-APP-000516-CTR-001325 , SRG-APP-000516-CTR-001330 , SRG-APP-000516-CTR-001335 , 3.2.1

Figure 35 Example of policy representation

The final layout will be implemented in the following phases of the project and is the object of the final version of the deliverable.

7 PoC Integrated Architecture & Validation Scenario

7.1 Integrated Architecture

In this paragraph, the Security and Data Governance for AI Systems integrated architecture is displayed. The architecture will boost data protection and reliability against poisoning and evasion attacks.

Figure 36 shows the interactions of the Security and Data Governance different modules with of Star Security Policy Manager, (SSPM), the module for security and data governance of AI system, which integrates the cyber-defence mechanism of the task (3.3), strictly related to task (3.4) activities, described in the present report.

The Star Security Policy Manager receives inputs from the Runtime Monitoring System and XAI cyber defence module and can validate the data from data provenance and traceability components (red path).

The risk assessment functionalities based on Olistic, Risk Assessment Engine and the interaction with the SSPM are also represented in Figure 36 (green path). Olistic gives input to the tool and communicates the existence of a threat to the SSPM.

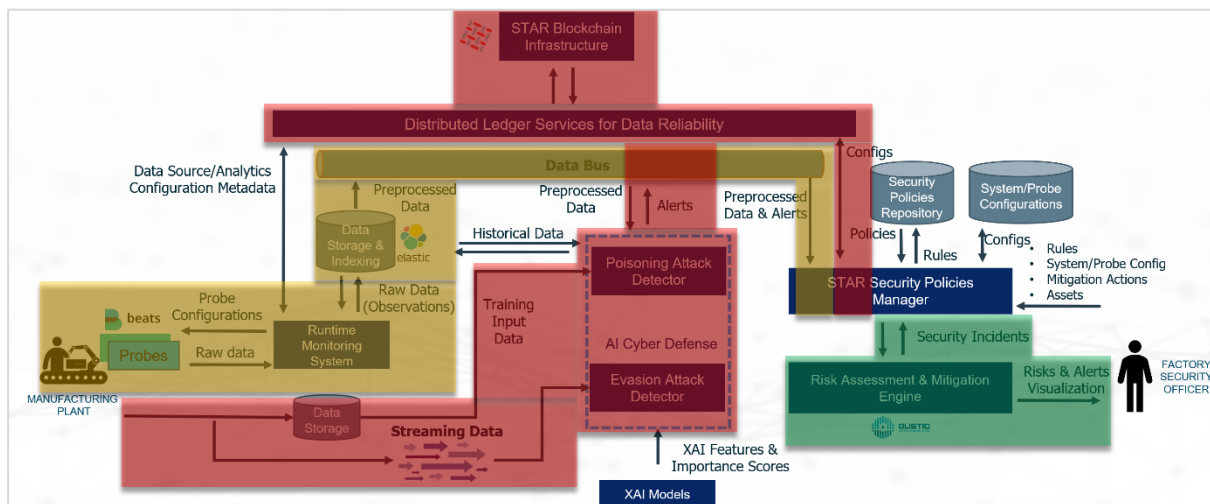


Figure 36 STAR Security and Data Governance integrated Architecture

The Security Policy Manager interacts with the other components by using a KAFKA Queue.

KAFKA Queue is a queueing system that offers low-latency message processing, high availability, and fault tolerance. It operates through a publish-subscribe pattern.

The security policy manager implements such a system. In input, it creates a KAFKA client that subscribes to a topic and receives data every time the topic is updated, and the output creates a KAFKA publisher that creates topics for external systems to subscribe to and sends data through it. Such data would be the policy manager's decisions after the policies have been evaluated with the received input.

7.2 Common Data Models

In this section, we introduce the core data models used for data exchange and the business context for the Security & Data Governance solution. These Data models are extended & adapted from other EU projects and more specifically H2020-SecureIoT (ID: 779899).

7.2.1 Observations

As described in D3.1 and D3.3 the Observation entity is used to format measurements and results coming from various data sources. For the security and data governance solution it is used to exchange messages/reports through the various components (e.g., RMS, AI Cyber Defence & Security Policy Manager described in previous chapters). It is also used for persisting results to the Data Reliability component to be used for validation purposes. An Observation is associated with a timestamp and keeps track of the location of the data source in case it is associated with a mobile (rather than a stationary) data source. Hence, it has a location attribute as well.

As shown in Figure 37 below the “Observation” has the following parameters:

- id: A unique required ID which is assigned to every observation when captured from the STAR system.
- dataSourceID: The ID of the Data Source Manifest (physical or virtual) these observations refer to.
- systemID: a unique identifier of the monitored IoT system
- reportType: which provides information about the Observation produced report type.
- timestamp: The timestamp indicating the instance in which a measurement was acquired by the STAR system.
- Location: which provides the geographical or virtual location an incident took place. The “Location” has:
 - geolocation: which provides the coordinates (longitude and latitude) of a physical location.
 - virtualLocation: which provides information about a virtual location (it could be the ID of a resource or subsystem).
- value: which provides the value of the measurement. The value can be of simple (e.g., a float number depicting temperature) or complex (e.g., measurements from a weather station, which are consisted of multiple measurements or a threat report) structure. The type and structure of the value is described in the reportType entity.

