

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

D2.6 – STAR Reference Architecture and Blueprints-Initial version

Dissemination level	PU -Public
Type of Document	Report
Contractual date of delivery	30/09/2021
Deliverable Leader	INTRASOFT
Status - version, date	V1.0, 30/09/2021
WP / Task responsible	WP2
Keywords:	Architecture; Industry 4.0; Artificial Intelligence; Trusted AI; Security

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 is researching, developing and validated technologies for trusted AI solutions in production lines. The technologies of the project are destined to address different dimensions and elements of trust, security and safety in the operation of Cyber Physical Production Systems. These technologies can be combined and integrated in holistic solutions in order to reinforce each other and boost the overall trustworthiness of production systems. In this direction, the present deliverable introduces the main components, functionalities, and the overall architecture of the STAR platform for trusted AI in manufacturing. The architecture will facilitate manufacturers and developers of industrial AI solutions, to design, implement and operate integrated solutions for trusted AI. It will also serve as a reference architecture that will specify components and structuring principles for trusted AI systems.

The deliverable reviews standards-based reference architecture models for industrial systems, such as the Industrial Internet Reference Architecture and its Industrial Internet Security Framework. The review aims at highlighting the pertinence of these industrial architectures to the STAR architecture and explaining how STAR architecture has leveraged some of their concepts and principles. The core of the report refers to the STAR architecture, reflected initially through a logical/conceptual view, introducing the main components of the architecture, their functionalities and the interaction between them. Following the initial conceptual view, the report presents a high-level reference model with the STAR functionalities, which can facilitate the understanding of the capabilities of the STAR platform and their efficient integration in trusted AI systems.

A number of information flows for specific use cases are also described, notably popular use cases that are classified as blueprints. Specifically, the architecture is used to specify process views for security functionalities, safety functionalities and human-robot collaboration functionalities i.e., functionalities spanning all the functional domains of STAR. Moreover, some of the project's use cases/pilots are illustrated by means of UML diagrams, along with their pertinence and mapping to the STAR architecture, to showcase the applicability of the architecture on these pilots towards the fulfilment of their requirements.

The deliverable concludes by providing an outlook for its further evolution and delivery, given that the present version of the deliverable is the first release of the STAR architecture. A second deliverable (D2.7) is destined to present more detailed and more complete views of the architecture, also considering the feedback from the implemented and validated core components of the architecture, as well as its actual use in the scope of the STAR use cases.

Deliverable Leader:	INTRASOFT
Contributors:	UBITECH, R2M, JSI, UPRC, SUPSI, DFKI, PCL
Reviewers:	UBITECH, UPRC
Approved by:	Charalampos Ipektsidis, John Soldatos (INTRASOFT)

Document History			
Version	Date	Contributor(s)	Description
V0.1	15/06/2021	INTRASOFT	Initial Table of Contents
V0.12	15/07/2021	INTRASOFT	Initial Inputs in Introductory Section
V0.15	30/07/2021	INTRASOFT	Updates to Section 2
V0.16	05/08/2021	INTRASOFT	Inclusion of Information about Reference Architectures relevant to STAR
V0.17	12/08/2021	INTRASOFT	Description of the logical view of the STAR Architecture (Section 3)
V0.18	27/08/2021	UBITECH, JSI	Inputs on Process Views
V0.19	27/08/2021	SUPSI, UPRC, R2M	Inputs on Process Views
V0.20	30/08/2021	INTRASOFT	Updates to Sections 3 and 4
V0.21	31/08/2021	INTRASOFT	Updates to the Process Views, Information on the Physical and Implementation Views
V0.22	01/09/2021	INTRASOFT	Review, quality control and various updates; Version circulated to WP2 partners for further comments and contributions
V0.23	15/09/2021	SUPSI, PCL, THALES, DFKI	Inputs on STAR Use Cases
V0.24	17/09/2021	SUPSI, PCL, THALES, DFKI, INTRASOFT	Initial list of Blueprints based on the STAR RA
V0.25	20/09/2021	INTRASOFT	Fine-tuning of use cases description
V0.26, V0.27	22/09/2021	INTRASOFT	Draft of Executive Summary; Various Edits and Fine-Tuning of the text
V0.28	27/09/2021	UBITECH	Quality Review – Comment and Edits
V0.29	28/09/2021	UPRC	Quality Review – Comment and Edits
V0.99	29/09/2021	INTRASOFT	Edits to Address Review Comments
V1.0	30/09/2021	INTRASOFT	QA and creation of the version for delivery to the EC

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.....	10
1.1 SCOPE AND PURPOSE.....	10
1.2 METHODOLOGY.....	11
1.2.1 4+1 Views Methodology	11
1.2.2 Alignment to Reference Architectures.....	11
1.2.3 Validation and Evolution	12
1.3 RELATIONSHIP TO OTHER STAR DELIVERABLES	12
1.4 DOCUMENT STRUCTURE	14
2 RELEVANT REFERENCE ARCHITECTURES.....	15
2.1 OVERVIEW AND SCOPE.....	15
2.2 INDUSTRIAL INTERNET CONSORTIUM REFERENCE ARCHITECTURE (IIRA)	15
2.2.1 Overview.....	15
2.2.2 Relevance to STAR.....	18
2.3 INDUSTRIAL INTERNET SECURITY FRAMEWORK (IISF).....	18
2.3.1 Overview	18
2.3.2 Relevance to STAR.....	20
2.4 OPENFOG REFERENCE ARCHITECTURE	21
2.4.1 Overview.....	21
2.4.2 Relevance to STAR.....	21
2.5 BDVA RA.....	21
2.5.1 Overview	21
2.5.2 Relevance to STAR.....	22
3 STAR REFERENCE ARCHITECTURE.....	24
3.1 HIGH LEVEL REFERENCE MODEL	24
3.2 LOGICAL VIEW.....	25
3.2.1 Overview	25
3.2.2 Digital Manufacturing Platforms and CPPS Systems.....	26
3.2.3 Industry 4.0 AI Applications.....	27
3.2.4 Data Probes – Data Connectors.....	27
3.2.5 Data Provenance and Traceability (DPT).....	27
3.2.6 STAR Blockchain – Distributed Ledger Infrastructure.....	27
3.2.7 AI Cyber-Defence Strategies (ACDS).....	28
3.2.8 Risk Assessment and Mitigation Engine (RAME).....	29
3.2.9 Security Policies Manager (SPM) - Security Policies Repository (SPR)	29
3.2.10 Machine Learning and Analytics Platform.....	29
3.2.11 STAR XAI (Models & Library).....	29
3.2.12 Simulated Reality (SR).....	30
3.2.13 Active Learning (AL).....	30
3.2.14 Production Processes Knowledge Base (PPKB).....	30
3.2.15 AMR Safety.....	31
3.2.16 Human Centred Digital Twin	31

3.2.17	<i>Human Models – Human Digital Images</i>	32
3.2.18	<i>Application UI – Graphical User Interface (GUI)</i>	32
3.2.19	<i>Natural Language Processing (NLP)</i>	32
3.2.20	<i>Feedback Module</i>	32
3.3	PROCESS VIEWS.....	33
3.3.1	<i>Overview</i>	33
3.3.2	<i>Defending a Poisoning Attack</i>	33
3.3.3	<i>Defending an Evasion Attack</i>	34
3.3.4	<i>Dynamic Management and Configuration of Data Sources</i>	35
3.3.5	<i>Security Policy Management</i>	36
3.3.6	<i>Human Centric Digital Twin</i>	36
3.3.7	<i>STAR XAI Models and Library Operations</i>	38
3.3.8	<i>Active Learning for Human Robot Interactions</i>	39
3.3.9	<i>Feedback Module Operation</i>	40
3.3.10	<i>NLP Interaction</i>	41
3.4	PHYSICAL VIEW CONSIDERATIONS	42
3.4.1	<i>Deployment Overview</i>	42
3.4.2	<i>Deployment & Ecosystem Management Technologies</i>	43
3.5	IMPLEMENTATION VIEW CONSIDERATIONS	47
3.5.1	<i>Implementation Overview</i>	47
3.5.2	<i>Implementation Technologies</i>	48
4	STAR ARCHITECTURE VALIDATION	49
4.1	HUMAN INTENTION RECOGNITION	49
4.2	ROBOT RECONFIGURATION BASED ON DYNAMIC FACTORY LAYOUT	50
4.3	SAFE HUMAN ROBOT COLLABORATION	51
5	BLUEPRINTS SPECIFICATION	53
5.1	INTRODUCTION.....	53
5.2	INITIAL LIST OF STAR BLUEPRINTS	53
5.2.1	<i>Defending a Poisoning Attack</i>	53
5.2.2	<i>Defending an Evasion Attack</i>	53
5.2.3	<i>Management and Configuration of Data Sources</i>	54
5.2.4	<i>Security Policy Management</i>	54
5.2.5	<i>Human Centred Digital Twin</i>	54
5.2.6	<i>Explainable Artificial Intelligence</i>	54
5.2.7	<i>Active Learning for Human Robot Collaboration</i>	55
5.2.8	<i>Feedback Provision</i>	55
5.2.9	<i>Trusted Reconfiguration for Mobile Robot</i>	55
5.3	ADDITIONAL BLUEPRINTS – FUTURE OUTLOOK	55
6	OUTLOOK AND CONCLUSIONS	57
	REFERENCES	59

Table of Figures

FIGURE 1: OVERVIEW OF THE STAR-RA SPECIFICATION AND DEVELOPMENT METHODOLOGY12

FIGURE 2: THE FIVE FUNCTIONAL DOMAINS SPECIFIED IN THE IIRA17

FIGURE 3: OUTLINE OF A THREE TIER ARCHITECTURE FOR IIoT SYSTEMS IN-LINE WITH THE IIRA18

FIGURE 4: FUNCTIONAL BUILDING BLOCKS OF THE IISF19

FIGURE 5: FUNCTIONS OF THE SECURITY MONITORING AND ANALYSIS FUNCTIONAL LAYER.....19

FIGURE 6: ALIGNMENT OF IIRA AND IISF SYSTEM VIEWS20

FIGURE 7: OVERVIEW OF STAR SECURITY AND TRUSTWORTHINESS FUNCTIONALITIES IN THE DoA20

FIGURE 8: LAYERED VIEW OF THE OPENFOG REFERENCE ARCHITECTURE (RA)21

FIGURE 9: THE BDVA/DAIRO REFERENCE ARCHITECTURE MODEL FOR BIG DATA SYSTEMS [BDVA17]22

FIGURE 10: HIGH LEVEL REFERENCE MODEL.....24

FIGURE 11: STAR FUNCTIONAL MODULES AND LOGICAL VIEW OF THE ARCHITECTURE.....26

FIGURE 12: INSTANTIATION OF THE SECURITY MODULES OF THE STAR ARCHITECTURE28

FIGURE 13: DETAILED ARCHITECTURE OF THE ACTIVE LEARNING SYSTEM FOR HUMAN ROBOT COLLABORATION30

FIGURE 14: DETAILED LOGICAL ARCHITECTURE OF THE HUMAN CENTRED DIGITAL TWIN (HDT)32

FIGURE 15: INFORMATION FLOW FOR A DEFENDING A POISONING ATTACK34

FIGURE 16: INFORMATION FLOW FOR A DEFENDING AN EVASION ATTACK35

FIGURE 17: PROCESS VIEW OF A DATA SOURCE MANAGEMENT USE CASE.....35

FIGURE 18: PROCESS VIEW OF A SECURITY POLICY MANAGEMENT USE CASE36

FIGURE 19: PROCESS VIEW OF THE HUMAN CENTRIC DIGITAL TWIN OPERATION37

FIGURE 20: PROVISION OF COUNTERFACTUALS INFORMATION BY THE STAR XAI38

FIGURE 21: FEATURES RANKING OPERATIONS BY THE STAR AI39

FIGURE 22: ACTIVE LEARNING MODULE OPERATION IN SUPPORT OF HUMAN ROBOT INTERACTION40

FIGURE 23: OPERATION OF THE FEEDBACK MODULE41

FIGURE 24: HIGH LEVEL VIEW OF THE NLP OPERATION IN THE CONTEXT OF THE STAR IMPLEMENTATION.....42

FIGURE 25 A COMPLETE GIT BRANCHING MODEL.....45

FIGURE 26: PRELIMINARY SEQUENCE DIAGRAM OF HUMAN INTENTION RECOGNITION USE CASE.....49

FIGURE 27: PRELIMINARY SEQUENCE DIAGRAM OF ROBOT RECONFIGURATION USE CASE50

FIGURE 28 HDT ARCHITECTURE INSTANCE FOR THE SAFE COLLABORATION BETWEEN HUMAN AND COBOT.....51

FIGURE 29 SAFE COLLABORATION BETWEEN HUMAN AND COBOT: SEQUENCE DIAGRAM52

List of Tables

TABLE 1: GUIDE FOR INDUSTRIAL DEPLOYMENTS AT THE CLOUD/EDGE/FAREDGE	42
TABLE 2: EDGE/CLOUD DEPLOYMENT CONSIDERATIONS FOR THE MAIN COMPONENTS OF THE STAR ARCHITECTURE .	43
TABLE 3: ENVISAGED IMPLEMENTATION TECHNOLOGIES.....	48
TABLE 4: STAR-BLPR-1 - POISONING ATTACK DEFENCE.....	53
TABLE 5: STAR-BLPR-2 - EVASION ATTACK DEFENCE.....	53
TABLE 6: STAR-BLPR-3 – MANAGEMENT AND CONFIGURATION OF INDUSTRIAL DATA SOURCES	54
TABLE 7: STAR-BLPR-4 – SECURITY POLICY MANAGEMENT	54
TABLE 8: STAR-BLPR-5 –HUMAN CENTRED DIGITAL TWIN	54
TABLE 9: STAR-BLPR-6 – EXPLAINABLE ARTIFICIAL INTELLIGENCE.....	54
TABLE 10: STAR-BLPR-7 – ACTIVE LEARNING FOR HUMAN ROBOT COLLABORATION.....	55
TABLE 11: STAR-BLPR-8 – PROVISION OF FEEDBACK IN HUMAN IN THE LOOP SCENARIOS	55
TABLE 12: STAR-BLPR-9 – TRUSTED RECONFIGURATION OF MOBILE ROBOT	55
TABLE 13: LIST OF ADDITIONAL BLUEPRINTS TO BE SPECIFIED OVER THE STAR PLATFORM.....	55

Definitions, Acronyms and Abbreviations

Acronym/ Abbreviation	Title
ACDS	AI Cyber-Defense Strategies
AI	Artificial Intelligence
AL	Active Learning
API	Application Programming Interface
ART	Adversarial Robustness Toolbox
BDVA	Big Data Value Association
CE	Community Edition
CERT	Computer Emergency Response Team
CoAP	Constrained Application Protocol
CPPS	Cyber Physical Production System
CPS	Cyber Physical System
DAIRO	Data AI and Robotics
DL	Deep Learning
DLT	Distributed Ledger Technologies
DoA	Description of Action
DPT	Data Provenance & Traceability
ERP	Enterprise Resource Planning
GUI	Graphical User Interface
HDT	Human Centered Digital Twin
HTTP	HyperText Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IIAF	Industrial Internet Architecture Framework
IIC	Industrial Internet Consortium
IIoT	Industrial Internet of Things
IIRA	Industrial Internet Consortium Reference Architecture
IISF	Industrial Internet Security Framework
IoT	Internet of Things
ISO	International Organization for Standardization
LIME	Local Interpretable Model Agnostic Explanations
MES	Manufacturing Execution Systems
MQTT	Message Queue Telemetry Transport
ML	Machine Learning
NLP	Natural Language Processing
PLM	Product Lifecycle Management
PPKB	Production Processes Knowledge Base
RA	Reference Architecture
RAME	Risk Assessment and Mitigation Engine
RFC	Request for Comments
RL	Reinforcement Learning
RM	Reference Model
SC	Smart Contract

SPC	Security Policies Manager
SPR	Security Policies Repository
SR	Simulated Reality
STT	Speech To Text
TCP	Transport Control Protocol
TTS	Text To Speech
UI	User Interface
UML	Unified Modelling Language
WIRM	Worker Intention Recognition Module
WP	Work Package
XAI	Explainable Artificial Intelligence

1 Introduction

1.1 Scope and Purpose

The main goal of the STAR project is to research, implement, validate, and demonstrate trusted AI technologies for production lines and manufacturing use cases. In this direction, the project takes a holistic approach that addresses multiple aspects of AI trustworthiness from data reliability to the cybersecurity and explainability of AI systems. In this context, STAR project designs and implements multiple prototype systems including systems for data provenance and traceability, cyber-defence against some of the most prominent attacks that target AI systems, Explainable AI (XAI) algorithms, human-centric AI-based systems such as human centric digital twins, systems for the trusted and safe operation of mobile robots in production lines, human robot collaboration systems, simulated reality systems for effective cobots and more.

Each of the aforementioned systems can be researched independently from each other. Nevertheless, many of these prototypes are also closely connected in ways that one reinforces the operation of the other. As a prominent example, data provenance and traceability mechanisms can boost the operation of cyber-defence strategies that protect AI systems from cybersecurity attacks, notably attacks that can be launched through data poisoning and data tampering. As another example, XAI systems can be used to support the operation of cyber-defence strategies (e.g., through detecting abnormalities in their operation), as well as the operation of simulated reality systems (e.g., through helping in the production of reliable data to setup the simulated environment). In this context, STAR is not limited to researching each of the above systems individually. Rather it also explores ways and methods that could boost the optimal interplay and integration of the above systems towards a holistic and efficient approach to trusted AI in manufacturing.

In quest of the best possible integration of the STAR AI systems, the project is researching the structuring principles that drive the optimal integration of the various AI prototypes, while documenting a software architecture that reflects these structuring principles. In this direction, the project introduces a reference architecture model that can support the development, deployment and operation of end-to-end integrated systems for trusted AI in production environments. The model is characterised as “reference” as it is not limited to supporting the integration and deployment of the STAR platform. Rather it is also destined to serve as a blueprint for a wider class of trusted AI systems i.e., helping integrators of AI solutions to develop and deployed trusted AI in dynamic manufacturing environments.

Hence, the presented deliverable is devoted to the introduction and documentation of the STAR reference architecture model. The model considers the specifications and functionalities of the various AI building blocks of the project’s solutions and provides a set of principles for their integration in trusted AI solutions. As already outlined, the STAR architecture model is aimed at being abstract and general to support the development of trusted AI beyond the boundaries of the project. To this end, the model is developed based on principles and concepts of existing reference architecture models for Industry 4.0, the Industrial Internet of Things, and BigData systems. Especially, existing models are extended and/or customized in order to address the project’s trusted AI vision. In this way, the STAR reference architecture model will become directly usable and exploitable by the numerous manufacturers and AI/IIoT solution providers for manufacturing environments, who are already familiar with existing reference architectures for Industry 4.0.

1.2 Methodology

1.2.1 4+1 Views Methodology

The development, documentation and presentation of the STAR architecture are based on the popular 4+1 views methodology for describing software systems architectures. The methodology [Kruchten95]. According to this methodology, a software system is described based on different viewpoints, including:

- **Logical view**, which illustrates the main functionalities provided by the system. In this direction, the main modules that comprise the system can be introduced at a logical level.
- **Process view**, which presents the dynamic aspects of the system, including the interactions between its main modules and the communications that drive the run time behaviour of the system.
- **Development view**, which presents the software development perspective of the system i.e. how the system can be perceived and implemented by a software developer.
- **Physical view**, which provides an engineering view of the system, including the physical components of the system and their interactions.
- **Scenarios (or use cases) view**, which includes a number of representative scenarios that are used to validate the architecture.

The STAR architecture is presented according to the above listed viewpoints. However, some views are preliminary at this first version of the STAR architecture document/deliverable. Specifically, in the present deliverable, the logical viewpoint of the architecture is emphasized, along with the main interactions between the various components (i.e., process view). However, information about the development of the system (i.e., development view) is given (e.g., software packages and libraries used), along with information about the physical deployment of the components that comprise the architecture (i.e., physical view). Finally, in this first version of STAR architecture deliverable, the architecture is validated against the STAR use cases that are specified in WP2. As the use cases evolve and utilize STAR research outcomes, these will also be utilized in the context of WP2 and will be reflected in a follow-up round of validation of the overall architecture. Moreover, they will be documented in the next release of this deliverable (i.e., D2.7).

