

**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:** Netcompany-Intrasoft



Funded by the Horizon 2020  
Framework Programme of the  
European Union

## DELIVERABLE

### D2.7 – STAR Reference Architecture and Blueprints- Final version

|                                     |   |
|-------------------------------------|---|
| <b>Dissemination level</b>          | PU -Public  |
| <b>Type of Document</b>             | Report  |
| <b>Contractual date of delivery</b> | 30/06/2022  |
| <b>Deliverable Leader</b>           | INTRA   |
| <b>Status - version, date</b>       | Final v1.0, 13/07/2022  |
| <b>WP / Task responsible</b>        | WP2   |
| <b>Keywords:</b>                    | 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

One of the main objectives of the STAR project is to outline the main building blocks of trusted AI systems for manufacturing applications and to provide structuring principles for the integration of these building blocks in AI use cases. In this direction, the project specifies and validates a reference architecture for trusted AI systems in manufacturing, which serves as a basis for the integration of the STAR technology modules and use cases. Moreover, the project specifies a set of blueprints for the configuration, deployment, and use of AI modules in trustworthy industrial use cases in manufacturing. This deliverable is devoted to the documentation of the STAR reference architecture, as well as to the presentation of the lower-level architecture concepts of the STAR technologies and use cases. The deliverable updates and enhances the contents of the earlier STAR deliverable D2.6, which provided an initial specification of the abovementioned architecture and blueprints.

The STAR reference architecture is introduced based on:

- A high-level reference model for trusted AI functionalities, which clusters the various functions in three domains namely cybersecurity, safety and human robot collaboration; and
- A set of more detailed views of the STAR architecture, which outline the main functionalities of the STAR technologies and use cases (logical view), the information flows between them (process view) and their implementation and deployment (physical/implementation views).

The deliverable presents logical, process, implementation, deployment and scenarios views of the STAR modules and use cases, in-line with the popular 4+1 view methodologies for representing software/middleware systems architectures. The present version of the deliverables illustrates the physical, the deployment and the implementation views of most STAR components such as the active learning, the simulated reality, the AI cyber defence, and AI security policies management, the mobile robots' safety and the human digital twin modules of the STAR project. Such views were not included in the first version of the deliverable. Moreover, this final version of the deliverable broadens the number of architecture blueprints that are aimed to facilitate integrators in the specification, deployment and use of popular functionalities of the STAR technologies. The updated list of blueprints addresses regulatory adherence issues as well, through indicating how STAR technologies facilitate compliance to the emerging European AI regulation i.e., to the mandates of the AI Act of the European Parliament and the Council of Europe. Such regulatory compliance blueprints were lacking in the earlier version of the deliverable.

Overall, the present deliverable provides a 360° architectural view of the STAR technologies, along with guidelines for integrating these technologies in real-life industrial use cases. As such, it serves as a useful guide for integrators of trusted AI systems both within and outside the STAR consortium. The deliverable is therefore valuable input to the technical workpackages of the project, including WP6 where integration activities are carried out. However, it will also provide useful inputs to other activities of the project, including training and pre-standardization activities, which can benefit from the presented architectural concepts and blueprints.

|                            |  |
|----------------------------|--|
| <b>Deliverable Leader:</b> | INTRA  |
| <b>Contributors:</b>       | UBITECH, R2M, JSI, UPRC, SUPSI, DFKI, PCL, GFT |
| <b>Reviewers:</b>          | UBITECH, GFT                                   |
| <b>Approved by:</b>        | INTRA  |