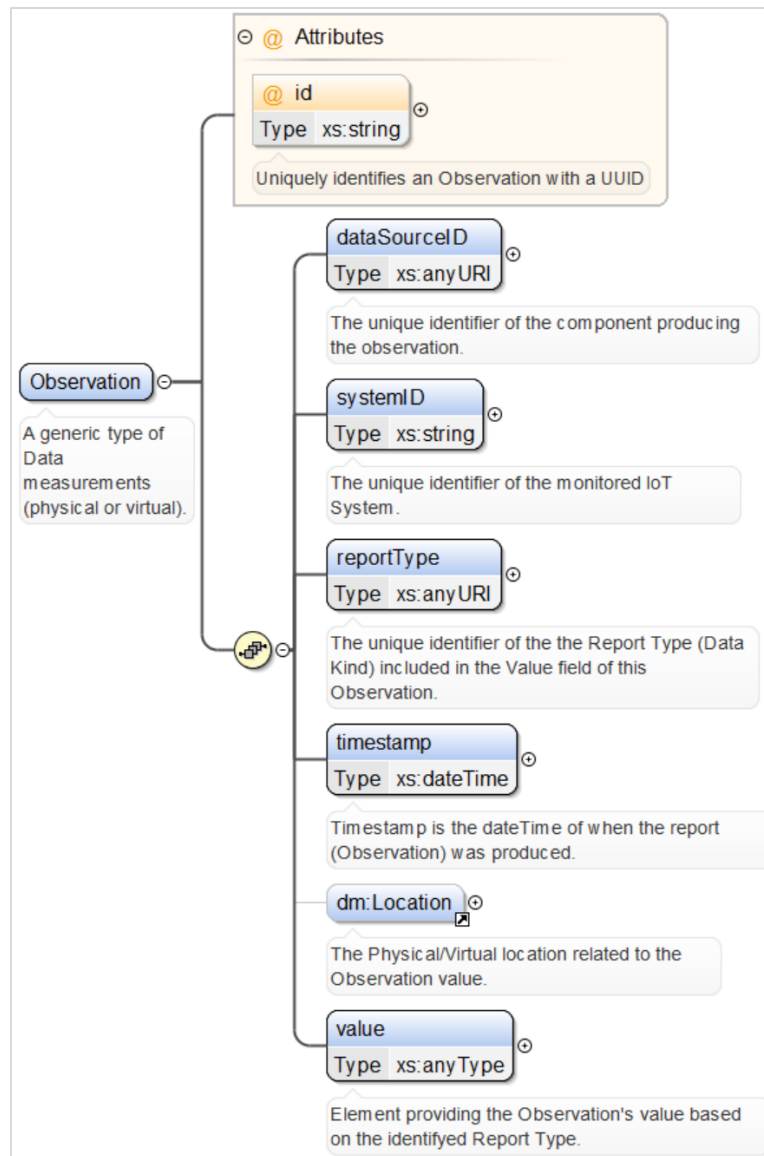


Figure 37 Observation Entity structure

7.2.2 Security-focused Configuration Management (SecCM) Data Model

The SecCM focuses on the configuration aspects of the data-driven security infrastructure. It comprises information about the monitored IoT systems and the attack scenarios against them. As shown in Figure 38 below the SecCM group consists of the following core entities:

- **System:** Provides a set of observed characteristics for the IoT System.
- **Asset Category:** Provided the ability to group Assets in categories.
- **Asset:** Models a cyber, physical, or cyber-physical element within an organization that is used in the frame of business operations. Figure 39 below illustrates the entity structure.
- **Vendor:** Specifies an Asset’s Vendor.
- **Control:** Models a control element that is used to prevent attack impact.
- **Mitigation Plan:** Provides a Mitigation Plan based on specified asset.
- **Mitigation Measures:** Provides a list of available Mitigation Measures to be applied for a given Abuse Case.

- **Vulnerabilities:** It is used to extend the MITRE Common Vulnerabilities and Exposures (CVE) collection.
- **Attack Scenarios:** Models a set of Abuse Cases based on the monitored Asset and specified scenarios. Figure 40 below illustrates the entity structure.