1.2.2 Alignment to Reference Architectures

The STAR architecture is destined to serve as a reference (i.e. blueprint) for the development of trusted AI systems. To this end, it has been developed considering existing reference architectures for Industry 4.0 and BigData/AI systems. These architectures serve as a starting point for the definition of STAR Reference Architecture. They provide a set of important concepts/baseline regarding the structuring of data-driven and data-intensive systems in manufacturing environments. Nevertheless, existing reference architectures do not provide the means (e.g., guidelines, blueprints) for the development of trusted AI applications. The present deliverable extends existing architectures and combines their building blocks towards introducing blueprints for trusted AI systems.

As part of the deliverable different reference architectures such as the Reference Architecture of the Industrial Internet Consortium (IIRA) and the Reference Model of the Big Data Value Association (BDVA) for data-intensive systems are presented. It is illustrated how they have been considered in the development of the STAR-RA (Reference Architecture).

1.2.3 Validation and Evolution

The development and validation of the STAR architecture is influenced by on-going development and research activities of the project such as the specification of reference scenarios for Trusted AI in manufacturing, the development of the various modules of the STAR platform and solutions, the specification/evolution of Industry 4.0 and AI standards, as well as the collection and management of data for training and developing AI systems. The evolution of these activities will therefore have an essential impact on the STAR architecture. This is the reason why an agile approach to specifying, validating and documenting the STAR-RA has been chosen. In particular, the STAR-RA is specified in two iterations. The first iteration is driven by the project’s activities during the first semester of STAR’s lifetime, while the final version of the deliverable will incorporate inputs from the final versions of the STAR reference scenarios and technologies specifications. Moreover, between the two iterations, the project will use the first version of the architecture to drive the development and integration of technical/technological systems in WP3, WP4, WP5 and WP6. This will enable the project to receive feedback from the actual, implementation, deployment and use of the first version of the architecture. Accordingly, the project will use this feedback to fine-tune the specification of the architecture. Figure 1 illustrates the methodology for the specification and validation of the STAR-RA.

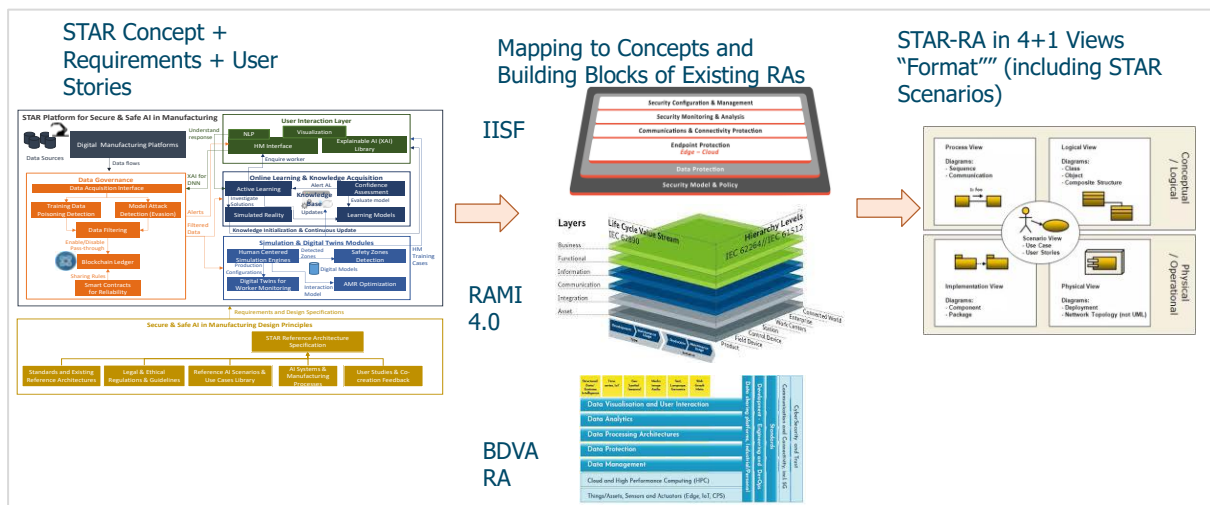


Figure 1: Overview of the STAR-RA Specification and Development Methodology

1.3 Relationship to other STAR Deliverables

The deliverable is closely related to the following STAR deliverables:

- **D2.1 Requirements Analysis and State-of-Art Research:** The specification of the STAR architecture considers the requirements and design principles/specifications provided in deliverable D2.1. Likewise, state-of-the-art systems and technologies have been taken into account in presenting the architecture implementation considerations of

this deliverable. Note that the present deliverable does not perform any in-depth state-of-the-art analysis of the technologies that will be used to implement the STAR architecture. Rather it relies on the analysis that was already performed in D2.1.

- **D2.2 Reference Scenarios and Use Cases for AI in Manufacturing:** The reference scenarios of these deliverables are considered for the validation of the architecture as part of the 4+1 methodology. The scenarios view of the present deliverable is primarily concerned with validating the architecture against the STAR use cases. Nevertheless, other scenarios have been also considered in the inception of the STAR logical architecture.
- **D2.3 Review of Applicable Standards and Regulations:** D2.3 presents an overview of industrial standards. It also includes a brief presentation of relevant reference architecture models. The latter is highly relevant to this deliverable i.e., D2.6. The present deliverable does not replicate the information of D2.3 with respect to the reference architecture, but rather provides a more in-depth review of the actual models that are used in the specification and implementation of the overall STAR architecture.
- **D2.4 Data Models and Data Collection:** Information about the data models, the data collection processes, and the underlying storage infrastructure are considered towards specifying the number and types of datastores that are needed to effectively implement the STAR architecture in terms of enabling the utilization of different datastores for different components of the overall architecture.
- **D3.1 Decentralized Reliability for Industrial Data and Distributed Analytics:** The present deliverable provides insights into the logical interactions of the components that comprise the decentralized data provenance infrastructure of the project. As such it can be considered as a forerunner of the information that will be documented in D3.1 regarding the decentralized infrastructure for data reliability in industrial environments. D3.1 is expected to provide additional information and technical details (e.g., APIs specifications) comparing to the current deliverable.
- **D3.3 Cyber-Defense Mechanisms against Poisoning and Evasion Attacks:** Similar to the case of D3.1 above, D3.3 will benefit from the initial specifications of the cyber-defense mechanisms that are provided in this deliverable. Specifically, the present deliverable illustrates some logical interactions between the main components of the cyber-defense systems, which will serve as a basis for more detailed specifications and prototyping in the context of D3.3.
- **D4.1 Library of XAI algorithms:** The explainable AI components of the STAR project will be used by various prototypes of the STAR systems in-line with the STAR architecture. The present deliverable provides insights into the different modules of the architecture that will benefit from the project's XAI library. As such it is related to deliverable D4.1, which will provide the detailed specifications and initial prototype implementations of STAR's XAI mechanisms.

1.4 Document Structure

The deliverable is structured as follows:

- Section 2 illustrates the reference architectures that have been considered for the development of the STAR-RA. Their relevance to STAR and trusted AI in manufacturing is outlined.
- Section 3 introduces the architecture in-line with the 4+1 methodology. Emphasis is paid on the presentation of logical and process views, while some development and deployment insights are also outlined.
- Section 4 validates the architecture against the use cases of the project. As already outlined the validation is preliminary.
- Section 5 presents a set of blueprints for the development of trusted AI technologies for production lines, leveraging elements and structuring principles of the STAR-RA.
- Section 6 is the final and concluding section of the deliverable. It provides a future outlook for the follow-up version of this report and the realization of the overall STAR architecture.

2 Relevant Reference Architectures

2.1 Overview and Scope

STAR is devoted to researching and developing trusted AI solutions for manufacturing environments. The project will boost the integration of such trusted technologies within industrial systems, notably within Industrial IoT and Industry 4.0 systems. Hence, beyond AI systems and algorithms, the practical deployment of STAR systems will include modules and components of broader Industry 4.0 systems, such as communication modules, IoT data collection modules, IoT data analytics modules, as well as components for industrial security. In this context, we have opted to specify the STAR architecture building on top of existing architectures for industrial systems, notably architectures that specify the modules of an Industry 4.0 system and the structuring principles that drive their integration. This is the main rationale behind reviewing and presenting standards-based architectures for industrial systems in the following paragraphs. STAR positions itself against these architectures and illustrates how its developments could be structured, integrated and used in the scope of end-to-end industrial systems.

Another reason for considering existing architectures for industrial systems is to use them as a basis for providing the specifications of the modules of the STAR architecture. This is for example the case with the specification of STAR AI systems that can be represented as data pipelines (notably Machine Learning pipelines). Such pipelines can be represented in standards-based ways i.e. ways compliant to existing reference models like the BDVA (Big Value Association) Reference Model (RM). As another example, the Industrial Internet Security Framework (IISF) provides a readily available model for integrating security systems with Industry 4.0 platforms and CPPS (Cyber Physical Production Systems).

The following paragraphs review various reference architectures and illustrate STAR's relevance to them. As already outlined, the following analysis extends and deepens relevant work undertaken as part of other deliverables of the project (i.e., D2.1 and D2.3).

2.2 Industrial Internet Consortium Reference Architecture (IIRA)

2.2.1 Overview

The IIRA prescribes a standards-based for developing, deploying and operating IIoT systems [IIRAv1.9]. It is destined to boost interoperability across different IoT systems and to provide a mapping on how different technologies can be exploited towards developing IIoT systems. The IIRA is described at a high level of abstraction, as it strives to have broad applicability. Its specification has been driven by the analysis of a rich collection of industrial use cases, notably use cases prescribed in the scope of the activities of the Industrial Internet Consortium (IIC).

The IIRA is described based on the ISO/IEC/IEEE 42010:2011[IEEE42010] standard, which has been adopted by IIC to define its Industrial Internet Architecture Framework (IIAF). The IIAF is a quite different methodology than the 4+1 views used in the STAR. This is due to the need for IIRA to address a broader audience and range of stakeholders, when compared to the STAR architecture.

In-line with the selected methodology for architecture presentation, IIRA is defined in terms of four complementary viewpoints:

- **The Business Viewpoint** presents the functional modules that are destined to support the business goals of different stakeholders.
- **The Usage Viewpoint** presents the way systems compliant to IIRA are used. It includes various sequences of activities involving human or logical actors, which deliver the functionality prescribed by the architecture.
- **The Functional Viewpoint** presents the functional components of an IIoT system, including their structure and interrelation, as well as the interfaces and interactions between them.
- **The Implementation Viewpoint** is devoted to the presentation of the technologies that are used to implement the various functional components and their interactions.

In the scope of the IIRA there are also cross-cutting elements i.e. elements and functions that are applicable to all viewpoints.

The above-listed four viewpoints can be mapped to the various view of the 4+1 methodology. For instance, there is a direct mapping of the functional viewpoint to the logical view and a mapping of the implementation viewpoint to the implementation view. The present deliverable is particularly concerned with the functional and implementation viewpoints of the IIRA due to their direct mapping to the target views of the STAR architecture.

The functional viewpoints of the IIRA specify five sets of functionalities, which are depicted in Figure 2 and are characterized as functional domains. These include:

- **The Control Domain**, which comprises functions conducted by industrial control and automation. Control domain functions enable the creation of closed-loop systems that read data from field sensors (“sensing”) and apply control over the physical world (“actuation”).
- **The Operations Domain**, deals with the management and operation of the control domain. It comprises functions for the provisioning, management, monitoring and optimization of control domain systems and functions. Specifically, it comprises prognostics, diagnostics, optimization, provision, deployment, and asset management functions, among others.
- **The Information Domain**, focuses on managing and processing data from other domains, notably from the control domain. It collects and analyses data from various domains towards creating high-level intelligence about the overall industrial system.
- **The Application Domain**, which provides the application logic required to implement the various business functions. It comprises logic, rules, APIs and UIs (User Interfaces).
- **The Business Domain**, implements business logic that supports business processes and procedural activities in the scope of an IIoT system. It facilitates the end-to-end integration of IIoT functionalities, including integration with business information systems like Enterprise Resource Planning (ERP), Product Lifecycle Management (PLM), and Manufacturing Execution Systems (MES).

The IIRA specifies cross-cutting functions that support all the different domains, including connectivity, industrial analytics, distributed data management, as well as intelligent and resilient control.

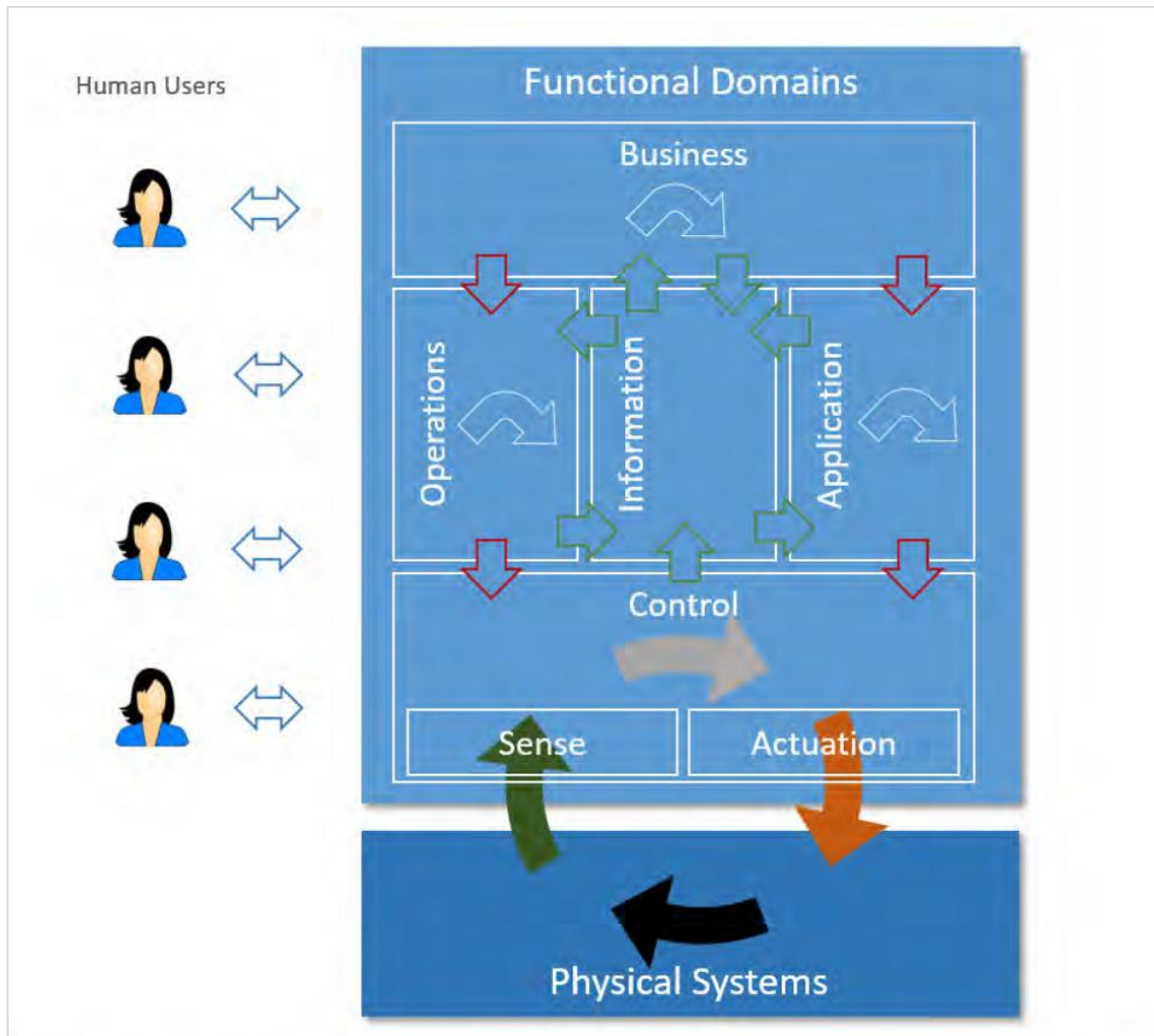


Figure 2: The Five Functional Domains Specified in the IIRA

The implementation viewpoint of the IIRA illustrates patterns and technologies that can support the implementation of IIoT system. It focuses on: (i) The components of an IIoT system and the structuring principles that drive their interconnection i.e. the general architecture of an IIoT system; (ii) A technical description of the components of an IIoT system, including its interfaces, protocols and behaviors; and (iii) A mapping of the activities of the usage viewpoint to functional components, along with a mapping of functional components to implementation components. In this direction, the IIRA specifies the use of popular patterns for the implementation of industrial systems, such as: (i) the three-tier architecture pattern (illustrated in Figure 3); (ii) The Gateway-Mediated Edge Connectivity and Management architecture pattern; and (iii) The Layered Databus pattern applied at different levels of an industrial system i.e. field level, site level and supply chain level.



Figure 3: Outline of a Three Tier Architecture for IIoT systems in-line with the IIRA

2.2.2 Relevance to STAR

The IIRA provides a taxonomy of the main functional areas of industrial systems, which is used towards specifying the STAR architecture and mapping/classifying the STAR technical modules based on their functionalities. For instance, the STAR RA clusters the functionalities of the STAR modules in three domains, much in the same way the IIRA specifies different functional domains.

Also, the IIRA illustrates how specific functions like asset management and cyber-security functions can be integrated with IIoT systems. Moreover, the functional and implementation viewpoints of the IIRA provide insights about how to best structure the logical and implementation view of the STAR architecture.

2.3 Industrial Internet Security Framework (IISF)

2.3.1 Overview

The IISF complements the IIRA with a security viewpoint for industrial systems. One of the main objectives of the IISF is to prescribe the functions needed for the development, deployment and operation of trusted IIoT, much in the same way STAR is concerned about the trustworthiness of AI systems in industrial environments. Hence, STAR considers the structure and functions of IISF as a basis for supporting the trustworthiness of AI systems in manufacturing. Note that IISF specifies functionalities that are destined to secure all the different elements of an industrial system such as the various communication endpoints of the system. Most of these functions are relevant to the security, safety and trustworthiness functionalities of the STAR systems as well, yet STAR research concentrates only on the functions that are relevant to AI systems.

IISF is concerned with the five main characteristics that affect the trustworthiness of IIoT deployments, namely security, safety, reliability, resilience and privacy. These characteristics are relevant to various STAR modules and technologies. The framework specifies a

functional viewpoint that is destined to secure IIoT systems compliant to the IIRA. To this end, the functional viewpoint specifies six interacting and complementary building blocks, which are organized in a layered fashion. The top layer comprises the four security functions (as shown in Figure 4), namely endpoint protection, communications and connectivity protection, security monitoring and analysis, and security configuration management. Likewise, a data protection layer and a system-wide security model and policy layer are specified to support the above-listed functions.

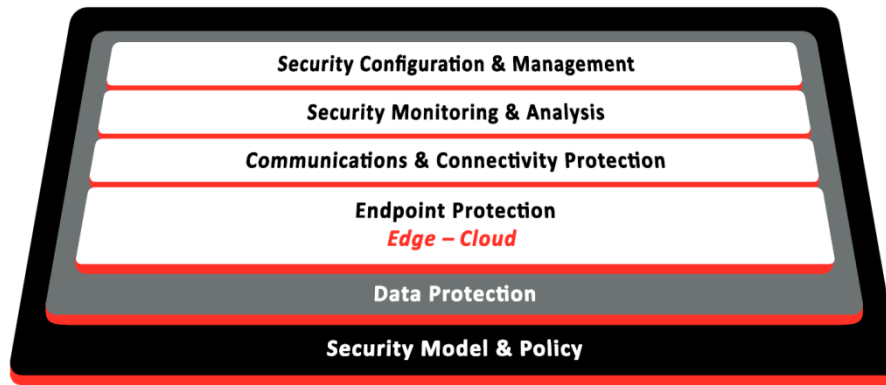


Figure 4: Functional Building Blocks of the IISF

Each one of the functional building blocks of the IISF can be further analyzed in more fine grained functions. This is illustrated in Figure 5, which details the security monitoring & analysis building block of the IISF. Specifically, it presents how this building block is broken down into three types of functionalities including monitoring functionalities, analysis functionalities and actuation functionalities. Each of these three types of functionalities including individual security related functions. Note that the “Security Monitoring and Analysis” functional layer of the IISF is particularly relevant to several research and development activities of the STAR project, notably to the activities that analyse industrial data in order to detect security or trust issues, while at the same time initiating actions to remedy them. In short, STAR implements several Monitor→Analyze→ Act functionalities as AI/ML pipelines.