| <b>Document History</b> |             |                                 |  |
|-------------------------|-------------|---------------------------------|--|
| <b>Version</b>          | <b>Date</b> | <b>Contributor(s)</b>           | <b>Description</b>   |
| V0.1                    | 15/04/2022  | INTRA                           | Initial Table of Contents  |
| V0.12                   | 18/04/2022  | INTRA                           | Assignment of Partners Responsibilities                                      |
| V0.14                   | 20/05/2022  | INTRA                           | Introduction   |
| V0.15                   | 25/05/2022  | SUPSI                           | Updates about Physical and Deployment Views (WP5)                            |
| V0.16                   | 30/05/2022  | JSI                             | Updates about Physical and Deployment Views (WP4)                            |
| V0.17                   | 31/05/2022  | INTRA                           | Regulatory Compliance Blueprints in Section 4                                |
| V0.18                   | 31/05/2022  | UPRC                            | Physical and Implementation Views for XAI modules                            |
| V0.19                   | 01/06/2022  | INTRA                           | Various Edits and Quality Control  |
| V0.20                   | 06/06/2022  | JSI, UPRC                       | Inputs regarding Human Robot Collaboration in Section 2                      |
| V0.21                   | 08/06/2022  | INTA<br>UBITECH, GFT            | Inputs regarding Physical and Implementation Views in Section2               |
| V0.22                   | 14/06/2022  | DFKI, THALES                    | Inputs regarding the Physical and Implementation views of the AMR/RL systems |
| V0.23                   | 15/06/2022  | INTRA,<br>UBITECH,<br>GFT, UPRC | Updates to the blueprints (Section 4)  |
| V0.24                   | 14/06/2022  | PCL, DFKI, JSI,<br>UPRC, THALES | Implementation views of the PCL and DFKI use cases (Section 3)               |
| V0.25                   | 16/06/2022  | INTRA                           | Quality control, edits, Executive Summary                                    |
| V0.26                   | 24/06/2022  | UBITECH, GFT                    | 1 <sup>st</sup> Quality Review by UBITECH and GFT                            |
| V0.27                   | 27/06/2022  | INTRA                           | Addressing Comments from the Quality Review                                  |
| V0.28                   | 28/06/2022  | INTRA                           | Revised version of Sections 1 and 4 according to the review comments         |
| V0.30                   | 30/06/2022  | GFT, INTRA                      | Second review by GFT; Comments Addressed                                     |
| V0.40                   | 01/07/2022  | INTRA                           | Coordinator's Version for Final Quality Review and delivery                  |
| V1.0                    | 13/07/2022  | INTRA                           | QA and creation of the final submitted version                               |

# Table of Contents

|   |           |
|---|-----------|
| <b>EXECUTIVE SUMMARY .....</b>  | <b>2</b>  |
| <b>TABLE OF CONTENTS .....</b>  | <b>4</b>  |
| <b>TABLE OF FIGURES .....</b>   | <b>7</b>  |
| <b>LIST OF TABLES .....</b>   | <b>9</b>  |
| <b>DEFINITIONS, ACRONYMS AND ABBREVIATIONS .....</b>                            | <b>10</b> |
| <b>1 INTRODUCTION .....</b>   | <b>12</b> |
| 1.1 SCOPE AND PURPOSE .....   | 12        |
| 1.2 DELIVERABLE METHODOLOGY.....  | 12        |
| 1.2.1 4+1 Views Methodology.....  | 12        |
| 1.2.2 Alignment to Reference Architectures and Regulatory Compliance .....      | 13        |
| 1.2.3 Validation and Evolution.....   | 13        |
| 1.3 UPDATES AND ENHANCEMENTS TO PREVIOUS VERSION.....                           | 14        |
| 1.4 RELATIONSHIP TO OTHER STAR DELIVERABLES .....                               | 14        |
| 1.5 DOCUMENT STRUCTURE .....  | 16        |
| <b>2 STAR REFERENCE ARCHITECTURE .....</b>                                      | <b>17</b> |
| 2.1 HIGH LEVEL REFERENCE MODEL .....  | 17        |
| 2.2 LOGICAL VIEW.....   | 18        |
| 2.2.1 Overview.....   | 18        |
| 2.2.2 Digital Manufacturing Platforms and CPPS Systems.....                     | 19        |
| 2.2.3 Industry 4.0 AI Applications.....   | 20        |
| 2.2.4 Runtime Monitoring System (RMS).....                                      | 20        |
| 2.2.5 Data Provenance and Traceability (DPT).....                               | 20        |
| 2.2.6 STAR Blockchain – Distributed Ledger Infrastructure .....                 | 20        |
| 2.2.7 AI Cyber-Defence Strategies (ACDS).....                                   | 21        |
| 2.2.8 Risk Assessment and Mitigation Engine (RAME) .....                        | 22        |
| 2.2.9 Security Policies Manager (SPM) - Security Policies Repository (SPR)..... | 22        |
| 2.2.10 Machine Learning and Analytics Platform .....                            | 22        |
| 2.2.11 STAR XAI (Models & Library).....   | 23        |
| 2.2.12 Simulated Reality (SR).....  | 23        |
| 2.2.13 Active Learning (AL) .....   | 23        |
| 2.2.14 Production Processes Knowledge Base (PPKB) .....                         | 24        |
| 2.2.15 AMR Safety.....  | 24        |
| 2.2.16 Human Centred Digital Twin.....  | 24        |
| 2.2.17 Human Models – Human Digital Images.....                                 | 25        |
| 2.2.18 Application UI – Graphical User Interface (GUI) .....                    | 25        |
| 2.2.19 Natural Language Processing (NLP).....                                   | 25        |
| 2.2.20 Feedback Module .....  | 26        |
| 2.3 PROCESS VIEWS .....   | 26        |
| 2.3.1 Overview.....   | 26        |
| 2.3.2 Defending a Poisoning Attack.....   | 26        |
| 2.3.3 Defending an Evasion Attack.....  | 27        |
| 2.3.4 Dynamic Management and Configuration of Data Sources .....                | 28        |
| 2.3.5 Dynamic Management and Configuration of Analytic Processors.....          | 29        |
| 2.3.6 Metadata Validation using the STAR Blockchain.....                        | 29        |
| 2.3.7 Probe Data Transformation and storage.....                                | 30        |
| 2.3.8 Security Policy Management.....   | 31        |
| 2.3.9 Human Centric Digital Twin.....   | 32        |