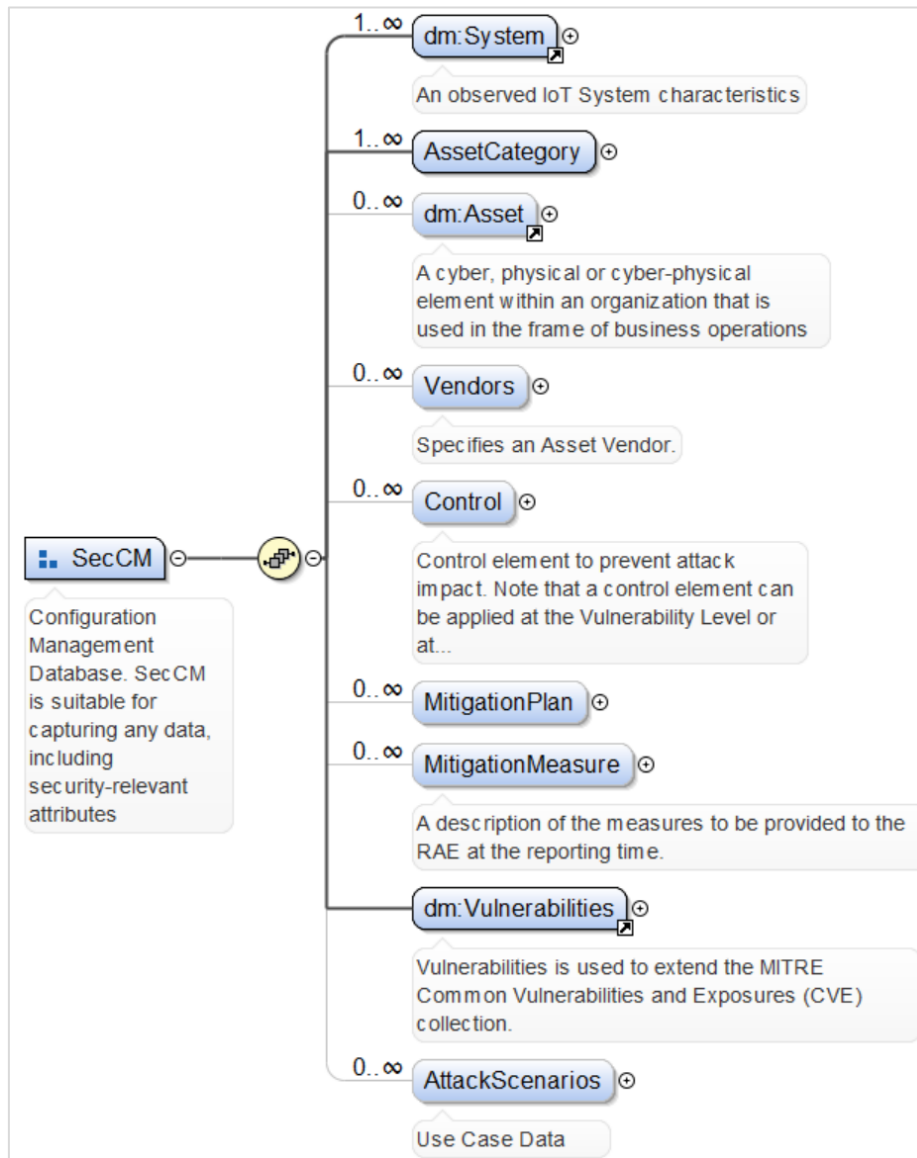


Figure 38 SecCM entity structure

One of the core entities used to describe the monitored system is the Asset depicted in Figure 39 below. An asset is used to describe the physical or virtual monitored devices and the relationships between them which is used for the graph representation in OLISTIC.