Security Monitoring & Analysis

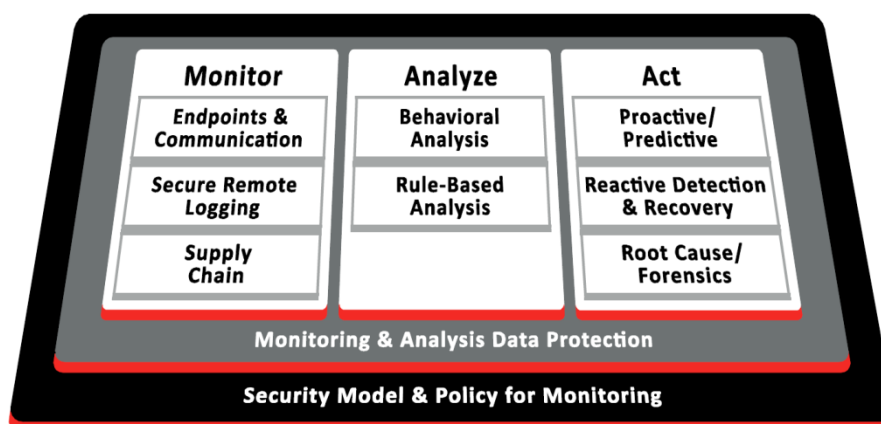


Figure 5: Functions of the Security Monitoring and Analysis Functional Layer

Finally, Figure 6 illustrates the alignment between the IISF and IIRA. It makes evident that the IISF is destined to secure industrial functions developed in-line with the IIRA functional

architecture, including actuation, sensing, control, information management, operations and applications functions.

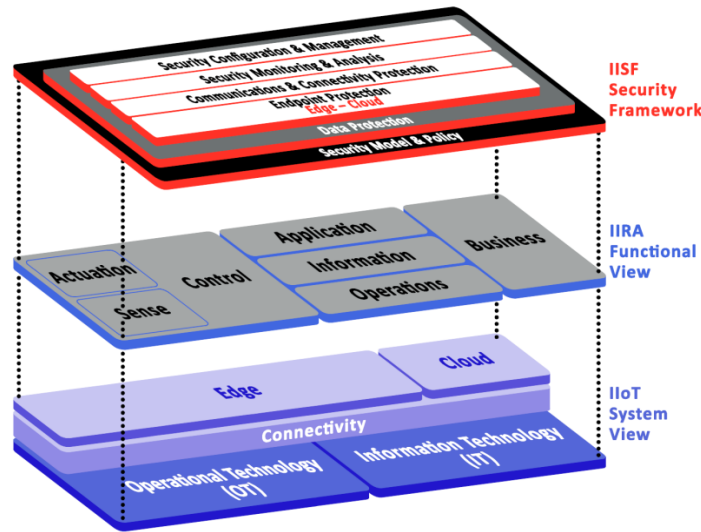


Figure 6: Alignment of IIRA and IISF System Views

2.3.2 Relevance to STAR

The relevance of the IISF to STAR is already outlined in the STAR DoA. STAR deals with security and trustworthiness functionalities in industrial AI systems, which are used to protect AI-based CPPS systems in industrial plants. This resembles the approach taken by IISF to protect IIRA compliant systems and is illustrated in Figure 7. The figure depicts that STAR’s research modules (e.g., the Simulated Reality, Active Learning and Cyber-Defence modules developed in the project) operate over a functional layer of CPPS systems that provide automation and control functionalities. Overall, STAR is well aligned to the IISF concept of protecting CPPS systems across different functional domains.

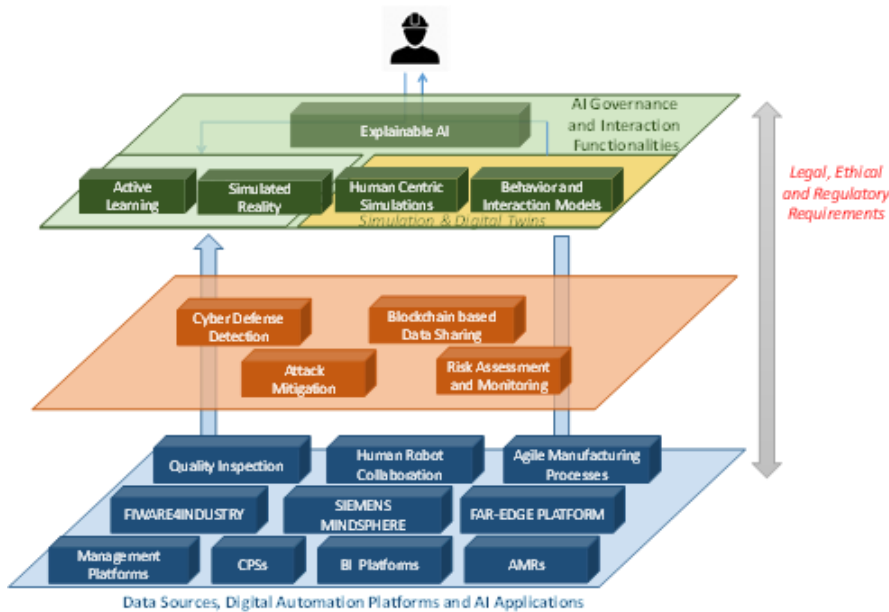


Figure 7: Overview of STAR Security and Trustworthiness Functionalities in the DoA

2.4 OpenFog Reference Architecture

2.4.1 Overview

The OpenFog Consortium was established in 2015 as a consortium of high-tech industrial enterprises companies and research centres, which had the mission of developing and promoting fog computing standards for different use cases and vertical sectors. Since 2019, the members of the consortium were incorporated within the Industrial Internet Consortium. Prior to the incorporation, the consortium specified a Reference Architecture (RA) for fog computing systems, which illustrates how fog nodes are connected to enhance the intelligence and boost the efficiency of Industrial IoT systems. The RA specifies the structure of large-scale fog systems that comprise multiple nodes. A layered view of the architecture is provided in Figure 8, yet the RA is describes based on other views as well, including functional and deployment views. The RA specifies some cross-cutting functionalities, which are characterized as “perspectives”.

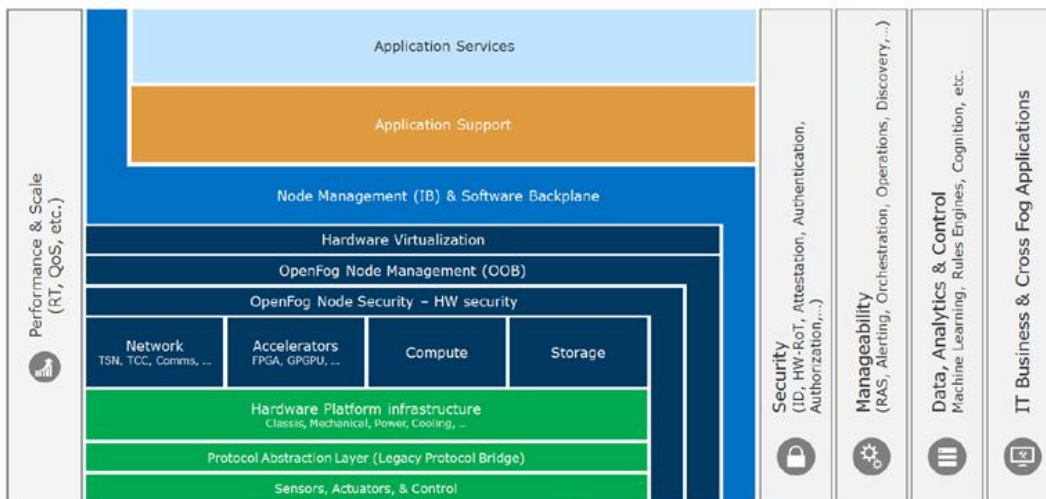


Figure 8: Layered View of the OpenFog Reference Architecture (RA)

One of these perspectives, deals with the security functionalities, which implies that security is applicable to all layers and use scenarios from the hardware device to the higher software layers of the architecture.

2.4.2 Relevance to STAR

The OpenFog RA confirms the cross-cutting character of the security and trustworthiness functions that will be developed in the project. Furthermore, the RA provides insights on how to secure fog computing and edge computing systems. This is useful for the STAR architecture, given that its implementation will be based on the cloud/edge paradigm as outlined in the following sections.

2.5 BDVA RA

2.5.1 Overview

The Big Data Value Association (BDVA)¹ has specified the structure of big data systems based on the introduction of a reference model for big data system. The model illustrates a

¹ BDVA has recently evolved to the DAIRO (Data Artificial Intelligence and Robotics) Association

set of modules that are commonly used in big data systems along with structuring principles that drive their integration. A high-level overview of the BDVA reference model is provided in Figure 9. It consists of:

- **Horizontal layers** that illustrate the modules and the structure of data processing chains. The modules of data processing chains support functions like data collection, data ingestion, data analytics and data visualization. Note however that the horizontal layers of the BDVA RM do not map to a layered architecture, where all layers must co-exist in the scope of a system. For instance, it is possible to have a data processing chain that leverages data visualization and data collection functions (i.e., visualizing collected data) without necessarily using data ingestion and data analytics functionalities. STAR leverages the structure of the BDVA horizontal layers to represent data pipelines for the project’s AI systems.
- **Vertical layers** that deal with cross-cutting issues such as cybersecurity and trust. The latter affects all the horizontal concerns i.e., they are applicable to all functionalities of the horizontal layers. Likewise, vertical layers can be used to specify and address non-technical aspects such as the ever important legal and regulatory aspects of AI systems.

The horizontal and vertical layers of the reference model drive the specification of detailed architectures for big data systems. Specifically, the various layers are used to map big data functionalities to concrete architectures. Note also that the architecture addresses big data systems, rather than AI systems. However, there are clearly many commonalities between big data and AI systems as many AI systems (e.g., the majority of deep learning architectures) are data-intensive and process large amounts of data.

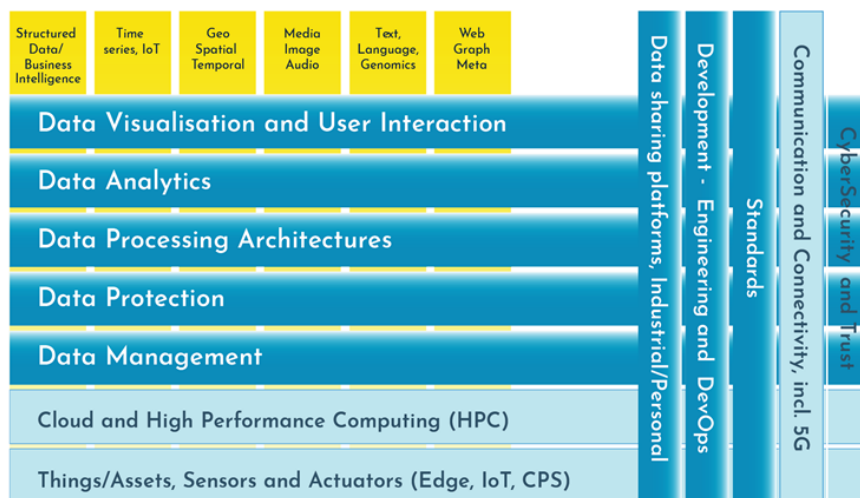


Figure 9: The BDVA/DAIRO Reference Architecture Model for big data systems [BDVA17]

2.5.2 Relevance to STAR

STAR will leverage the BDVA reference model to identify the data processing functions of the project’s AI systems. Specifically, the project uses data processing functions of the horizontal layers of the reference model to specify STAR’s AI systems in the form of machine learning pipelines. The latter can be model as a processing chain that leverages machine learning in the analytics part. Beyond pipelines modelling and specification, STAR will follow the structure of the reference model when defining AI systems, including the vertical cross-

cutting functions of the model. However, STAR is oriented to security and trust for industrial systems rather than big data, which is the reason why the architecture will primary follow the IISF structure as illustrated in Figure 7.

3 STAR Reference Architecture

3.1 High Level Reference Model

In line with earlier deliverables on the main requirements of the STAR platform (i.e., deliverable D2.1) and the project’s DoA (Description of Action), the main functionalities of the STAR platform can be clustered in three main categories (or domains according to the IIRA terminology). These three domains are illustrated in Figure 10, which provides a high-level reference model for the functionalities of the STAR platform. The domains are as follows:

- **Cybersecurity Domain:** Comprises functionalities that are destined to ensure the reliability and security of industrial data, as well as of AI algorithms that are trained and operational based on them. The functionalities of these domains support and reinforce the trustworthiness of the project’s functions in the other two domains.
- **(Trusted) Human Robot Collaboration Domain:** Provides functionalities for the trusted collaboration between human and robots. Leverages cybersecurity functionalities, while being used to reinforce functionalities in the safety domain as well.
- **Safety Domain:** Ensures the safety of industrial operations, including operations that involve workers and/or automation systems. For instance, functionalities in this domain reinforce worker safety, while catering for the safe operation of AMRs in industrial sites.

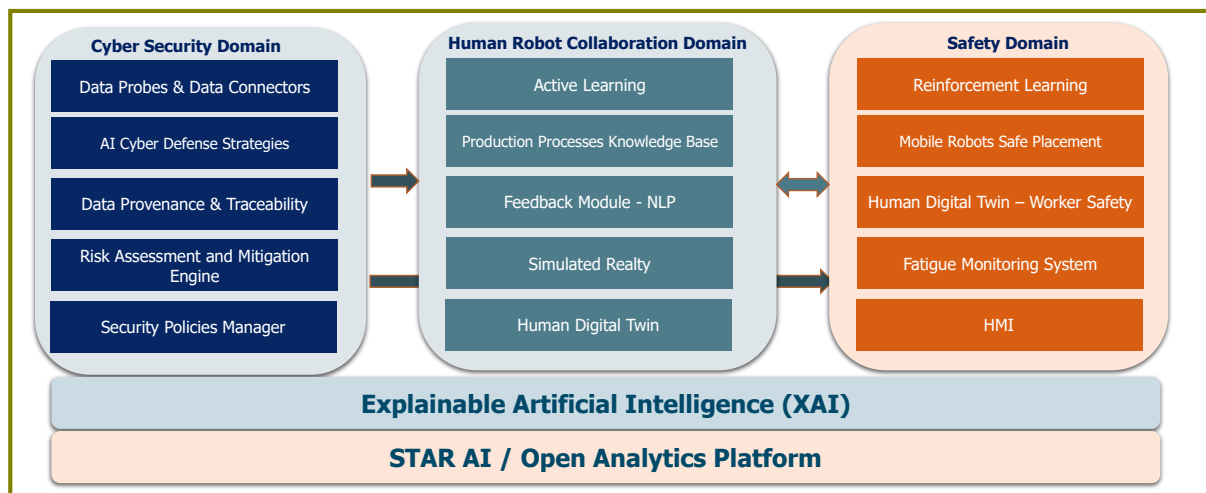


Figure 10: High level Reference Model

As illustrated in Figure 10, the functionalities of all domains depend on XAI and AI algorithms. As such, they depend on the STAR AI platform and on the XAI models developed on top of it. XAI plays an instrumental role in the operation of the security platform, as it supports defense strategies (in the cybersecurity domain), data generation for simulated reality and active learning functionalities (in the human robot collaboration domain), as well as the development of human digital twins (in the safety domain). The functionalities that are part of each domain are illustrated in the following paragraph, as part of the presentation of the logical view of the architecture.

3.2 Logical View

3.2.1 Overview

A logical view of the STAR architecture is presented in Figure 11. 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:

- **Secure and Reliable Data:** The STAR AI systems must operate over reliable industrial data i.e., the STAR architecture must make provisions to alleviate the inherent unreliability of industrial data.
- **Secure and Trusted AI algorithms:** The STAR systems must enhance the secure operation of the AI systems and algorithms that they comprise. In this direction, they must make provisions for implementing cyber-defense strategies that protect and defend AI systems from malicious security attacks. STAR focuses primarily on defenses against cyber-security attacks. Physical security attacks are applicable to some STAR systems (e.g., the robotics systems of the project), yet they are not considered in the scope of the STAR project.
- **Trusted Human AI interactions:** STAR focuses on the implementation of trusted interactions between humans and AI systems. On the one hand, the project aims at ensuring that AI systems are transparent and explainable to humans towards boosting their acceptance and adoption. On the other, the project focuses also on safe and trusted interactions between humans and AI systems in scenarios like human robot collaboration.
- **Safe AI systems:** STAR includes research towards ensuring the safety of autonomous AI systems such as mobile robots. It focuses for example on the secure placement and movement of Autonomous Mobile Robots (AMRs) in the context of the plant. These systems fall in the broader scope of the safe operation of autonomous systems.

The above-listed pillars can work in isolation, but also in synergy. For instance, Reinforcement Learning (RL) algorithms can be used to ensure the safe operation of AMRs, which contributes to the trusted operation of AI systems. These RL algorithms can operate independently from other STAR modules. However, they can also be integrated with STAR's industrial data reliability systems towards ensuring that they operate over reliable data. This boosts and reinforced their trustworthiness. Moreover, they can be integrated with the STAR's cyber-defense strategies to ensure that they cannot be tampered with or compromised by malicious parties. This is yet another step to strengthen the trustworthiness of STAR systems for safe AMR operation. Overall, when integrating and combining multiple STAR systems, manufacturers and system integrators can gain a multiplicative trustworthiness benefit, as one system can reinforce the other. The STAR architecture provides the structuring principles for integrated the project's systems for trusted AI.

As illustrated in Figure 11 the STAR systems receive data from the shopfloor (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).

Note that the STAR architecture does not prescribe an all-of-nothing use of the various modules. Rather, STAR compliant systems can be developed based on subsets of the modules of the architecture towards providing specific security and trustworthiness functionalities. This is illustrated in the following paragraphs, where instantiations of the reference architecture are given.

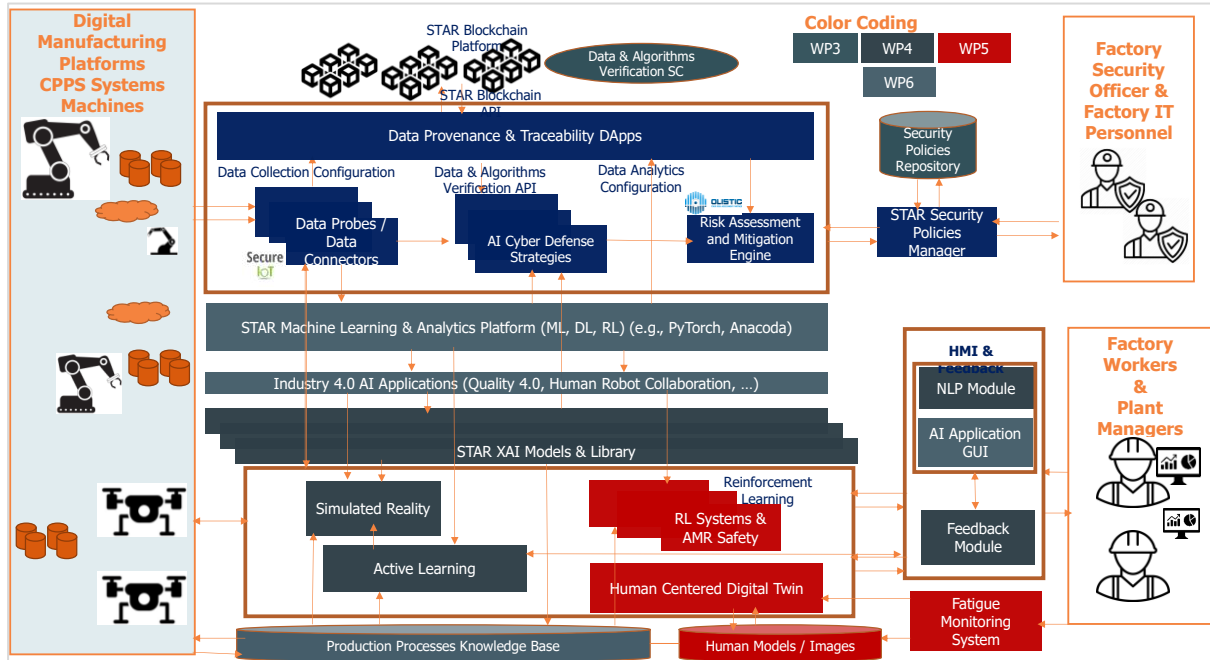


Figure 11: STAR Functional Modules and Logical View of the Architecture

The following paragraphs describe the various modules and building blocks of the architecture by providing a summary of their core functionality. Additional technical details will be provided in the respective design and implementation deliverables of WP3, WP4 and WP5. However, since the scope of the current deliverable is the capturing and description of the interaction between main building blocks and components, a summary is provided, while a detailed view of their interactions (reflected in the corresponding sequence diagrams) follows – i.e. Process Views.