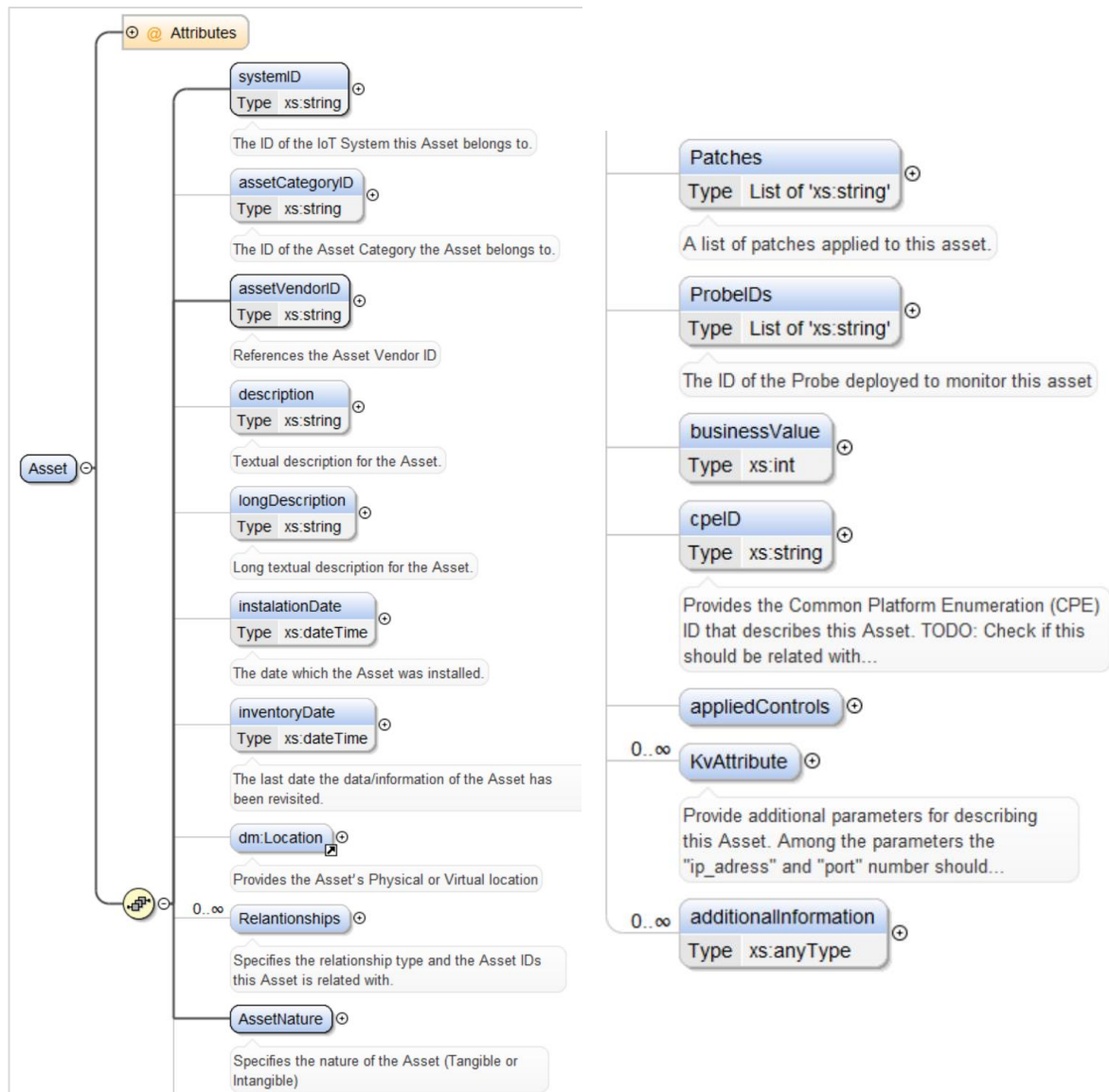


Figure 39 Asset entity structure

Figure 40 below depicts the structure of the Attack Scenarios entity. This entity models a set of Abuse Cases based on the monitored Asset and specified scenarios including information on the Asset applicability, vulnerabilities, and possible attacks.

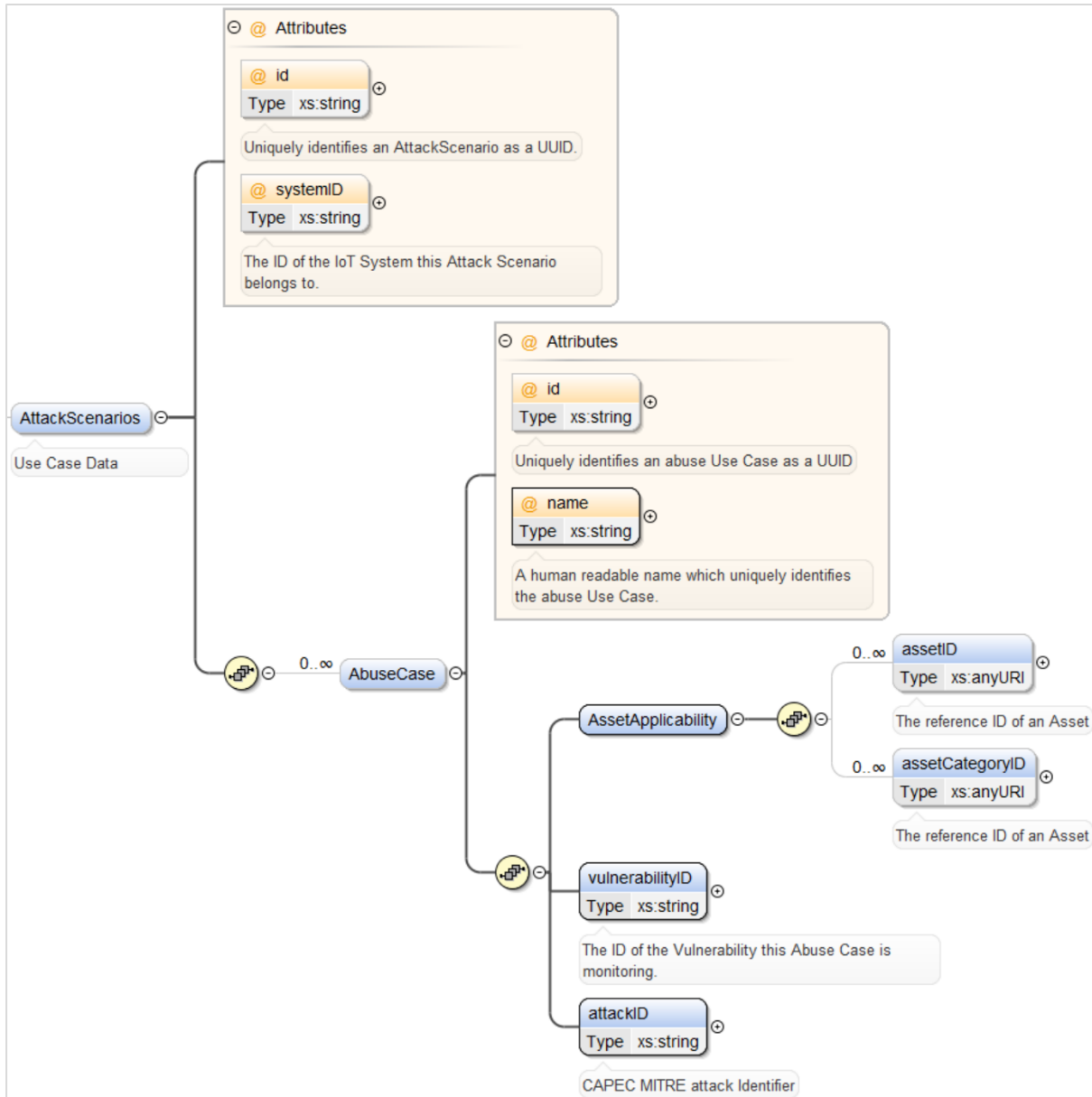


Figure 40 Attack Scenarios entity structure

7.3 Integration & Validation

7.3.1 Validation Scenario and Components Integration

For the initial validation of the Security & Data Governance solution we have identified a simple scenario for demonstrating the end-to-end usage of the system. An AI-assisted visual quality inspection is undertaken by a fixed system on PCL’s manufacturing floor. In this setup, an abnormal increment of CPU utilization is recorded, while the AI-cyber defence module reports the detection of adversarial samples. The policy manager detects a policy violation and threats are highlighted in OLISTIC.

Table 10 provides an overview of the system integration, for the integration and validation scenario deployment.

Table 10 Component integration dependencies and flows

Component Name	Dependencies with other components	Description	Inputs from other STAR components	Outputs to other STAR components	Flow description
Distributed Ledger Services for Data Reliability	<ul style="list-style-type: none"> AI Cyber Defence RMS Security Policy Manager 	Infrastructure for decentralized data reliability	<ul style="list-style-type: none"> AI Cyber Defence (Observations see Chapter 3) AI Cyber Defence (Processors) RMS (Observations see Chapter 3) 	<ul style="list-style-type: none"> AI Cyber Defence (Observations) see Chapter 3) AI Cyber Defence (Processors) Security Policy Manager (Observations)see Chapter 3) 	The storage and retrieval of a PoC metadata entity (e.g., Processor Definition).
AI Cyber Defence Module		Detection of poisoning and evasion attacks	Dataset from pilot sites. Images from PCL production lines. If data are not available, publicly available datasets will be used.	Classification models with increased resilience against the offensive strategies.	Generation of adversarial examples and the deployment of defenses against them
Runtime Monitoring System	<ul style="list-style-type: none"> AI Cyber Defence DLSDR Security Policy Manager 	Collection of security related data to drive analytics algorithms that detect patterns of abnormal behaviour.	Dataset and system measurements from monitored IoT system (e.g., pilot sites).	<ul style="list-style-type: none"> AI Cyber Defence (Observations) see chapter 3) Security Policy Manager (Observations) see chapter 3) DLSDR (Observations see chapter 3) 	Collect system and scenario data (e.g., CPU/RAM utilization) from a monitored system, filter them (based on configurations), transforms them to observations, push them to Elastic Stack and Data Bus for further usage by other components.
Security Policy Manager		Tool to be used by the personnel of the factory, in particular security/IT officers, to configure security policies	Rules, Probes data, AI Cyber Defence output	Security Policies, evaluated output	Receive events from the RMS and the AI Cyber defence module, fire the policies on them (evaluation) and communicate the output to Olistic.
Olistic		Risk assessment visualization considering potential policy violations	Reference Appendix A1.1	Reference Appendix A1.1	

8 Conclusions

This Deliverable summarizes the advancements made by task 3.5 partners in the development of the Security and Data Governance Infrastructure Initial Version so far in the project stage. In the next reporting period, partners will carry on activities mainly related to the definition and implementation of information coming from Use Cases to adapt the general findings into real industrial environments and thus deploy the final version of the Security and Data Governance Infrastructure.

This deliverable has introduced the first version of the Security and Data Governance architecture, including the main modules that comprise it, namely the Runtime Monitoring System, the AI Cyber Defence Module and the Security Policy Manager, and the structuring principles that can enable their integration, as well as the main information, flows between them. In this version of the deliverable, the logical and process views have been prioritized, while physical deployment and implementation considerations have been briefly discussed. The resulting Security and Data Governance Infrastructure provides a high-level description of the structure and the proof of concept of a validation scenario: AI-assisted visual quality inspection undertaken by a fixed system on PCL's manufacturing floor. However, the deployment of the Security and Data Governance initial version demonstrates a high-level of flexibility, capable to comply, and be in-line with the business requirements that it targets.

The Security and Data Governance architecture initial version description mainly focuses and elaborates on the logical and process views detailing the interactions between the main building blocks and the modules involved. Furthermore, the present deliverable utilizes the initial views and descriptions of the use cases / pilots and select a validation scenario to validate the current design of the Security and Data Governance architecture initial version architecture. In the next stages, the detailed designs of all modules, their implementation, integration and evaluation through all the use cases / pilots, will be described in the follow-up version of this report (D3.5 final version), which will further elaborate on the physical and implementation views. Currently, the involved STAR partners undertook work towards the above-listed directions, including the implementation of the architecture and its deployment in all STAR use cases, to delineate specific security policies and related rules. In D3.5 final version, an updated and more complete version of the architecture will be presented. Furthermore, the next version of the deliverable will update the modules and the structuring principles of the architecture based on feedback received during the actual implementation of STAR project use cases.