3.2.2 Digital Manufacturing Platforms and CPPS Systems

STAR’s primary goal is to ensure the secure, safe and trusted operation of AI systems in production lines. To this end, the STAR modules will collect and process data from AI-based systems in the shopfloor, including machines, robotic cells, AMRs and digital manufacturing platforms that implement AI-based algorithms. A practical instantiation of the STAR architecture will therefore comprise a series of CPPS systems and digital manufacturing platforms, which will serve as data sources that will connect to the STAR modules. The STAR modules may also consume data from other data sources in the shopfloor like business information systems (e.g., ERP (Enterprise Resource Planning)) and manufacturing databases (e.g., historian databases). The latter systems are not AI systems per se, yet their data are used/consumed by AI systems of the plant.

3.2.3 Industry 4.0 AI Applications

This main building block represents different types of AI-based industrial applications such as machine learning and robotics applications. They leverage information and data sources from the shopfloor. In some cases, they are integrated with the digital manufacturing platforms. Other STAR modules of the architecture collect data from them and analyse their behaviour towards boosting the security and trustworthiness of their operation. Similar to the CPPS systems and digital manufacturing platform, AI applications can be data sources, which contribute data to the operation of STAR's data driven systems.

3.2.4 Data Probes – Data Connectors

Data probes provide the means for acquiring data from the shopfloor sources (e.g., CPPS systems and digital manufacturing platforms). They provide interfaces for transferring data from the shopfloor to the realm of a STAR deployment. Probes and data connectors connect to the shopfloor systems via some connectivity protocol (e.g., IoT protocols like MQTT (Message Queue Telemetry Transport) and CoAP (Constrained Application Protocol)² over transport protocols like TCP (Transport Control Protocol) and HTTP (Hyper Text Transfer Protocol) respectively). Probes and connectors are envisaged as configurable components: They can be configured to interface to different data sources, using different protocols, data rates and security requirements.

3.2.5 Data Provenance and Traceability (DPT)

This block provides the means for tracking and tracing industrial data, notably the industrial data that are used by a STAR system. To this end, it interfaces to the data probes i.e. each data probe provides to the DPT module information about the data acquired from the shopfloor (e.g., information about data types, volumes, timestamps, etc.). The DPT module is aimed at reinforcing the reliability and the security of the source data used in the STAR system. It records information (i.e., metadata) about the acquired data to facilitate the detection of abuse and tampering attempts against these data. Specifically, data ingested in the DPT can be queried by other STAR modules to facilitate the validation of datasets and to ensure that the data that are used have not been tampered.

3.2.6 STAR Blockchain – Distributed Ledger Infrastructure

There are different ways for implementing a DPT infrastructure for industrial data. STAR promotes a decentralized approach, which leverages the benefits of a distributed ledger infrastructure i.e. blockchain. Specifically, distributed ledger infrastructures offer some advantages for industrial data provenance, such as the fact that they are tampered proof [Soldatos21], [Soldatos21a]. The STAR blockchain facilitates the implementation of Smart Contract (SC) over the distributed ledger infrastructure, as a means of validating the metadata of the industrial datasets that are recorded in the blockchain. Moreover, SC enables decentralized applications that provide information about the metadata to interested STAR modules such as the cyber-defence strategies module.

² CoAP is defined in IETF RFC 7252

3.2.7 AI Cyber-Defence Strategies (ACDS)

This module implements cyber-defence strategies for AI systems i.e., strategies that protect AI systems against adversarial attacks. These strategies operate based on access to industrial data from:

- The AI systems (including ML systems) that must be protected from cyber-security attacks.
- The CPPS and digital manufacturing platforms data sources.
- The relevant metadata of the industrial data from the DPT module and its blockchain implementation.
- The Explainable AI (XAI) module, which implements explainable AI models that illustrate and interpret the operation of various AI systems and algorithms.

The module implements different strategies in response to various attacks against AI system. STAR researches and implements cyber-defence strategies for certain types of attacks against AI systems, notably poisoning and evasion attacks. Nevertheless, additional cyber-defence strategies can be implemented and integrated with the rest modules (i.e., data probes, DPT) in the same way. In this direction, the AI Cyber-defence strategies module comprises a set of data-driven cyber-defence templates that implement distinct strategies. This is in-line with the objective of the STAR architecture to serve as a reference architecture: Many cyber-defence strategies can be implemented using industrial data from the CPS systems and metadata from the CPPS, even though STAR will implement a few examples only (e.g., defense strategies for evasion and poisoning attacks).

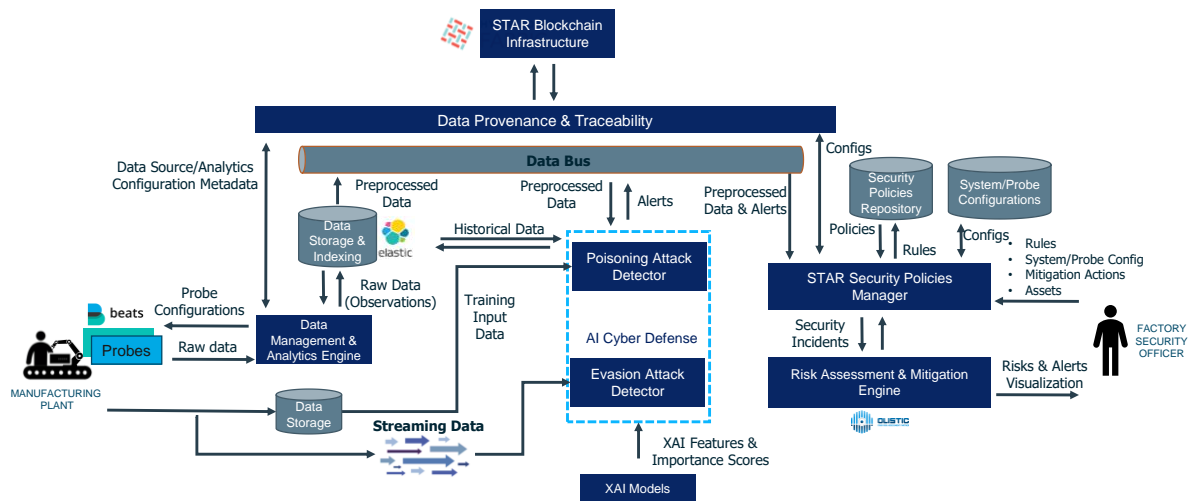


Figure 12: Instantiation of the Security Modules of the STAR Architecture

Figure 12 provides a logical view of how the ACDS can be instantiated and accordingly broken down into additional modules for detecting specific attacks like poisoning and evasion attacks. Specifically, the figure depicts two attack detectors (i.e. two attack detection templates) namely a Poisoning Attack Detector and an Evasion Attack Detector. As already outlined the operation of these detectors is based on interaction with the probes via a Data Management and Analytics Engine, as well as with the STAR XAI models and library. Furthermore, Figure 12 introduces a Data Bus as a data exchange middleware infrastructure

(i.e. a data exchange pattern), which facilitates data transfer and data sharing across different elements of the STAR systems, including the DPT, the ACDS and the SPM.

3.2.8 Risk Assessment and Mitigation Engine (RAME)

This module implements the main security service of the STAR platform i.e. security risk assessment and mitigation. Risk assessment and mitigation is one of the most important security functions for industrial systems. The module is destined to assess risk for assets associated with AI-based systems in manufacturing lines. In this direction, it also interacts with the AI cyber-defence strategies modules: (i) The defence strategies communicate to the RAME information about identified risks for AI assets; and (ii) The RAME consumers information from the DPT to assess risks. Likewise, it offers mitigation actions for the identified risks, including risks implemented through the STAR platform (e.g., changing the configuration of a probe).

3.2.9 Security Policies Manager (SPM) - Security Policies Repository (SPR)

This module specifies and configures security policies that are destined to govern the operation of the DPT, AI Cyber-Defence and RAME modules. Specifically, the module specifies security policies that provide information about the probes and data sources to be integrated, the configurations of the probes, as well as the cyber-defence strategies to be deployed. By changing the applicable policies, the SPM changes the configuration and the operation of the STAR security systems (DPT, RAME, ACDS). The operation of the SPM is supported by a Security Policies Repository (SPR), where policy files are persisted. Furthermore, the SPM offers a GUI (Graphic User Interface) to the security officers of the factory (e.g., members of CERT (Computer Emergency Response Teams)).

3.2.10 Machine Learning and Analytics Platform

Several STAR modules are based on machine learning algorithms, including deep learning and reinforcement learning. This is for example the case of the ACDS module, which implements data-driven, AI-based defence strategies among others. Another prominent example is the XAI module of the project, which produces explainable ML models. To support the operation of the STAR AI systems, the architecture specifies a machine learning and analytics platform. The platform enables users of the STAR modules (i.e. data scientists, domain experts, ML engineers) to specify and execute ML models, but also to access their metadata and outcomes. All functional modules of the architecture that execute AI algorithms (e.g., ACDS, Active Learning (AL), XAI) interact with the ML and analytics platform, as the platform enables the execution of AI models. Likewise, the platform interacts with modules that contribute or provide datasets for training and executing AI algorithms such as the data probes and the data connectors.

3.2.11 STAR XAI (Models & Library)

This module provides and executes Explainable Artificial Intelligence models and algorithms. Similar to the ACDS module, it provides the means for executing different types of XAI algorithms such as algorithms for explaining deep neural networks and general-purpose algorithms (e.g., LIME - Local Interpretable Model Agnostic Explanations-) that explain the outcomes of AI-based classifiers. As such the module is a placeholder of XAI techniques. The latter are structured as a library of algorithms. XAI provides its services to several other

modules that leverage explainable algorithms for their operation, such as the AI Cyber Defence Strategies module and the Simulated Reality (SR) module.

3.2.12 Simulated Reality (SR)

This module simulates production settings in a virtual world with a twofold objective: (i) Producing data to be used by AI algorithms, especially in cases where real world data are not available in adequate quantities; and (ii) Utilizing reinforcement learning techniques in artificial settings (i.e., simulated environments) towards accelerating the convergence of RL techniques. SR leverages services from the XAI module, which facilitates humans to assess the appropriateness and correctness of the simulated data that are generated by the SR.

3.2.13 Active Learning (AL)

This module provides a placeholder for AL systems i.e., AI systems that are able to consult an authority (e.g., a human) in cases where they lack data/information to take proper decisions. In the STAR reference architecture, this module is a placeholder for different AL techniques. In the scope of the STAR implementation, the module is further decomposed into several other modules that comprise its practical implementation.

A detailed logical architecture of the implementation of the AL module in a human robot collaboration is detailed in Figure 13, yet its description is beyond the scope of this deliverable, but is detailed in an initial relevant publication by STAR partners researching on this topic [Rozanec21].

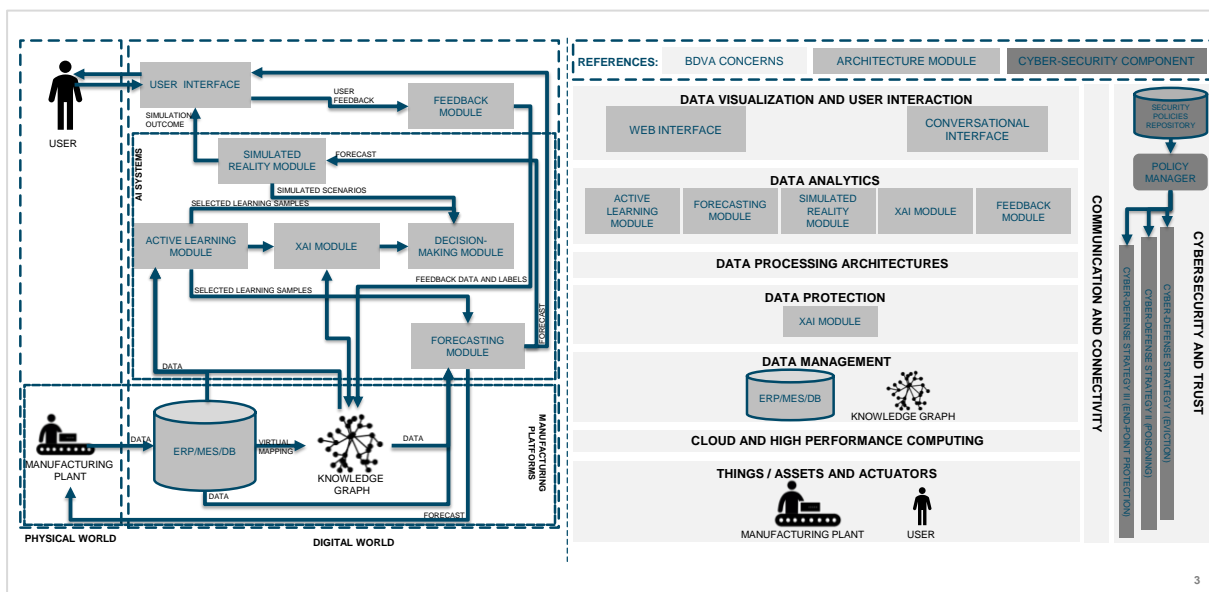


Figure 13: Detailed Architecture of the Active Learning System for Human Robot Collaboration

3.2.14 Production Processes Knowledge Base (PPKB)

This module consolidates domain knowledge about the production processes of the manufacturing environment. It is used for inferencing by the other STAR modules such as the AL module. It is also updated by AL module whenever the AL consults the authority. This helps accelerating knowledge acquisition.

3.2.15 AMR Safety

This module comprises RL techniques and is used to boost the safety of AMRs in manufacturing environments. It is used to provide insights on the safe placement of robots in a manufacturing environment.

3.2.16 Human Centred Digital Twin

This module implements a digital twin that factors human-centred parameters (e.g., fatigue, emotional status of the worker). It is a placeholder for digital twins of human-centered processes, including AI-based processes that have the human in the loop. It interacts with the analytics platforms, the workers and the humans' digital models.

The Human Digital Twin (HDT) architecture offers a centralized access point to exploit a wider set of workers' related data. STAR creates a digital representation of the workers, seamlessly integrated with production system DTs, that can be exploited by AI-based modules to compute complex features, feeding and enriching the HDT itself, or to make better decisions, dynamically adapting automation systems behaviour targeting both production performance and workers' safety and well-being. The HDT architecture is composed of the following components, as depicted in Figure 2:

- **Shop-floor entities, agents and gateways:** sensors, wearables and PLCs collect and stream data from the shop-floor. To facilitate data gathering from the workers and the production system entities, the HDT integrates agents and gateways to ensure the data collection, harmonisation and accessibility from heterogeneous sources and to create bridges between these sources and the upper layers.
- **IIoT Middleware:** this layer supports M2M connection and it is based on the MQTT lightweight messaging protocol. It allows bi-directional communication under a publish-subscribe mechanism and the organisation of important amounts of heterogeneous data into multiple topics. Each user has a set of channels where data are streamed to and accessed by the modules that need them for further computations.
- **Data storage and Time Series Data Storage:** in the data storage all the structure and core information about the HDT are stored. In addition, the workers' quasi-static data are persisted in this component. Meanwhile, the Time Series Data Storage acts as a backlog of sensors data, in which the various entities of the HDT can access in order to make predictions or extract feature for computations.
- **Orchestrator and Models:** this component is responsible to manage all the entities in the HDT. It knows exactly which kind of data each sensor is producing, who are the workers online and where their data are published. In addition to that, it also knows the modules currently in use, which information they take as input and where they publish their outputs. Models are a set of descriptors defined by the administrator of the HDT that describes any worker or contextual feature.
- **Functional Modules monitoring modules (Data processing, analysis and decision modules):** these modules allow to elaborate data from workers, contextual sensors, or any kind of system that publishes data on the IIoT Middleware. These modules target the detection of human status and conditions and compute complex features to allow human and machines decision-makers to consider the human factors within their execution and control logics.

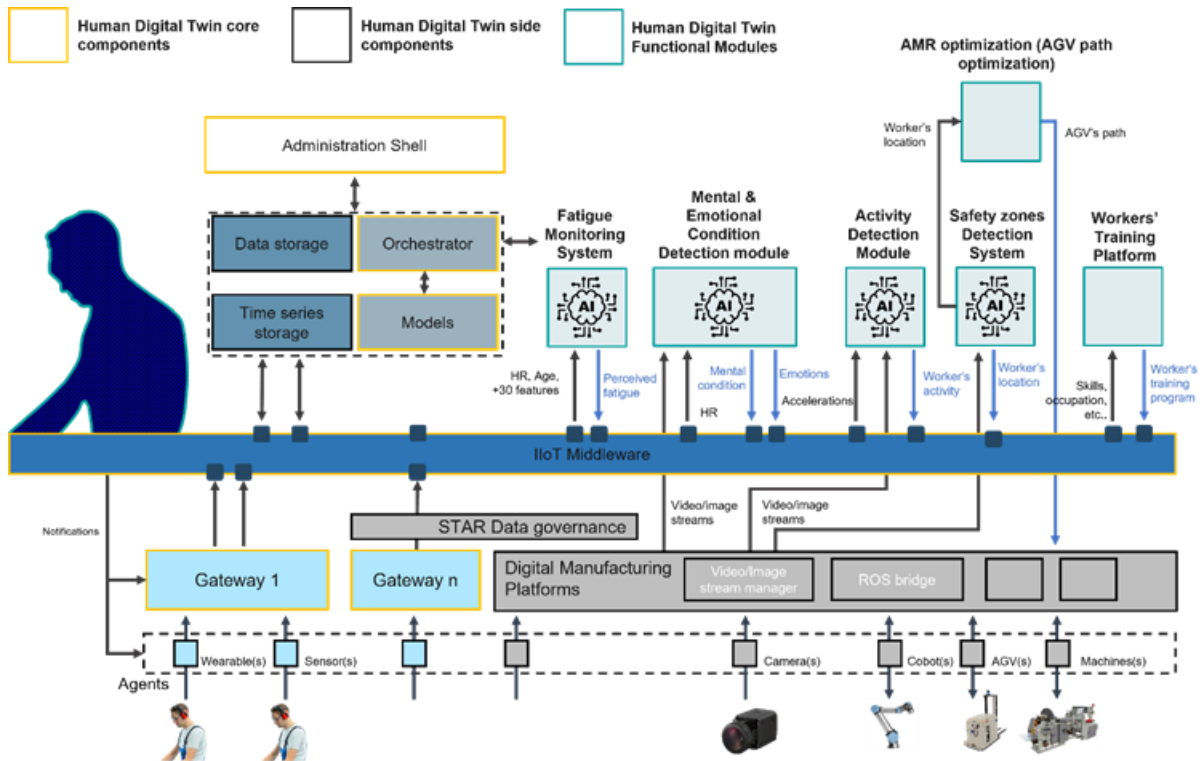


Figure 14: Detailed Logical Architecture of the Human Centred Digital Twin (HDT)

3.2.17 Human Models – Human Digital Images

This module persists and manages data about the human worker towards supporting the construction, deployment and operation of human centric digital twin.

3.2.18 Application UI – Graphical User Interface (GUI)

This module provides a GUI interaction modality between factory workers/users and STAR AI systems. It comprises visualization elements (e.g., dashboards), while enabling users to interact with the STAR modules (e.g., provide form-based input).