References

Page	Link to document
27	https://www.elastic.co/elastic-stack/
28	https://www.elastic.co/guide/en/beats/libbeat/current/index.html
28	https://github.com/elastic/beats
28	https://www.elastic.co/guide/en/beats/libbeat/master/community-beats.html
28	https://github.com/elastic/beats/tree/master/filebeat
28	https://github.com/elastic/beats/tree/master/metricbeat
28	https://github.com/elastic/beats/tree/master/packetbeat
29	https://www.elastic.co/kibana/
29	https://www.elastic.co/elasticsearch/
38	https://www.first.org/cvss/
55	Perspectives on transforming cybersecurity, McKinsey and Company, 2019

Appendix A Component Scripts

A.1 Component Inputs & Outputs

```
{
  "id": "5ea6b598-632e-11ec-90d6-0242ac120003",
  "dataSourceID": "f59c4669-d3a6-40a0-9269-83619e5eb915",
  "assetID": "4daef4f2-487e-48e1-8f8f-d526d36aa5cd",
  "dataKindID": "59912e6d-7ee2-4fc4-bba7-0c430d7f39ce",
  "timestamp": "2022-01-10 13:00:37.861639",
  "location": {
    "geoLocation": {
      "latitude": "53.107731",
      "longitude": "6.088499"
    },
    "virtualLocation": "8.162.203.200"
  },
  "value": {
    "cpu": "90",
    "timestamp": "2022-01-10 13:00:19.709098"
  }
}
```

Figure 41 RMS-Monitored System Observation data (Probe Monitoring Data)

```
{
  "id": "1ee5f356-632e-11ec-90d6-0242ac120003",
  "dataSourceID": "3341e5b2-632e-11ec-90d6-0242ac120003",
  "assetID": "4daef4f2-487e-48e1-8f8f-d526d36aa5cd",
  "dataKindID": "65a7604e-9a94-4a74-9a34-3e44c6cebd49",
  "timestamp": "2022-01-10 13:01:29.709071",
  "location": {
    "geoLocation": {
      "latitude": "53.107731",
      "longitude": "6.088499"
    },
    "virtualLocation": "8.162.203.200"
  },
  "value": {
    "attackID": "f777d14b34c3cdff92468fbfa55aeddd8298745",
    "attackContext": "Evasion attack",
    "Confidence": "90.12",
    "timestamp": "2022-01-10 13:01:22.709095"
  }
}
```

Figure 42 AI Cyber Defence Observation

```
[{
  "abuseCaseID": "0140d7e2-7108-4985-8410-0501bc1a3c1d",
  "assetApplicability": {
    "assetCategories": [
      "a8db5ade-77c7-460c-95a7-688feb199080"
    ],
    "assets": [
      "4daef4f2-487e-48e1-8f8f-d526d36aa5cd"
    ]
  },
  "attackID": "AML.T0015",
}
```

```

    "cveID": "string",
    "lastModifiedDate": "2021-12-22T09:32:54.158Z",
    "name": "Evade ML Model"
  }
]

```

Figure 43 Security Policies Manager Configurations

```

[ {
  "abuseCases": [ {
    "abuseCaseID": "0140d7e2-7108-4985-8410-0501bc1a3c1d",
    "assetApplicability": {
      "assetCategories": [
        "a8db5ade-77c7-460c-95a7-688feb199080"
      ],
      "assets": [
        "4daef4f2-487e-48e1-8f8f-d526d36aa5cd"
      ]
    },
    "attackID": "AML.T0015",
    "lastModifiedDate": "2021-12-22T09:32:54.158Z",
    "name": "Evade ML Model"
  }
],
  "attackScenarioID": "8d11255fb6bce45ab086ecca05536aff53bfb9c",
  "lastModifiedDate": "2021-12-22T09:32:54.158Z",
  "systemID": "4b20cf4c-8321-422b-b0a4-6c9b1d02dd7c"
}
]

```

Figure 44 Security Policies Manager Results

Appendix B Questionnaires

B.1 Security Policies needs assessment - Human Behavior Prediction and Safe Zone Detection for Routing

List of assets of the Use Cases

Table 11 List of Assets

Asset ID	Name	Short description	Details	Asset Category
1	ROS	Robot OS		
2	Ubuntu	OS		
3	Camera	Video streaming from testbed to THALESgroup.		
4	IMU Sensor	Shimmer3		
5	Smart Watch	Apple Watch		
6	Smart Phone	iPhone 12 Mini		
7	Smart Object detection & localisation (VCA modules)			
8	Smart Human detection & localisation (VCA modules)			
9	AMR Fleet Controller	AI (RL) controller that send commands to AMRs depending on the current need, and current AMRs status, and current factory workers occupancy	Python code relying on PyTorch, receiving information indirectly about workers through a heatmap produced by video analytics (Safety zone detection)	Software

Asset interdependencies and visualisation

The risk assessment engine can 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.

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

Table 12 List of Assets interdependencies

Asset_ID_X	Relation	Asset_ID_Y	Details
8	Is_Processed_by	9	through heatmap data
9	Is_connected_to	1	to get current position, and send to commands as waypoints
5	Is_connected_to	6	To send the sensor data to iPhone Mini

Which are the potential sources of cyber-attacks in your pilot lines?

- UC1 Mobile Robot Simulation: The main processing server to run worker’s activity recognition module connecting to IMU sensors must be protected.
- UC2 Reinforcement Learning for Path Control: The computer running the AMR Fleet controller must be protected. Otherwise, somebody may take control of the AMR fleet.
- UC3 Safety Zones Definition

Which are the input data utilised for running the UCs?

- UC1 Mobile Robot Simulation: IMU sensor data+ camera video stream
- UC2 Reinforcement Learning for Path Control: Camera video stream
- UC3 Safety Zones Definition: IMU sensor data+ camera video stream + Camera video stream

Which are the sensors used to collect data for the UCs set-up?

- UC1 Mobile Robot Simulation: IMU sensor data + camera video stream
- UC2 Reinforcement Learning for Path Control: Camera video stream
- UC3 Safety Zones Definition: IMU sensor data+ camera video stream + Camera video stream

Which are the data sets available and related repositories?

- UC1 Mobile Robot Simulation: we do not plan to use public datasets but use the collected activity recognition dataset by the IMU sensors.

- UC2 Reinforcement Learning for Path Control: A simulation is used for learning, not a data set.
- UC3 Safety Zones Definition

Is the data collection continue or does it stop during the sensors stand-by?

- UC1 Mobile Robot Simulation: Not continue
- UC2 Reinforcement Learning for Path Control: Not continue
- UC3 Safety Zones Definition: Not continue

Is there a Security Policy manager overlooking the Pilot?

The main aspect of the security policy in the Smart factory DFKI is the streaming of the video from the cameras installed on the ceiling must be informed to people working on the testbeds. For this reason, we could not guarantee continued (2.7) video streaming and it should be managed by one of the STAR project's colleagues in Smart factory. The Video streaming will last for 30 minutes if there is no person available on the streaming modules.

Moreover, we do not have any connection from our sensors to any existing security systems in the pilot environment.

B.1.1. Do you have already figured out a list of security policies and related rules?

- UC1 Mobile Robot Simulation: No
- UC2 Reinforcement Learning for Path Control: No.
- UC3 Safety Zones Definition

B.2 Security Policies needs assessment - Human Centred AI for Agile Manufacturing 4.0

List of assets of the Use Cases

Table 13 List of Assets

Asset_ID	Name	Short description	Details	Asset Category
1	PLC (Linux)	PLC that controls the functions of an equipment	Linux distribution	Proprietary Hardware
2	PLC (Windows)	PLC that controls the functions of an equipment	Windows CE	Proprietary Hardware
3	Smart camera	Firmware to control the camera		Proprietary Software and Hardware
4	Printer	Product labelling system	Connected in the network	Proprietary Software and Hardware
5	Database	Storage of quality inspection images	Local storage on server (MS SQL)	
6	Server	Sever for remote access	-	
7	HMI interface	Display on equipment for human-machine interface	Windows CE	Proprietary Software and Hardware
8	Switch	Communication device	Component where the device connects	Network communication
9	MES Software	Views from Manufacturing Execution System, real time production information Database	Views from SQL Database	Database
10	ERP Software	Views from ERP, Management, logistic and production Information	Views from SQL Database	Database
11	BDE	Human Interface terminal in work center	Proprietary Hardware	End-Point
12	Terminal PC	Human Interface terminal in work center	Computer	End-Point
13	Barcode Reader	Barcode Reader with a screen to human interface	Proprietary Hardware	End-Point
14	Wireless network	To connect barcode Readers	Component where the device connects	Network communication

Asset interdependencies and visualisation

The risk assessment engine can 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.

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

Table 14 List of Assets interdependencies

Asset_ID_X	Relation	Asset_ID_Y	Details
PLC (Linux) (1)	Is_connected_to	Server (6)	Connected via Ethernet
PLC (Windows) (2)	Is_connected_to	Server (6)	Connected via Ethernet
Database (5)	Is_installed_on	Server (6)	
Smart camera (3)	Is_connected_to	PLC (Linux) (1) and PLC (Windows) (2)	Connected via Ethernet
Printer (4)	Is_connected_to	PLC (Linux) (1) and PLC (Windows) (2)	Connected via Ethernet
HMI interface (7)	Is_connected_to	PLC (Linux) (1) and PLC (Windows) (2)	Connected via Ethernet
Server (6)	Is_connected_to	Switch (7)	
Switch (7)		All	
MES Software (8)	Is_connected_to	Database (5)	Connected via Ethernet
ERP Software (9)	Is_connected_to	Database (5)	Connected via Ethernet
BDE (10)	Is_connected_to	MES Software (8)	Connected via Ethernet
Terminal PC (11)	Is_connected_to	MES Software (8)	Connected via Ethernet
Barcode Reader (12)	Is_connected_to	ERP Software (9)	Connected via Wireless
Wireless network (13)		All	

- Which are the potential sources of cyber-attacks in your pilot lines?