3.2.19 Natural Language Processing (NLP)

This module enables NLP interactions between the factory users and STAR AI modules. It is a placeholder for different NLP implementations and interfaces to different STAR modules. In the context of the STAR implementation, NLP modality is used for the interaction between workers and the AL module.

3.2.20 Feedback Module

This module coordinates the provision of feedback from the human worker to the AI system. It is particularly important for the implementation of human-AI systems interactions (e.g., human robot collaboration scenarios). The feedback module interfaces to some interaction module (e.g., GUI or NLP) that enables the transferring of user data to the feedback module and vice versa.

3.3 Process Views

3.3.1 Overview

Process views describe the flow of information across the different STAR modules. As already outlined, process views are provided in the context of specific use cases of the STAR systems, where a subset of STAR modules are used. In the following we illustrate process views for popular uses of the STAR modules, emphasizing on the information flows and the interactions across the modules. These popular use cases can be considered as blueprint functionalities of the STAR platform as they represent very common ways of using the platform in real-life industrial problems that demand trusted AI.

3.3.2 Defending a Poisoning Attack

One of the most popular cyber-defences for AI systems is the protection of Machine Learning systems against poisoning attacks (e.g., [Khurana19], [Chacon19]). The latter entail the task of polluting an ML model's training data towards compromising its ability to produce correct and credible outcomes (e.g., to classify instances correctly). One of the main objectives of STAR is to provide the means for defending against such attacks. Figure 15 illustrates the information flows of such a defence across STAR modules.

The process starts with the process of training or (re)training an AI model in the STAR ML and Analytics platform. The trained model is passed to the attack detection module of the ACDS for checking. Training data are stored in a data storage infrastructure, while the attack detection module communicates with two different modules of the STAR architecture, namely:

- The STAR Blockchain module that provides information for checking whether the training data have been tampered.
- The XAI module that provides an explanation of the model's functionality which is confronted against the expected functionality of the AI system. Specifically, XAI provides information on the relevant importance of the different features of the model following its training with the given data.

Based on the above checks, the attack detection module detects malformed instances and provides information on possible risks to the Risk Assessment module (which is part of the RAME).

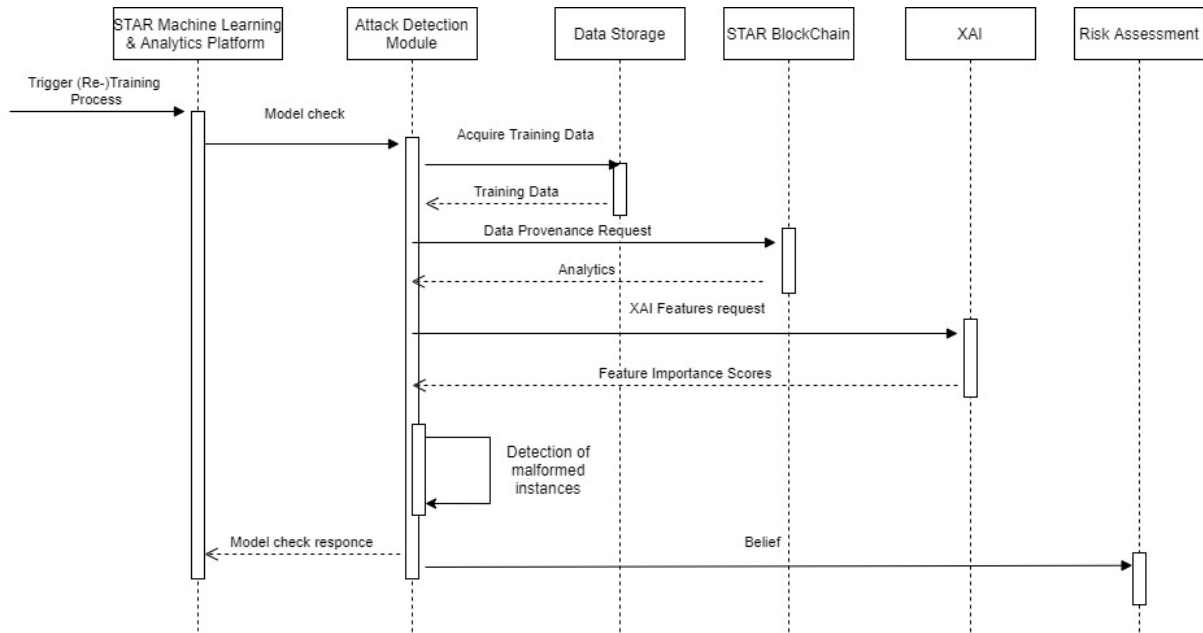


Figure 15: Information Flow for a Defending a Poisoning Attack

3.3.3 Defending an Evasion Attack

Figure 16 illustrates the process view of an evasion attack detection use case [Khorshidpour16]. Evasion attacks feed machine learning systems with adversarial examples i.e., selected perturbed inputs which resemble untampered copies and cannot be perceived by human users, yet cannot be classified correctly by the machine learning system.

The process leverages industrial data that stem from the shopfloor, which is validated and checked against two different modules of the STAR platform namely the blockchain-based DPT module and the XAI module. Specifically:

- The XAI module on the adversarial example is consulted to audit where the AI system behaves as expected. In this direction, the importance score of various features is provided from the XAI system.
- The blockchain-based DPT module is used to compare the potentially adversarial example against the statistical baseline of the data.

Based on this information, the attack detection module detects malformed instances and if needed alerts the risk assessment module. As a mitigation action and enhanced model for the AI system i.e. a model following adversarial training that mitigates susceptibility to the given example is provided to the STAR Machine Learning and Analytics platform. The new enhanced and secure/trusted model becomes in this way available for use in the manufacturing use case.

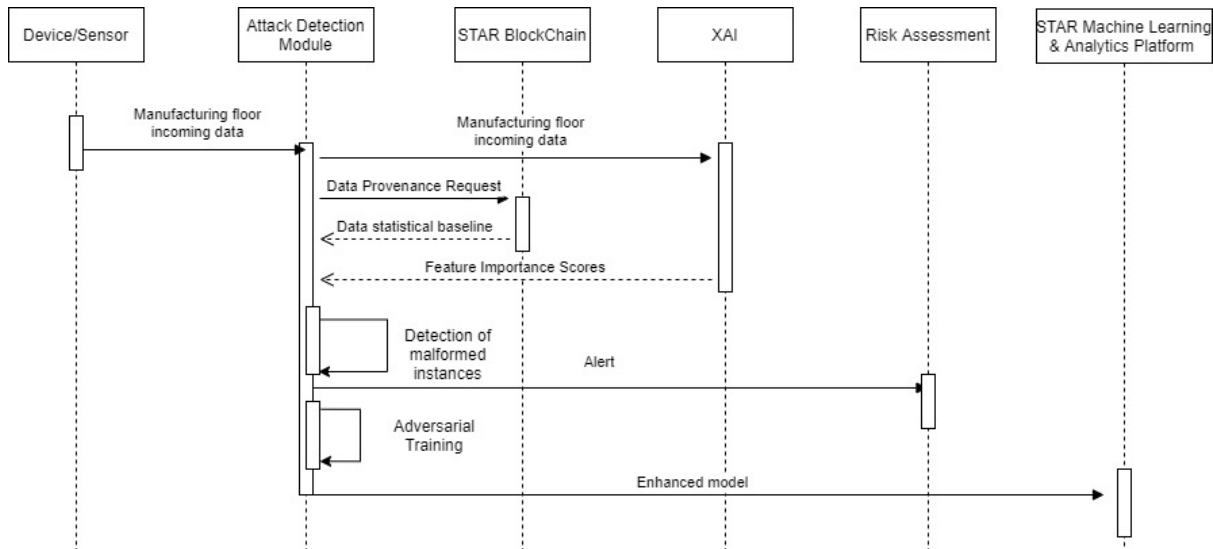


Figure 16: Information Flow for a Defending an Evasion Attack

3.3.4 Dynamic Management and Configuration of Data Sources

Figure 17 illustrates the process of introducing a new data source, which is specified through its metadata. The latter are defined based on a series of XML schemas (i.e. DK (Data Kind), DI (Data Interface), DSD (Data Source Definition)) that define the characteristics of a data source. These schemas are also used to specify and instantiate a probe that ensures access to the data of the source. Their detailed specification is beyond the scope of this deliverable. However, interested readers can consult [Soldatos21a], as well as STAR WP3 deliverables (e.g., deliverable D3.1 Decentralized Reliability for Industrial Data and Distributed Analytics-Initial version). The data source is registered with different registries like the registry of the distributed ledger and the registry of devices. These registries boost the dynamic operation and configuration of probes and data sources i.e. they enable the STAR system to cope dynamically with data sources that are added or removed from the system.

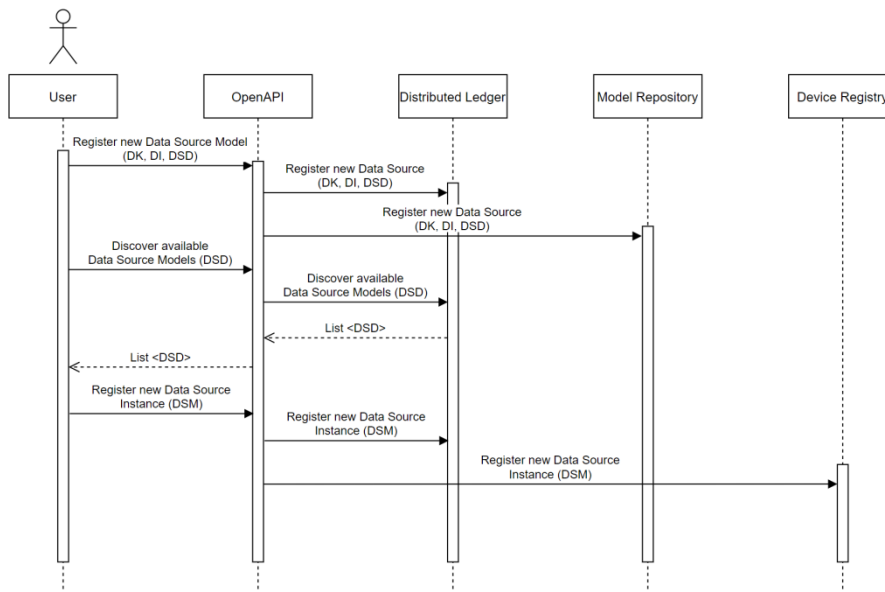


Figure 17: Process View of a Data Source Management Use Case

3.3.5 Security Policy Management

Figure 18 illustrates the process of specifying a security policy through the SPM component of the STAR architecture. The policy is specified considering probes and mitigation actions, which are structured into policies in the SPM and persisted in the security policies database. The latter policies are accordingly used to configure the Risk Assessment and Mitigation Engine (RAME).

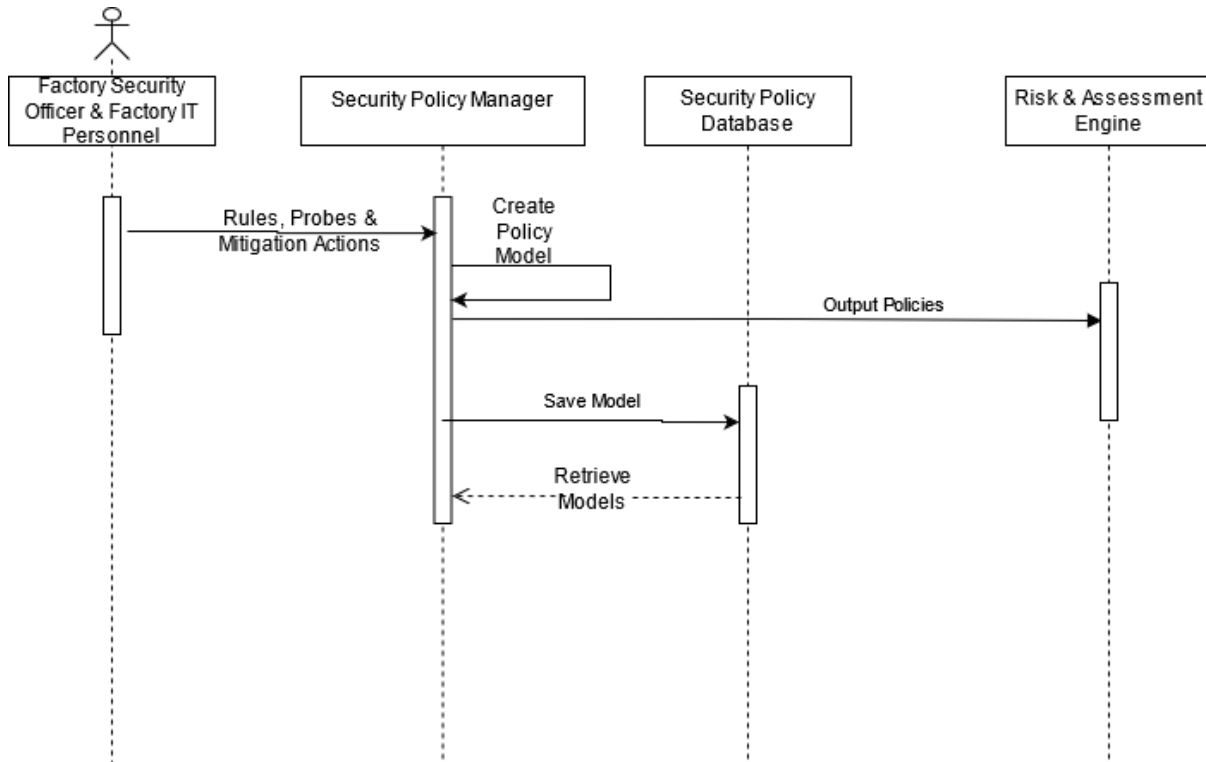


Figure 18: Process View of a Security Policy Management Use Case

3.3.6 Human Centric Digital Twin

The sequence diagram depicted in Figure 19 describes the high-level interactions between the HDT’s main components. We assume the existence of a Gateway, which collects data from different sensors, and an IIoT Middleware that supports a topic subscription mechanism.

The starting point is the user login: a user can log in to the systems through the Gateway. Right after the user logged in, the Gateway notifies the system Orchestrator, which in turn issues an "establish connection" request to both the IIoT Middleware and the Function Module components. The IIoT Middleware thus establishes a connection with the Gateway, waiting for new data coming from the sensors.

At the same time, the Functional Module component establishes a connection with the IIoT Middleware. Moreover, the Functional Module issues a new request to the Orchestrator to know which topics contain the information needed to run its internal logic. As a response, the Orchestrator returns a map <parameter: topic>, e.g., <heart_rate: HRtopic>, which instructs the Functional Module about the topics relevant to the logged user. Finally, the Functional Module subscribes to all the topics and waits for new messages.

The Gateway writes new messages on relevant topics every time it collects a certain amount of data (e.g., every 10 minutes of data collection). The IIoT Middleware automatically forwards the messages to both the Functional Module and the Time Series Storage. For each received message, the Functional Module executes its internal logic and publishes a message to the IIoT Middleware. Eventually, the message can be forwarded by the IIoT Middleware to the Gateway to notify the user about a particular event. Note that the IIoT middleware may comprise the STAR cyber-security and cyber-defence functions to ensure data reliability and security.

In some cases, the Functional Module may need additional data to execute its internal logic. The Functional Module can access additional data by querying the Time Series Storage (for historical dynamic data), or the Orchestrator (for quasi-static data). The system runs until the user logs out: at this point, the Orchestrator notifies all components to unsubscribe topics and close any active connections.

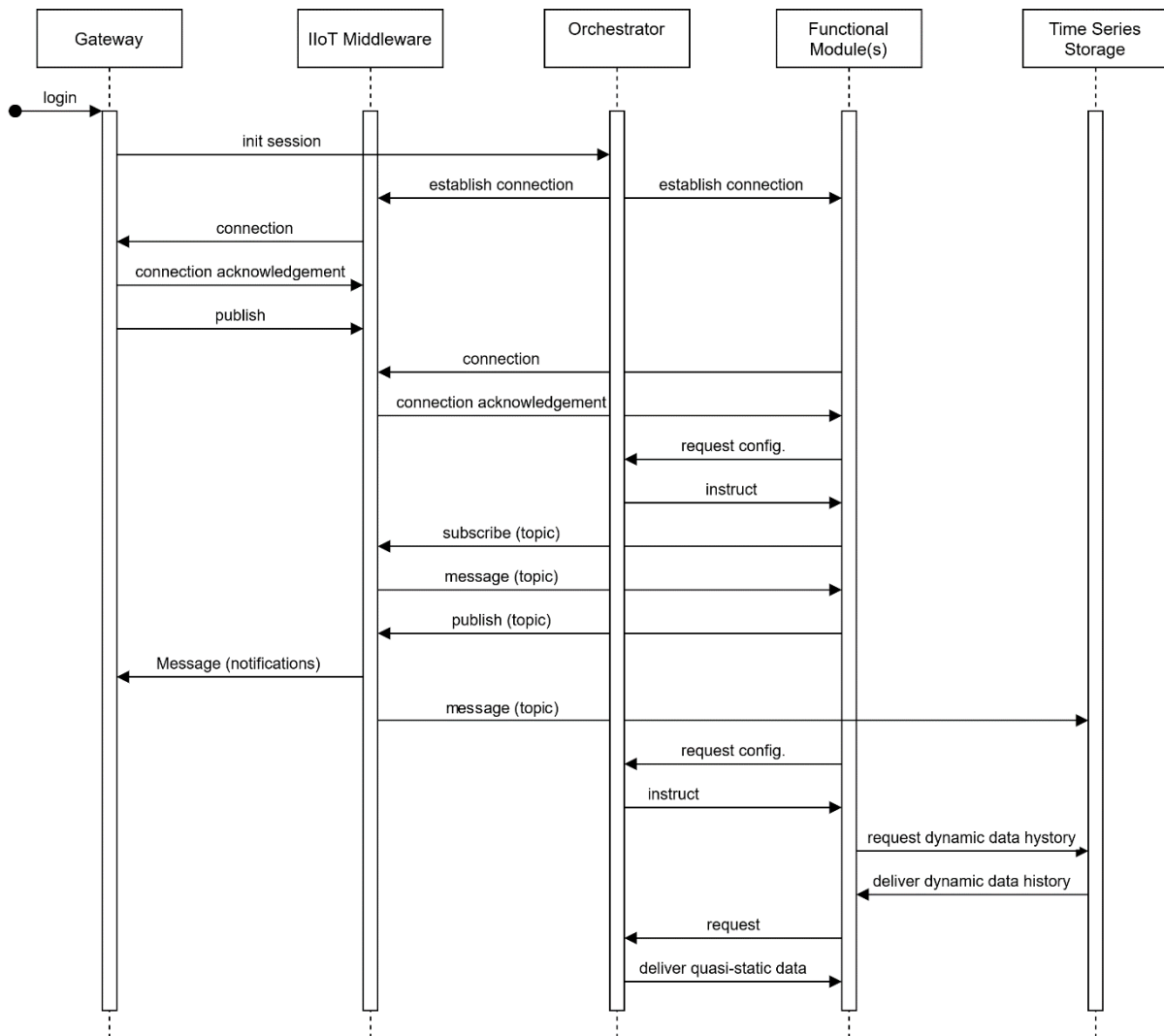


Figure 19: Process View of the Human Centric Digital Twin Operation

3.3.7 STAR XAI Models and Library Operations

The following UML sequence diagrams illustrate two of the main “internal” operations of the XAI module, namely the execution of counterfactual logic [Stepin21] and the ranking of features based on their relevant importance in the decisions of the AI model. Specifically, Figure 20 illustrates how STAR XAI will produce counterfactual logic, while Figure 21 presents the features ranking operations. These are two very common operations for XAI systems, which are used by other STAR modules (e.g., the ACDS uses features ranking information in the detection of attacks). This is the reason why they are herewith presented as blueprints for STAR XAI operations.