UC1 Production Processes Simulations for Accelerated Decisions and Safe Processes:

UC2 Production Planning Optimization:

UC3 Employee Training for Reduction of Human Errors:

UC4 Agile Production Management System Data Integrity and Reliability:

The potential source of cyber-attacks (UC1...UC4) are:

- Database

- End-points

- Communication devices

- Users

- Which are the input data utilised for running the UCs?

To be defined (UC1...UC4)

- Which are the sensors used to collect data for the UCs set-up?

Vision smart sensors, position sensors (hall effect), position transducers, photoelectric sensors, electromechanical sensors, inductive sensors and capacitive sensors (UC1...UC4)

- Which are the data sets available and related repositories?

UC1 Production Processes Simulations for Accelerated Decisions and Safe Processes:

UC2 Production Planning Optimization:

UC3 Employee Training for Reduction of Human Errors:

UC4 Agile Production Management System Data Integrity and Reliability:

To be defined (UC1...UC4)

- Is the data collection continue or does it stop during the sensors stand-by?

The data collection stops during sensors stand-by (UC1...UC4)

- Is there a Security Policy manager overlooking the Pilot?

Yes, Rafael Almeida has this role (UC1...UC4)

- Do you have already figured out a list of security policies and related rules?

UC1 Production Processes Simulations for Accelerated Decisions and Safe Processes:

UC2 Production Planning Optimization:

UC3 Employee Training for Reduction of Human Errors:

UC4 Agile Production Management System Data Integrity and Reliability:

Iber-Oleff has an ISMS (information security management system) that cover this pilot (UC1...UC4), based on best practices of ISO 27001 and NIST.

B.3 Security Policies needs assessment - Pilot Human-Robot Collaboration for Quality Management

Which are the potential sources of cyber-attacks in your pilot lines?

UC1 Easy reconfiguration for automated part handling:

- TBD

UC2 Human supervised learning for visual quality inspections: data could be poisoned, so that the models we develop are not able to accurately predict the desired classes (good, interrupted print, double print). Furthermore, evasion attacks could happen if an intruder can penetrate the system and replace images from the existing stream with adversarial samples to attack the classification (visual inspection) model.

UC3 Safe collaboration between human and robot:

- TBD

Which are the input data utilised for running the UCs?

UC1 Easy reconfiguration for automated part handling: we are planning to use a set of 9 CAD drawings for different products, potentially supplemented by pictures of the products that can serve as a database for product recognition

UC2 Human supervised learning for visual quality inspections: we used a dataset provided by Philips (3.518 labelled images).

UC3 Safe collaboration between human and robot: we are discussing the use of the FaMS model to detect user fatigue, for exact data inputs we should ask SUPSI

Which are the sensors used to collect data for the UCs set-up?

UC1 Easy reconfiguration for automated part handling: Most likely a 3D camera will be used for creating vision on the to-be detected part

UC2 Human supervised learning for visual quality inspections: we do not use any sensors. We understand Philips has some camera, to obtain the images from the manufacturing line with the purpose of quality inspection.

UC3 Safe collaboration between human and robot: Depending on the data we want to collect using the FaMS, sensors / wearables should be selected.

Which are the data sets available and related repositories?

UC1 Easy reconfiguration for automated part handling: for now, only the set with CAD drawings of the part to-be recognized

UC2 Human supervised learning for visual quality inspections: we have some other datasets made available by related partners at the EU H2020 STAR project (IBER).

UC3 Safe collaboration between human and robot: None at Philips, perhaps at SUPSI

Is the data collection continue or does it stop during the sensors stand-by?

UC1 Easy reconfiguration for automated part handling: it stops during sensor stand-by

UC2 Human supervised learning for visual quality inspections: data collections stop during the sensors stand-by. We only gather images from the production line cameras (sensors).

UC3 Safe collaboration between human and robot: Not yet discussed, assuming data collection will only occur during labelling activities of operator

Is there a Security Policy manager overlooking the Pilot?

We do not have insights into this. But all systems in the pilot setup will have to be stand-alone systems at first, meaning we will not connect them to any existing security systems next to the pilot environment.

Do you have already figured out a list of security policies and related rules?

UC1 Easy reconfiguration for automated part handling: No, not from Philips side

UC2 Human supervised learning for visual quality inspections: aspects regarding cybersecurity of machine learning models could be monitored and assessed using the [Adversarial Robustness Toolbox](#)¹, which provides a variety of tools to assess evasion and data poisoning attacks. Evasion could be detected by monitoring the inputs or using advanced techniques such as the Fast Generalized Subset Scan², which attempts to identify a set of anomalous data instances. Poisoning can be prevented using spectral signatures³ or activation clustering⁴, depending on the machine learning model implementations.

UC3 Safe collaboration between human and robot: No, not from Philips side.