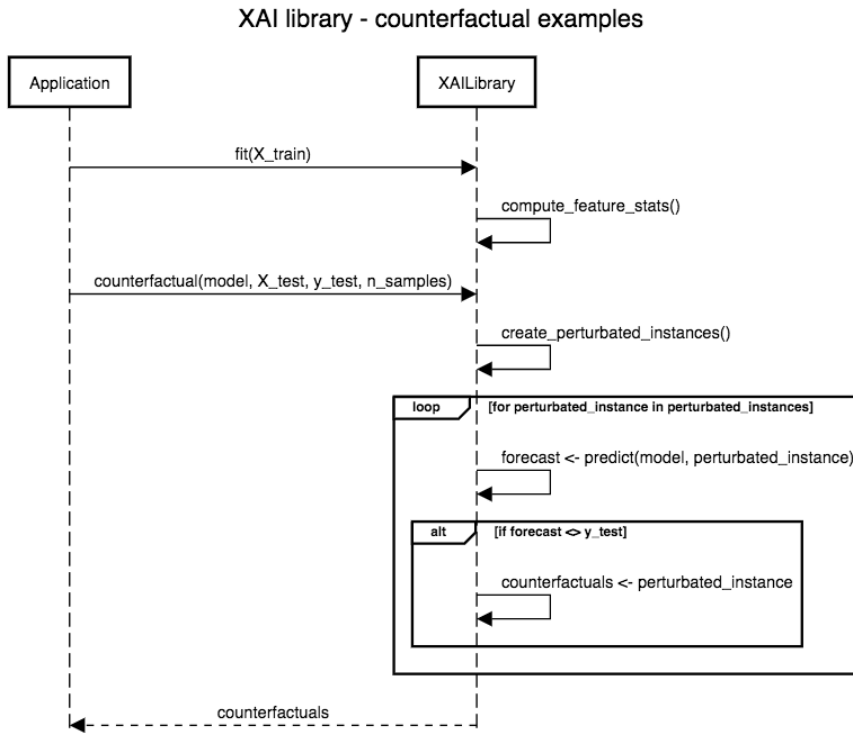


Figure 20: Provision of Counterfactuals Information by the STAR XAI

XAI library - features ranking

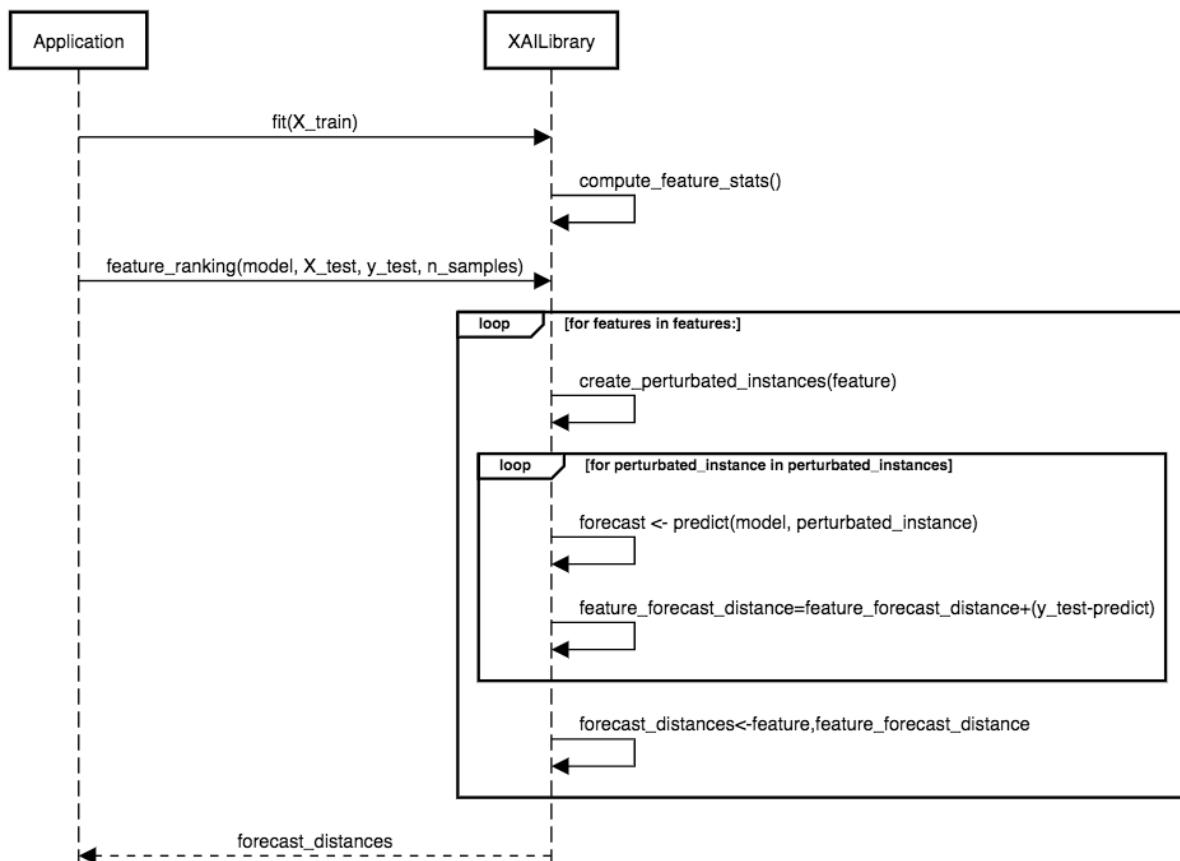


Figure 21: Features Ranking Operations by the STAR AI

A typical workflow for the operation of the XAI systems in terms of feature scoring is as follows:

- Data acquisition and input.
- Machine Learning models.
- XAI methods utilization.
- Calculation of feature importance scores.
- Visualization of results.
- Presentation of results to end users and domain experts.

3.3.8 Active Learning for Human Robot Interactions

Figure 22 illustrates the blueprint for the operation of the AL module in scenarios involving human robot interactions. It illustrates the interaction of the user and data experts with the AI application and the AL model. The DB (Database Module) in the figure can be part of the Production Processes Knowledge Base (PPKB).

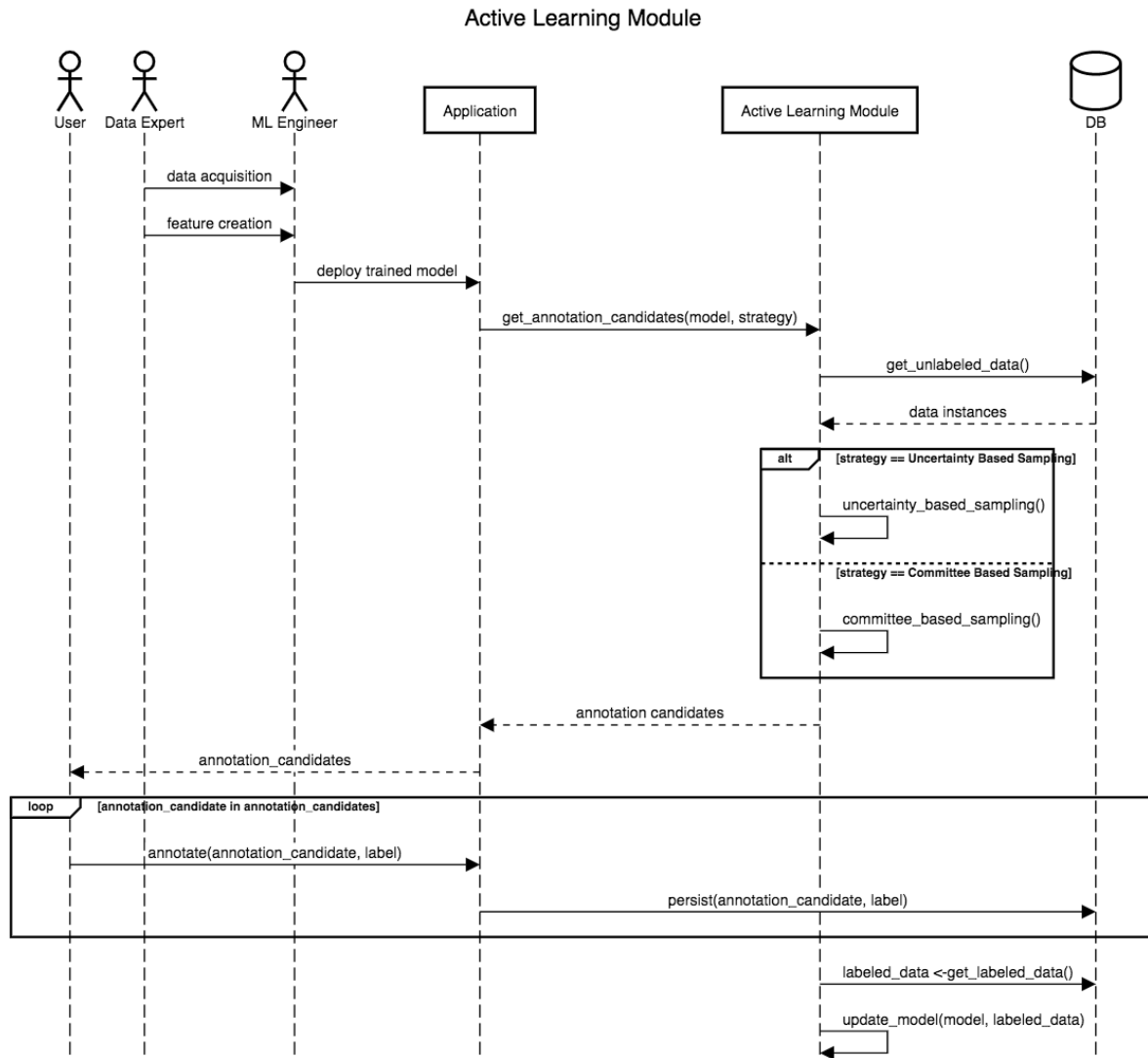


Figure 22: Active Learning Module Operation in support of Human Robot Interaction

3.3.9 Feedback Module Operation

The UML diagram of Figure 23 presents the operation of the feedback module, including interactions between end-user, analyst and the Industry 4.0 application. The user is provided with feedback options and provides his/her feedback, while the analyst analyses the feedback and persists new/improved options in a database.

Feedback Module

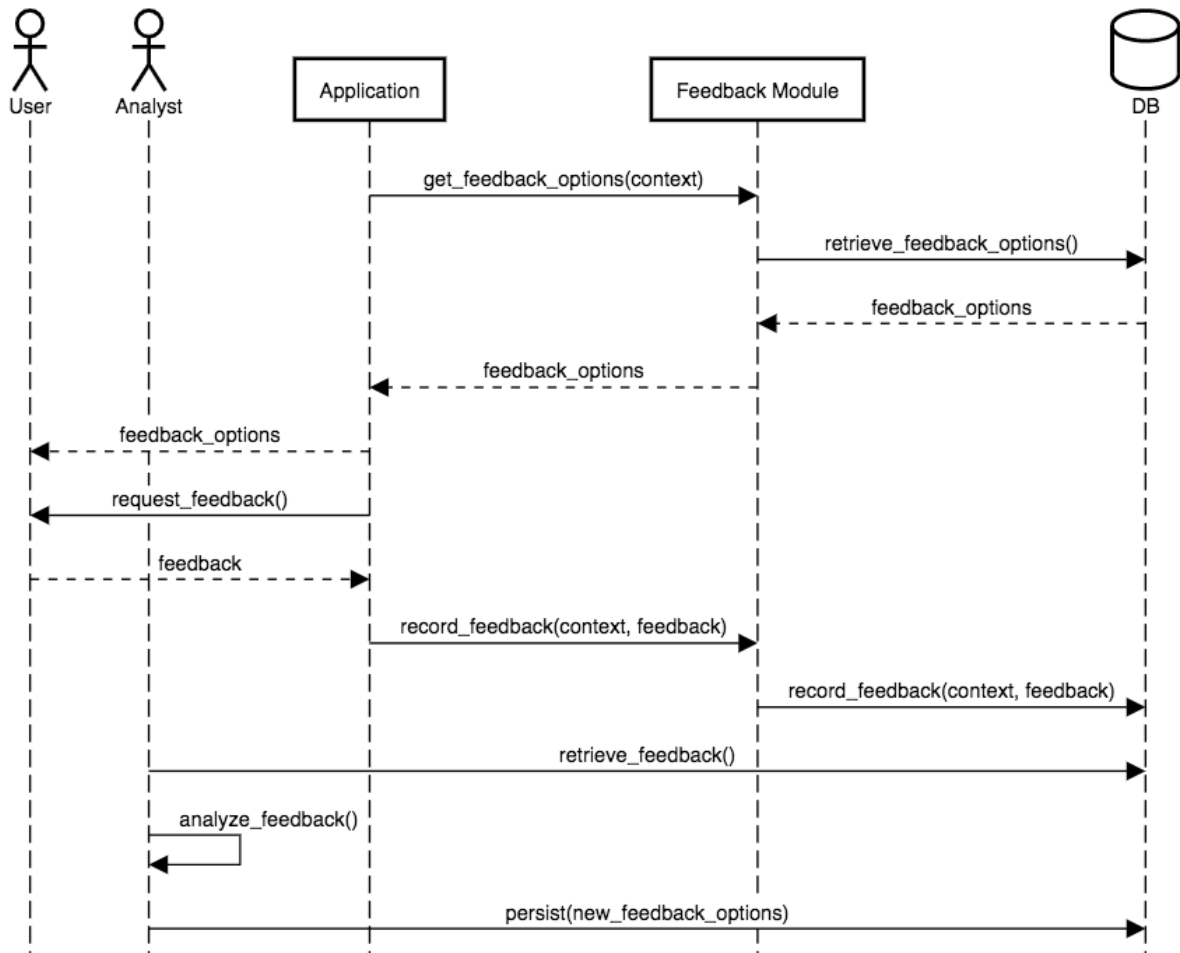


Figure 23: Operation of the Feedback Module

3.3.10 NLP Interaction

The NLP module provides one of the most user-friendly way for interacting with AI systems and robots. Its operation is illustrated at a high level in Figure 24. The diagram illustrates how the NLP module provides feedback to JSI’s CuriousCat system, which will support the implementation of the AL system.

STT, TTS, SA:
Preliminary Diagram

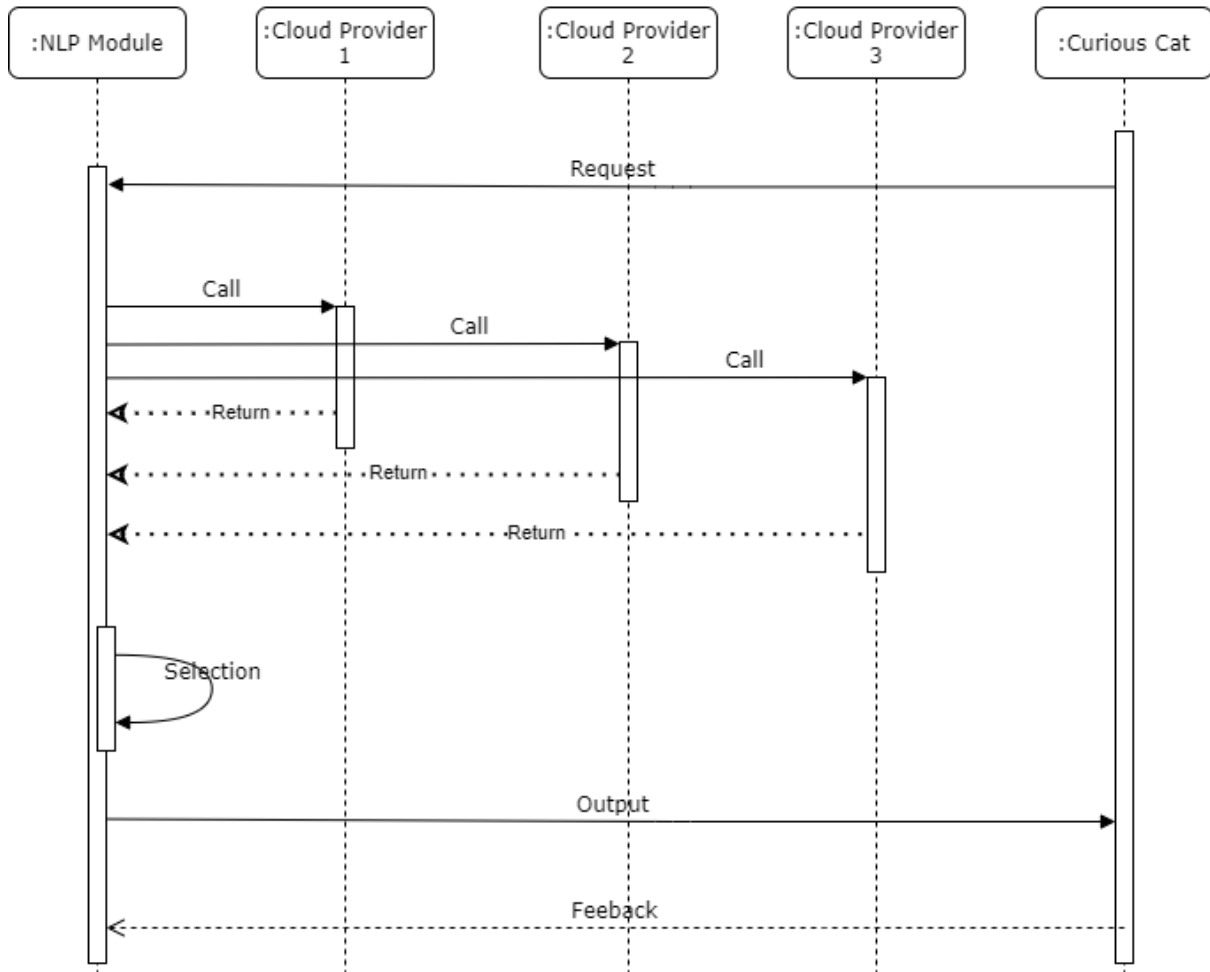


Figure 24: High Level View of the NLP Operation in the context of the STAR Implementation

3.4 Physical View Considerations

3.4.1 Deployment Overview

The physical deployment of the STAR system will be based on the cloud/edge deployment paradigm. In principle, low-latency operations that require real-time performance will be deployed at the edge, while operations requiring more data points will be carried out in the cloud. This is in-line with the IIRA deployment view. Table 1 illustrates some considerations that drive the selection of physical deployment of certain components to the cloud, the edge (e.g., an edge cluster or gateway) or even the faredge (e.g., an embedded device or machinery if applicable).

Table 1: Guide for Industrial Deployments at the Cloud/Edge/FarEdge

Features	Cloud	Edge	FarEdge
Data Points Availability	High	Medium	Low

Energy Efficiency	Low	Medium-High	Very High
Privacy	Low-Medium	Medium-High	High
Low Latency / Real-Time Performance	Low	Medium-High	Very High

Table 2 presents initial edge/cloud deployment considerations for the main components of the STAR architecture. Specifically, illustrates which components are prioritization for deployment at the edge and which ones at the cloud. Moreover, there are components labeled edge/cloud, which means that they will be deployed either in the edge or the cloud depending on the use case at hand.

Table 2: Edge/Cloud Deployment Considerations for the main components of the STAR architecture

Component Name	Physical Deployment Choice
Data Probes / Data Connectors	Cloud/Edge, specifically: Cloud: Monitoring Engine; Edge: Data Collectors (Beats)
STAR Blockchain (DLT)	Cloud
AI Cyber Defense Strategies	Cloud
Risk Assessment and Mitigation Engine (RAME)	Cloud
Security Policies Manager (SPM)	Cloud/Edge, specifically: Cloud: Policy Management Engine, Policy Validation; Edge: Policy enforcement, Policy Validation
XAI Library	Cloud/Edge
Simulated Reality	Cloud/Edge
Active Learning (AL)	Cloud
NLP Module (incl. TTS, STT, Sentiment Analysis)	Cloud
Production Processes Knowledge Base	Cloud
Feedback Module	Cloud

3.4.2 Deployment & Ecosystem Management Technologies

In this section we provide the tools and platforms which are proposed to be used in STAR project, based on well accepted best practices and deployment trends, in order to provide the packaging and integration of the offered components in the scope of trusted AI technologies for production lines and manufacturing use cases.

3.4.2.1 Software Packaging with Docker images.

For the STAR software packaging, we have considered Docker³ images which is currently the dominant technology/methodology and is considered a de facto. A Docker image is a file, comprised of multiple layers, used to execute code in a Docker container. An image is essentially built from the instructions for a complete and executable version of an application, which relies on the host OS kernel. In the following sections (wherever relevant i.e. stack management, monitoring tools etc.) we are only considering tools that are compliant with the Docker Platform.

Docker is an open platform for developing, shipping, and running applications. With Docker, an infrastructure can be managed in the same ways applications are managed. Docker offers shipping, testing, and deploying methodologies easily and quickly, where the time between writing code and running it in production can be significantly reduced.

Docker provides the ability to package and run an application in a loosely isolated environment called a container. The isolation and security allow you to run many containers simultaneously on a given host. Containers are lightweight because they don't need the extra load of a hypervisor but run directly within the host machine's kernel. This means you can run more containers on a given hardware combination than if you were using virtual machines. You can even run Docker containers within host machines that are actual virtual machines [Docker].

There are many tutorials in order to containerize an application or a system and offer it thru a repository management service which spans from beginners to more advanced ones depending on the technologies used. An intermediate one that doesn't focus on a specific technology and provides the relevant aspects that are necessary to establish a well-defined contract between Dev and Ops teams can be found in [Souza18], which provides an article on how to "dockerize" any application. The article provides a 10-steps checklist which includes the:

- Choice of a base Image.
- Installation of the necessary packages.
- Addition of custom files.
- Definition of users that will run your container.
- Definition of the exposed ports.
- Definition of the entry point.
- Definition of the configuration method.
- Externalization of the data.
- Logs handling.
- Logs rotation and other append-only files.

3.4.2.2 Container Tool with Docker Compose.

As briefly mentioned above Docker Compose is a tool for defining and running multi-container Docker applications. It uses YAML files to configure the application's services and performs the creation and start-up process of all the containers with a single command. The *docker-compose.yml* file is used to define an application's services and includes configuration options. In STAR as the preferred container runtime management method was Docker Compose every component will be accompanied by a *docker-compose.yml* file which

³ <https://docs.docker.com/>

will facilitate its installation. Additionally, different collections of interoperable components that will be used as solutions for the STAR use cases will be provided as ready-to-install *docker-compose.yml* files.

Information on how to edit a *docker-compose.yml* file can be found at Docker Docs [Docker] and more specifically at the Get started with Docker Compose⁴.

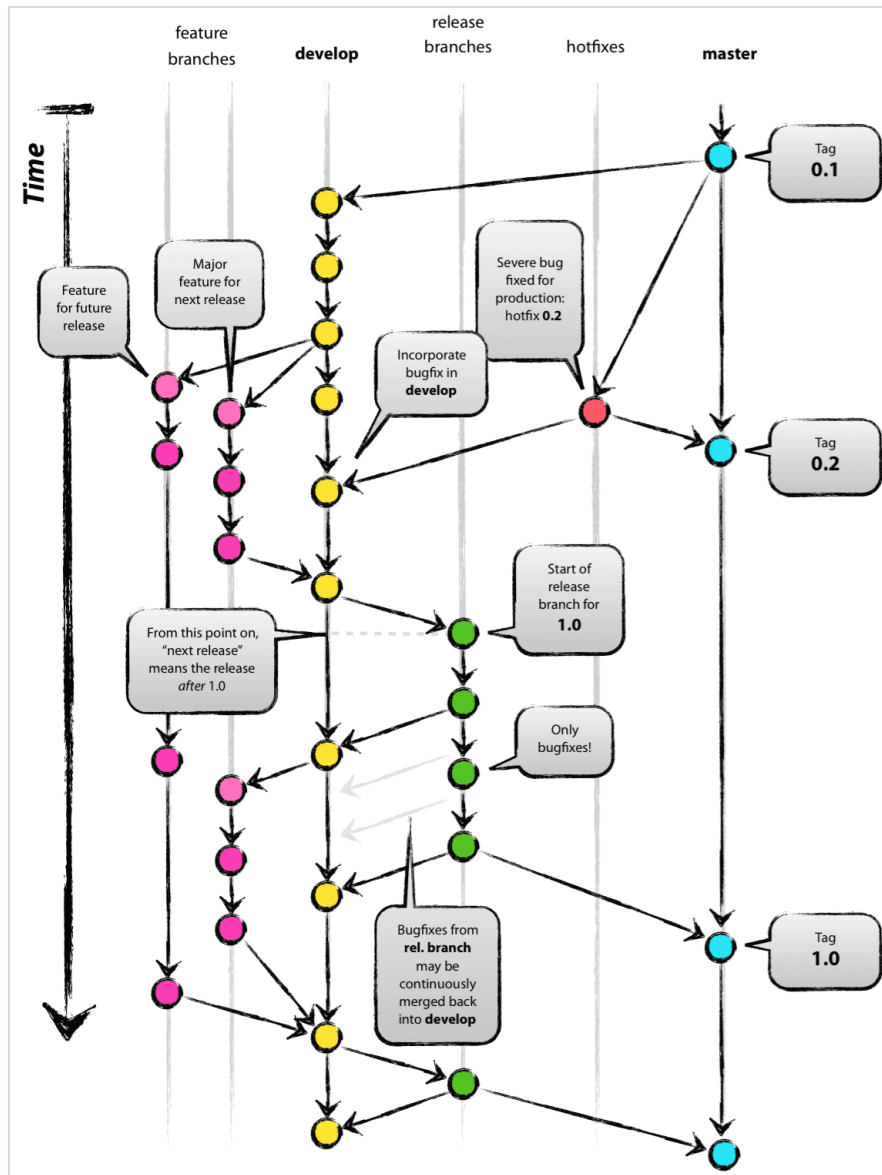


Figure 25 A Complete Git branching model⁵

3.4.2.3 Code Management with GitLab.

STAR will facilitate the component's code management by using two immensely popular open-source technologies, Git and GitLab⁶. Git serves as the Version Control Systems (VCS), while GitLab is a powerful and intuitive Git repository hosting service. The latter offers a web-based graphical interface with several built-in features. It allows the creation of

⁴ <https://docs.docker.com/compose/gettingstarted/>

⁵ <https://nvie.com/posts/a-successful-git-branching-model/>

⁶ <https://gitlab.com/>

collaboratively owned and maintained code repositories, code branching and merging, version control, issue tracking, code review, wikis, etc. Multiple developers can concurrently create, merge and delete parts of the code they are working on independently at their local system, before pushing the changes back to branches of the shared GitLab repository. The instantiated STAR GitLab group can be found under the following URL: <https://gitlab.com/star-ai>

STAR could use the branching model (or part of it) proposed by Mr. Vincent Driessen "A successful Git branching model" and a complete version of which is shown in Figure 25.

In cases where existing components (available in other GitLab branches) are used, repository mirroring mechanisms may be employed to ensure access to the components from the project's GitHub.

3.4.2.4 Container Repository & Registry Management

A container registry is a catalogue of storage locations where one can push and pull container images. However, the actual physical locations where images are stored are known as repositories. Each repository stores a collection of related images with the same name. Each image within a repository represents a different version of the same container deployment. A specific image is identified by either its tag or its own unique reference [Kisller21].

For the STAR project, JFrog Container Registry has been selected to be used to setup a secure private Docker Registry. The JFrog Container Registry supports Docker and Helm registries and Generic repositories, allowing users to build, deploy and manage container images while providing powerful features with fine-grained permission control behind a sleek and easy-to-use UI. JFrog Container Registry imposes no limitations on the number of Docker Registries one may apply and hosts two kinds of repositories: i) local repositories and ii) remote repositories.

Both local and remote repositories can be aggregated under virtual repositories to create controlled domains for artifact resolution and search. Local repositories are physical locally managed repositories where one can deploy artifacts to. Remote repositories are served as a caching proxy for a repository managed at a remote URL. Virtual repositories aggregate several repositories under a common URL. The repository is virtual in the sense that one can resolve and gets artifacts from it, however, they cannot deploy anything on it.

3.4.2.5 Management/Monitoring with Portainer.

Since the preferred deployment strategy is the docker containerization in order to facilitate the ecosystem management and monitoring there are various offerings one of which is the Community Edition (CE) of Portainer⁷.

Portainer CE is a lightweight management toolset that allows you to easily build, manage and maintain Docker environments. Portainer offers a GUI (Graphical User Interface) which alleviates the complexity of using CLI (Command Line Input) commands. Portainer offers the following features which may be used over the STAR deployments:

- UI that covers all of essential docker CLI actions
- Enhanced functions, not available from the command line

⁷ <https://www.portainer.io/products-services/portainer-community-edition/>

- Expert configuration built into the software
 - Including pre-validation checks for complex deployments
- Management of access control and LDAP authentication
- Aggregation view of swarm clusters
- Log viewer
- Remote console with process performance viewer

Directions on how the technology providers can install Portainer environment in a local Docker instance can be found at the Portainer's Deployment⁸ documentation. General documentation along with user and configuration guides can be found in Portainer's Documentation⁹.

3.4.2.6 Access Control

In order to offer secure access to the infrastructure and more specifically for the platforms and services that do not implement authentication the option of a SSO (Single Sign On) identity and access management can be offered. One of the Most commonly used Open-Source identity and access management software is Keycloak¹⁰ . Some of the Key features of Keycloak are that it:

- Provides Single-Sign On functionality,
- Offers standard protocols like OpenID connect, OAuth 2.0 and SAML 2.0,
- Offers centralized management,
- Offers adapters for applications and services,
- Provides LDAP and active directory to connect existing user directories.

Directions on how to install Keycloak using docker can be found at the Keycloak getting started Docker page¹¹. Moreover, all the information related to the Keycloak functionalities, deployment and usage can be found at the Keycloak's documentation¹².

3.5 Implementation View Considerations

3.5.1 Implementation Overview

The implementation of the STAR architecture requires the development and integration of a very wide range of components from different disciplines, including distributed ledger technologies, AI and robotics systems, machine learning technologies (including deep learning and reinforcement learning), NLP systems, advanced AI systems like Active Learning, Digital Twins and more. These components create a very diverse and heterogeneous technological landscape: The components of the STAR architecture cannot be implemented in a single platform. Rather different platforms will be employed and integrated based on modern integration infrastructures such as containers and microservices.

⁸ <https://portainer.readthedocs.io/en/stable/deployment.html>

⁹ <https://portainer.readthedocs.io/en/stable/#>

¹⁰ <https://www.keycloak.org/>

¹¹ <https://www.keycloak.org/getting-started/getting-started-docker>

¹² <https://www.keycloak.org/documentation>

3.5.2 Implementation Technologies

The implementation of the STAR architecture is on-going and will be reported in the second version of this deliverable, namely deliverable D2.7. Nevertheless, Table 3 provides an overview of the implementation platforms and other considerations for the primary components of the STAR platform. Specifically, for various components of the architecture, it outlines its implementation platform and other details.

Table 3: Envisaged Implementation Technologies

Component Name	Platform	Programmin g Language	Interfaces to other Modules	Database
Data Probes / Data Connectors	ElasticStack	Java	REST API Apache Kafka Pub/Sub	Elastic Stack (ELK)/Mongo DB
STAR Blockchain (DLT)	Hyperledge Fabric	Java	REST API	MySQL/Coach DB
AI Cyber Defense Strategies	Adversarial Robustness Toolbox (ART) ¹³	Python	REST API	MySQL
Risk Assessment and Mitigation Engine (RAME) ¹⁴	Olistic.io / Spring Framework	Java	REST API Apache Kafka Pub/Sub	MySQL/Mongo DB
Security Policies Manager (SPM)	Xacml4j		REST API	
XAI Library		Python	REST API	
Simulated Reality		TensorFlow and/or PyTorch	REST API	
Active Learning (AL)	Linux	Python	REST API	
NLP Module (incl. TTS, STT, Sentiment Analysis)	Linux + Cloud Services for 3rd party providers	Python	REST API	
Production Processes Knowledge Base	Linux	Python	REST API	noSQL
Feedback Module	Linux	Python	REST API	noSQL

¹³ <https://adversarial-robustness-toolbox.org/>

¹⁴ The implementation with leverage Ubitech’s Olistic.io product

4 STAR Architecture Validation

As an early validation of the various modules of the architecture, we herewith present the mapping of some of the project’s manufacturing use cases to architecture elements. Note that these use cases are associated with the manufacturing functionalities to be offered by the project, rather than with the fulfilment of non-functional requirements like cyber-defence requirements addressed in the use cases that are presented in the previous section.

4.1 Human Intention Recognition

This use case focuses on the prediction of human behaviour to configure the mobile robot later to avoid possible collisions, as well as to create safety zones. It will be realized at DFKI’s SmartFactory testlab, in Kaiserslautern, where the latest developments in the industry are tested. The use case will convert algorithm-based human activity into an AI-approach and predict their next actions based on their daily activities. The goal of the use case is to keep the production level high, maintaining human’s safety during robot activity.

Although the design of the use-case is still under development, several components from the STAR Reference Architecture are planned to be used. Human-Centered Digital Twin (HCDDT) will be used to virtualize the activities before applying them in the real demonstrator. For the use case, a Worker’s Intention Recognition Module (WIRM) is also going to be developed by DFKI. The WIRM module and Fatigue Monitoring System will be integrated into HCDDT to be later used together with Active Learning.

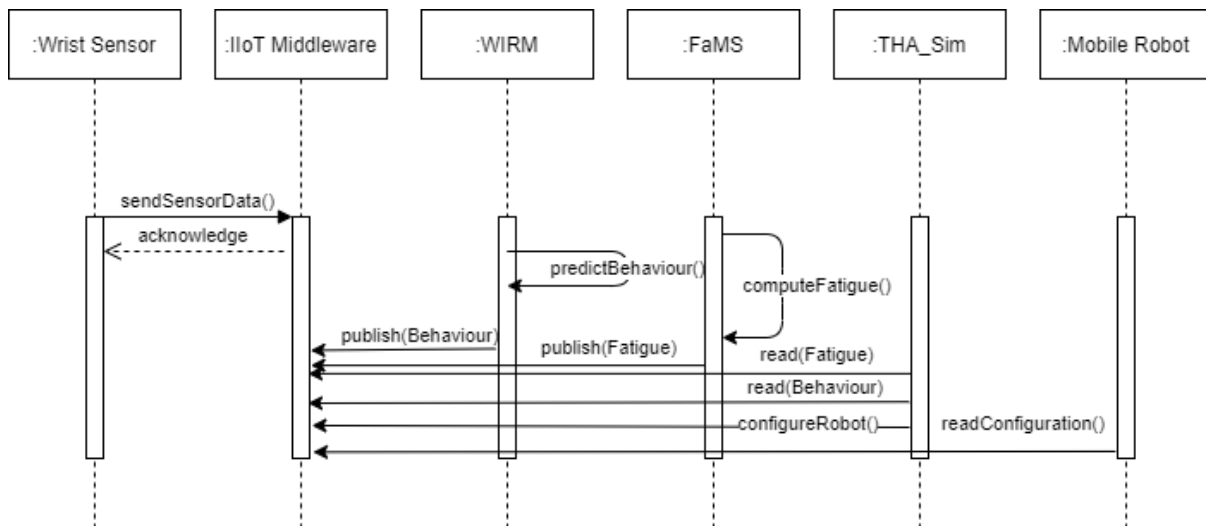


Figure 26: Preliminary sequence diagram of Human Intention Recognition Use Case

An initial diagram illustrated in Figure 26, which shows how the use case will interact with other components to achieve its goals. First, the data from the wrist sensors will be collected. This data will be read by WIRM and the current activity of the human will be detected. Based on the current activity, next activity will be predicted. By combining the tiredness possibility from the FaMS, the simulation software from THALES will create possible collision points for the robot. Based on these points, the robot will be configured. This use case will be completed after human can be taken into consideration. The following

use case will take the responsibility from this point. Note that the IoT Middleware element in the diagram denotes a module that comprises the security and cyber-defence modules of the STAR RA.

4.2 Robot reconfiguration based on dynamic factory layout.

This use case deals with the mobile robot at DFKI to reach its destination without reducing the production rate, by considering the human and equipment safety at the vicinity. Currently, the robot is used to scan the whole environment manually, to let it detect the obstacles and the boundaries of the room. Later, these obstacles are registered using its proprietary tool and new path is drawn to allow it to reach its destination. This procedure is cumbersome if the layout dynamically changes, and/or robot moves between zones where a high movement activity is foreseen. After the new collision points for the robot has been created, it will be necessary to create new route for the robot to reach its estimation.

By realizing this use case, the layout changes will actively be monitored by cameras and path planning will be automated using AI approach(es). Initially, the use case will use the Simulated Reality component to test the solution before realization. Later, Reinforcement Learning (RL) Systems as well as Autonomous Mobile Robot (AMR) Safety components are going to be utilized to combine the data to be later sent to the robot as commands. This use case also requires the input received from the previous use case, since not only the current action of the human is required, but also the possible next actions.

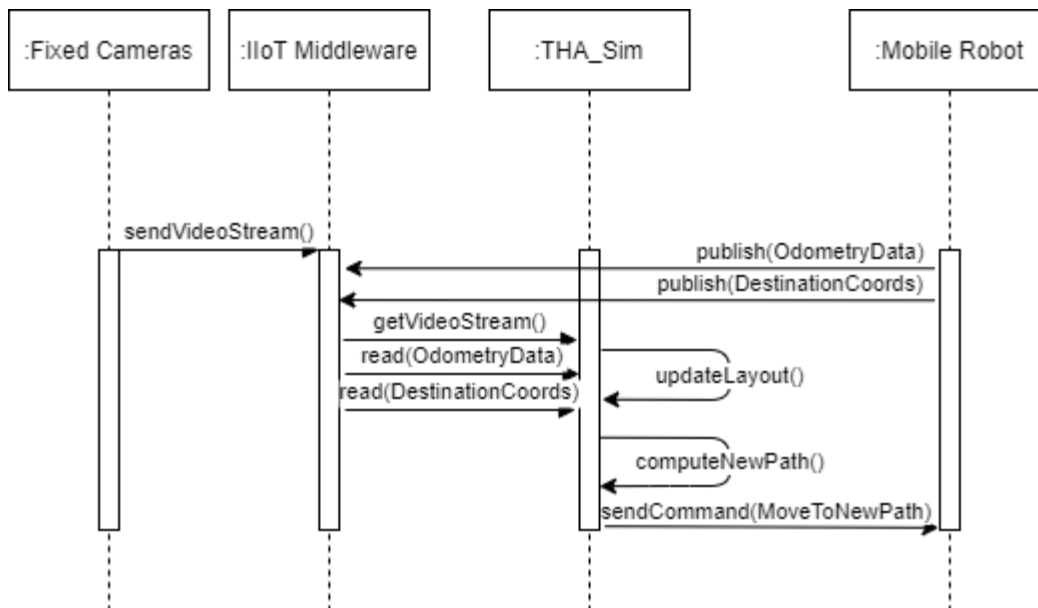


Figure 27: Preliminary sequence diagram of Robot Reconfiguration Use Case

Figure 27 illustrates the initial interactions between the components. First, the cameras will stream video to the software from THALES. At the same time, the mobile robot is expected to send its odometry data to inform about its current coordinates, direction, as well as its speed. Later, the destination coordinates from the robot will be published. The software from THALES will update the layout with the information gathered from other sources and create new path(s) for the robot to reach its destination. The destination may contain one or more waypoints to reach, in advance. All this data will be sent as a command to the robot.

If no possible path is foreseen, the robot will need to be stopped. This can also be done with the command directly sent to the robot.

Similar to the previous use case, the IIoT middleware comprises the secure and reliable data management functionality of the STAR RA, which are implemented based on cybersecurity and blockchain technologies, as illustrates in the following sections.

Based on the training data, the safety zones for the mobile robot will be defined for a faster response time in the future.

4.3 Safe Human Robot Collaboration

This use-case is focused on the human-robot collaboration and is developed within the Philips factory in Drachten, an advanced factory for mass manufacturing of consumer goods in Europe. The main objective of the use-case is to leverage the best of both humans and machines. This means that both human- and machine-driven processes can take place in flexible production cells, where tasks can be assigned to either a cobot or a human depending on their complexity and requirements. In this way, the workload can be optimised and shared between humans and robots. Moreover, this use-case employs Human Digital Twins (HDTs) as models to observe human workers from a quantifiable and measurable perspective within the production process.

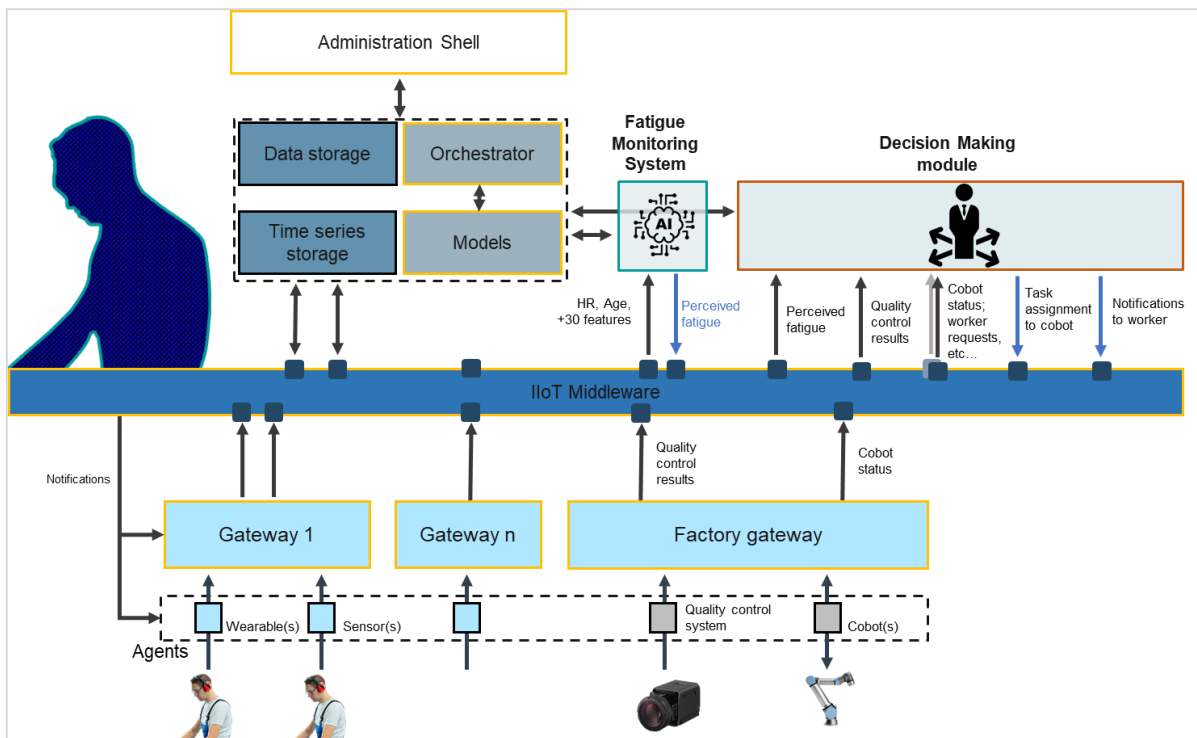


Figure 28 HDT architecture instance for the Safe collaboration between human and cobot¹⁵

To realise this mutualism, a HDT is deployed in the Philips work cell to create a digital representation of the worker and the contextual elements of the factory. The HDT describes workers and contextual elements by ingesting data gathered from sensors (e.g., wearable

¹⁵ Subset of earlier provided Figure 14

devices, machines) and software components (e.g., the Fatigue Monitoring System). The creation of this digital replica supports the decision-making and tasks assignment, managed through a Functional Module specifically developed. Moreover, the HDT can be consulted through specific GUIs to allow the analysis of the data and support human decision-makers to take decisions towards process continuous improvement.

Figure 28 presents the HDT architecture instance for the described use-case. Since the use-case design is still ongoing, a few modifications may be needed in the future, mainly depending on new emerging needs (e.g., new variables and parameters) to support the decision-making process.

Figure 29 exemplifies a decision-making process within the “safe collaboration between human and cobot” pilot, and shows how the different main components of the HDT interact with each other. The decision process is triggered by the Decision-Making module (DM module) from time to time. The module requests the decision parameters (param[]) to analyse the work cell’s current situation and determine the best new configuration, as well as the actions to be triggered. The request is forwarded to the Orchestrator, which triggers FaMS to calculate the perceived fatigue (fatigueLv) of the enrolled worker(s). FaMS needs to be instructed by the Orchestrator to find the topics (topics[]) where the required fatigue calculation features are stored. In this way, FaMS can fetch the data from the topics of interest, compute the fatigueLv value and publish it on the IIoT Middleware. Then, the Orchestrator instructs the DM module about the topics containing the required decision parameters. The parameters are finally processed by DM to decide the best configuration to implement (configuration: task assigned to cobot, task assigned to the worker) and which messages to deliver to the workers.

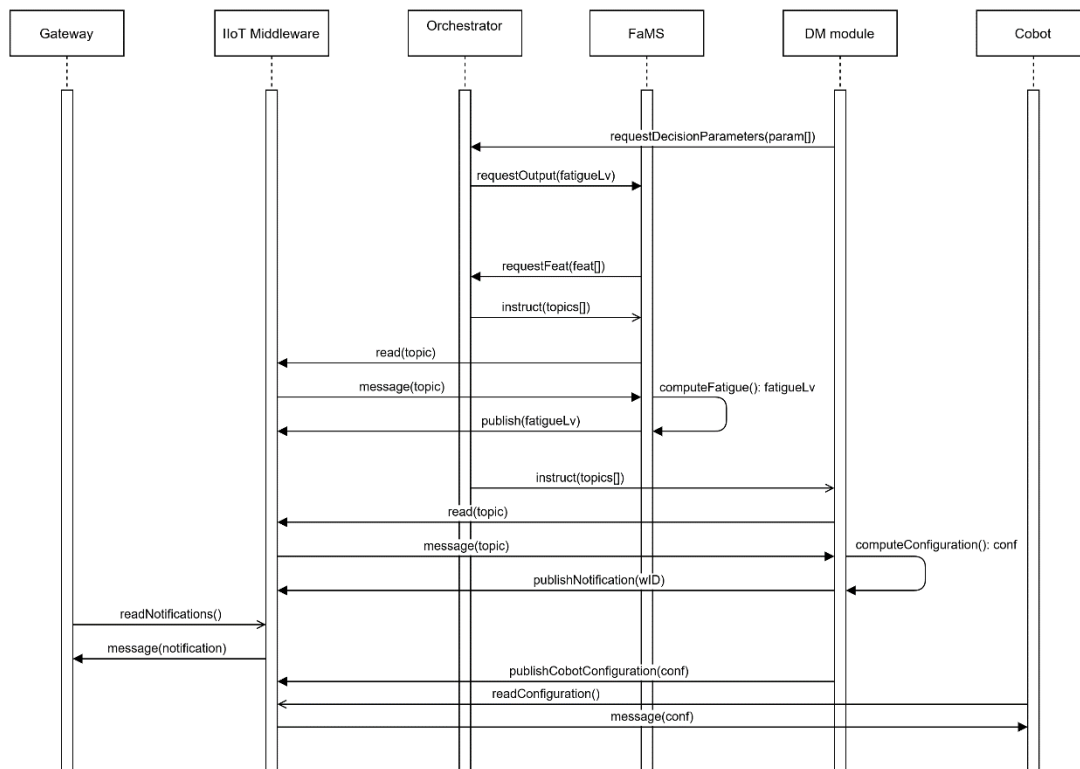


Figure 29 Safe collaboration between human and cobot: sequence diagram

5 Blueprints Specification

5.1 Introduction

Previous sections have presented how the introduced STAR architecture supports the implementation of popular secure and trustworthy data-driven use cases in industrial environments. These popular use cases form the initial sets of the project’s blueprints, which are specified in the following paragraphs, yet sequence diagrams for them have been presented in previous paragraphs. Moreover, we specify an additional set of use cases that will be mapped to the final version of STAR RA as part of the second version of the present deliverable.

5.2 Initial List of STAR Blueprints

5.2.1 Defending a Poisoning Attack

The following table illustrates the defending poisoning attack blueprint.

Table 4: STAR-BLPR-1 - Poisoning Attack Defence

Code	STAR-BLPR-1
Title	Poisoning Attack Defence
Scope/Purpose	Detect with High Accuracy a Poisoning attack against an AI/ML system i.e. cases where an attacker compromises the learning process based on adversarial examples, in ways that compromise the AI systems ability to produce correct/credible results.
STAR Components Involved	STAR Analytics Platform, Data Provenance & Traceability, STAR Blockchain, Risk Assessment and Mitigation Engine, XAI Module
UML Diagram	See Figure 15

5.2.2 Defending an Evasion Attack

The following table illustrates the defending evasion attack blueprint.

Table 5: STAR-BLPR-2 - Evasion Attack Defence

Code	STAR-BLPR-2
Title	Evasion Attack Defence
Scope/Purpose	Detect with High Accuracy an Evasion attack against an AI/ML system i.e. cases where an adversary alters input examples in directions that result in slightly different inputs that cannot be handled correctly by the AI system
STAR Components Involved	STAR Analytics Platform, Data Provenance & Traceability, STAR Blockchain, Risk Assessment and Mitigation Engine, XAI Module
UML Diagram	See Figure 16

5.2.3 Management and Configuration of Data Sources

The following table illustrates the management and configuration of data sources blueprint.

Table 6: STAR-BLPR-3 – Management and Configuration of Industrial Data Sources

Code	STAR-BLPR-3
Title	Management and Configuration of Data Sources
Scope/Purpose	Configure a source of data in the manufacturing shopfloor to take advantage of the STAR system and its trusted AI capabilities.
STAR Components Involved	Probes Registry, Data Models, Data Provenance & Traceability, STAR Blockchain (Distributed Ledger)
UML Diagram	See Figure 17

5.2.4 Security Policy Management

The following table illustrates the security policy management blueprint.

Table 7: STAR-BLPR-4 – Security Policy Management

Code	STAR-BLPR-4
Title	Security Policy Management
Scope/Purpose	Specify, Manage and Enforce policies regarding the operation of STAR AI systems
STAR Components Involved	Security Policy Management, Risk Assessment and Mitigation Engine
UML Diagram	See Figure 18

5.2.5 Human Centred Digital Twin

The following table illustrates the HDT blueprint.

Table 8: STAR-BLPR-5 – Human Centred Digital Twin

Code	STAR-BLPR-5
Title	Human Centered Digital Twin
Scope/Purpose	Enable the Collection and Management of Humans’ Contextual Information for use in Digital Twins Applications
STAR Components Involved	Orchestrator, STAR Data Governance (IIoT Middleware) / Data Provenance & Traceability, Gateway/Orchestrator
UML Diagram	See Figure 19

5.2.6 Explainable Artificial Intelligence

The following table illustrates the XAI blueprint.

Table 9: STAR-BLPR-6 – Explainable Artificial Intelligence

Code	STAR-BLPR-6
Title	Explainable Artificial Intelligence in Manufacturing
Scope/Purpose	Implement Counterfactual logic and Features Ranking for AI/ML algorithms in the scope of AI-based Manufacturing Use Cases
STAR Components Involved	XAI Models (collection of models & algorithms), AI Algorithms (used in applications)
UML Diagram	See Figure 20 & Figure 21

5.2.7 Active Learning for Human Robot Collaboration

The following table illustrates the XAI blueprint.

Table 10: STAR-BLPR-7 – Active Learning for Human Robot Collaboration

Code	STAR-BLPR-7
Title	Active Learning for Human Robot Collaboration
Scope/Purpose	Boost Human Robot Collaboration and Accelerate Robot’s Acquisition based on Active learning
STAR Components Involved	STAR AI Algorithms, Active Learning Module
UML Diagram	See Figure 22

5.2.8 Feedback Provision

The following table illustrates the XAI blueprint.

Table 11: STAR-BLPR-8 – Provision of Feedback in Human in the Loop Scenarios

Code	STAR-BLPR-8
Title	Provision of Feedback during Human Robot Collaboration
Scope/Purpose	Enable humans to interact with Robots and Cyber physical systems towards providing feedback about their operations
STAR Components Involved	Feedback Module, NLP Module
UML Diagram	See Figure 23 and Figure 24Figure 22

5.2.9 Trusted Reconfiguration for Mobile Robot

The following table illustrates the Mobile Robot Reconfiguration blueprint.

Table 12: STAR-BLPR-9 – Trusted Reconfiguration of Mobile Robot

Code	STAR-BLPR-9
Title	Context Aware Reconfiguration of mobile robots
Scope/Purpose	Configure and control AMRs with secure and trusted commands (using the STAR secure data governance functions)
STAR Components Involved	Robots, Cameras, STAR Data Governance & IoT Middleware
UML Diagram	See Figure 27Figure 22

5.3 Additional Blueprints – Future Outlook

The following table presents a list of additional blueprints that will be specified over the STAR platform and will be mapped to the STAR.

Table 13: List of Additional Blueprints to be Specified over the STAR Platform

STAR-BLRP-10: Validating Statistical Distributions of Training Data
STAR-BLRP-11: Validating the Integrity of Industrial Data
STAR-BLRP-12: Security Policy Enforcement
STAR-BLRP-13: Reliable Data Generation for Simulated Reality
STAR-BLRP-14: Fast and Trusted Human Robot Collaboration based on Simulated Reality

STAR-BLRP-15: Configuration of Risk Assessment and Mitigation Engine

STAR-BLRP-16: Data Probes and Data Sources Configuration

STAR-BLRP-17: Real-Time Data Monitoring and Configuration

The above-list of blueprints is non-exhaustive and could be enhanced or updated in the scope of the next version of the deliverable.

6 Outlook and Conclusions

This deliverable has introduced the first version of the overall STAR architecture, including the main modules that comprise the STAR systems, the structuring principles that can enable their integration, as well as the main information flows between them. The architecture has been introduced in-line with the 4+1 architecture views model. In this version of the deliverable, the logical and process views have been prioritized, while physical deployment and implementation considerations have been briefly discussed. Overall, the introduction of the architecture has been driven by trusted AI requirements expressed in deliverable D2.1 and in the project's DoA. Likewise, the mapping of these requirements to a concrete architecture has considered principles and practices for state of the art industrial architectures, notably reference architecture such as the IIRA/IISF, the OpenFog RA and the BDVA/DAIRO Reference Model. This has boosted the specification of the architecture, as it has accelerated choices regarding the structuring of STAR components with respect to other systems in the manufacturing shopfloor. The resulting STAR architecture provides a high-level description of the structure of trusted AI systems based on a holistic approach that combines data and algorithms reliability, trustworthiness and transparency of AI algorithms, safe and human centred human robot collaboration, as well as the safe operation of autonomous systems like AMR. In this direction, a significant number of different modules are specified along with their interactions. Nevertheless, the STAR architecture is not an all-or-nothing value proposition: Developers and deployers of STAR compliant, trusted AI systems can implement, deploy and use selected subsets of the STAR modules in-line with the business requirements that they target.

One of the objectives of this deliverable was to provide a set of blueprint solutions for trusted AI in manufacturing. To this end, the deliverable has illustrated a number of commonly used use cases, along with the ways they can be implemented based on STAR modules and in-line with the STAR architecture. These blueprints could be very useful for industrial engineers and manufacturers that are looking for guidelines and proven solutions for implementing and deploying trusted AI systems. The blueprints that have been specified in this deliverable will be validated in the scope of the project's use cases implementation, as well as in the scope of other experimentation activities of the project.

As the first version of the overall architecture, it mainly focuses and elaborates on the logical and process views detailing the interactions between the main building blocks and the modules of the STAR environment. Furthermore, the present deliverable utilizes the initial views and descriptions of the use cases/pilots to validate the current design of the overall architecture. Following the detailed designs of all modules, their implementation, initial integration and evaluation through the use cases/pilots, the follow-up version of this report will further elaborate on the physical and implementation views. Currently, the STAR partners undertook work towards the above-listed directions, including the implementation of the architecture and its deployment in the STAR use cases. This work will be reported in the next version of the deliverable (D2.7), where 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 for the actual implementation of STAR systems in the context of the project use cases.

Overall, we envisage the present version of the architecture as a solid step towards developing and integrating STAR prototype systems and the STAR use cases. We also believe that this initial version of the architecture could be of wider interest to the AI community, notably to AI engineers, AI solutions providers, and manufacturers that wish to develop and deployed trusted AI systems in production lines. Therefore, the project will disseminate this early version of the architecture to interested groups in the AI and digital manufacturing communities.

References

Reference	Name of document
[BDVA17]	European Big Data Value Strategic Research and Innovation Agenda, Version 4.0, October 2017, https://bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf Accessed 20 Jul 2021.
[Chacon19]	H. Chacon, S. Silva and P. Rad, "Deep Learning Poison Data Attack Detection," 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI), 2019, pp. 971-978, doi: 10.1109/ICTAI.2019.00137.
[Docker]	Docker, Inc., "Docker Documentation", available at: https://docs.docker.com/ , last accessed: March 2020
[IEEE42010]	ISO/IEC/IEEE: "ISO/IEC/IEEE 42010:2011 Systems and software engineering -- Architecture description", 2011 http://www.iso.org/iso/catalogue_detail.htm?csnumber=50508
[IIRAv1.9]	Industrial Internet Consortium, "The Industrial Internet Reference Architecture v 1.9", available at: https://www.iiconsortium.org/IIRA.htm , last accessed September 2 nd 2021
[IISF]	Industrial Internet Consortium, "The Industrial Internet Security Framework", available at: https://www.iiconsortium.org/IISF.htm , last accessed September 2 nd 2021
[Khorshidpour16]	Z. Khorshidpour, S. Hashemi and A. Hamzeh, "Learning a Secure Classifier against Evasion Attack," 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), 2016, pp. 295-302, doi: 10.1109/ICDMW.2016.0049.
[Khurana19]	N. Khurana, S. Mittal, A. Piplai and A. Joshi, "Preventing Poisoning Attacks On AI Based Threat Intelligence Systems," 2019 IEEE 29th International Workshop on Machine Learning for Signal Processing (MLSP), 2019, pp. 1-6, doi: 10.1109/MLSP.2019.8918803.
[Kisller21]	E. Kisller, "What Is a Container Registry? And Why Do I Need One?," 26 March 2021. [Online]. Available: https://jfrog.com/knowledge-base/what-is-a-container-registry/ . [Accessed August 2021]
[Kruchten95]	Kruchten P. "Architectural Blueprints — The "4+1" View Model of Software Architecture". (1995) IEEE Software 12 (6), pp. 42-50.
[Rozanec21]	Joze M. Rozanec, Patrik Zajec, Klemen Kenda, Inna Novalija, Blaz Fortuna, Dunja Mladenic, Entso Veliou, Dimitrios Papamartzivanos, Thanassis Giannetsos, Sofia-Anna Menesidou, Rubén Alonso, Nino Cauli, Diego Reforgiato Recupero, Dimosthenis Kyriazis, Georgios Sofianidis, Spyros Theodoropoulos, John Soldatos: "STARdom: an architecture for trusted and secure human-centered manufacturing systems", In the Proc. Of the Advances in Production Management Systems (APMS) Conference, Sep 5-9, 2021.
[Shearer00]	Shearer C., The CRISP-DM model: the new blueprint for data mining, J Data Warehousing (2000); 5:13—22.
[Soldatos21]	John Soldatos (ed.), Dimosthenis Kyriazis (ed.) (2021), "Trusted

	Artificial Intelligence in Manufacturing: A Review of the Emerging Wave of Ethical and Human Centric AI Technologies for Smart Production", Boston-Delft: now publishers, http://dx.doi.org/10.1561/9781680838770 .
[Soldatos21a]	John Soldatos, Angela-Maria Despotopoulou, Nikos Kefalakis and Babis Ipektsidis 2021. "Blockchain Based Data Provenance for Trusted Artificial Intelligence" in Trusted Artificial Intelligence in Manufacturing: A Review of the Emerging Wave of Ethical and Human Centric AI Technologies for Smart Production. Edited by John Soldatos and Dimosthenis Kyriazis. pp. 1–29. Now Publishers. DOI: 10.1561/9781680838770.ch1.
[Souza 18]	H. Souza, "How to dockerize any application", May 2018, available at: https://hackernoon.com/how-to-dockerize-any-application-b60ad00e76da , last accessed at: March 2020
[Stepin21]	I. Stepin, J. M. Alonso, A. Catala and M. Pereira-Fariña, "A Survey of Contrastive and Counterfactual Explanation Generation Methods for Explainable Artificial Intelligence," in IEEE Access, vol. 9, pp. 11974-12001, 2021, doi: 10.1109/ACCESS.2021.3051315.