|          |  |           |
|----------|--|-----------|
| 2.3.10   | <i>STAR XAI Models and Library Operations</i> .....                          | 33        |
| 2.3.11   | <i>Active Learning for Human Robot Interactions</i> .....                    | 34        |
| 2.3.12   | <i>Feedback Module Operation</i> .....                                       | 35        |
| 2.3.13   | <i>NLP Interaction</i> .....   | 36        |
| 2.4      | PHYSICAL VIEW(S) OF STAR ARCHITECTURE.....                                   | 37        |
| 2.4.1    | <i>Deployment Overview</i> .....   | 37        |
| 2.4.2    | <i>Deployment &amp; Ecosystem Management Technologies</i> .....              | 38        |
| 2.4.2.1  | Software Packaging with Docker images.....                                   | 39        |
| 2.4.2.2  | Container Tool with Docker Compose.....                                      | 39        |
| 2.4.2.3  | Code Management with GitLab.....   | 40        |
| 2.4.2.4  | Container Repository & Registry Management.....                              | 41        |
| 2.4.2.5  | Management/Monitoring with Portainer.....                                    | 41        |
| 2.4.2.6  | Access Control.....  | 42        |
| 2.4.3    | <i>Physical Views of STAR Cybersecurity Modules</i> .....                    | 42        |
| 2.4.3.1  | Physical View of DLSDR Infrastructure.....                                   | 43        |
| 2.4.3.2  | Physical View of Runtime Monitoring System (RMS).....                        | 44        |
| 2.4.3.3  | Physical View of Cyber-Defence Strategies.....                               | 45        |
| 2.4.3.4  | Physical View of Policy Manager.....   | 46        |
| 2.4.3.5  | Physical View of Risk Assessment and Mitigation Engine (OLISTIC).....        | 47        |
| 2.4.4    | <i>Physical Views of STAR Active Learning and XAI Modules</i> .....          | 48        |
| 2.4.4.1  | Physical View of Active Learning Module.....                                 | 48        |
| 2.4.4.2  | Physical View of the Explainable AI (XAI) Modules.....                       | 49        |
| 2.4.5    | <i>Physical Views of STAR Reinforcement Learning Modules (WP4/WP5)</i> ..... | 50        |
| 2.4.5.1  | Physical View of Safety Zones Detection Module.....                          | 50        |
| 2.4.5.2  | Physical View of Simulated Reality Module.....                               | 52        |
| 2.4.6    | <i>Physical Views of STAR Human Centric Digital Twins</i> .....              | 53        |
| 2.5      | IMPLEMENTATION VIEW OF STAR SYSTEMS.....                                     | 53        |
| 2.5.1    | <i>Implementation Overview</i> .....   | 53        |
| 2.5.2    | <i>Implementation Technologies</i> .....                                     | 54        |
| <b>3</b> | <b>STAR ARCHITECTURE VALIDATION – SCENARIOS VIEWS</b> .....                  | <b>56</b> |
| 3.1      | USE CASE #1 HUMAN ROBOT COLLABORATION FOR QUALITY MANAGEMENT.....            | 56        |
| 3.1.1    | <i>Process View</i> .....  | 56        |
| 3.1.2    | <i>Implementation View</i> .....   | 57        |
| 3.2      | USE CASE #2 HUMAN BEHAVIOUR PREDICTION AND SAFETY ZONES DETECTION.....       | 59        |
| 3.2.1    | <i>Process View</i> .....  | 60        |
| 3.2.2    | <i>Physical View</i> .....   | 60        |
| 3.2.3    | <i>Implementation View</i> .....   | 62        |
| 3.3      | USE CASE #3 HUMAN CENTRED AI FOR AGILE MANUFACTURING.....                    | 62        |
| 3.3.1    | <i>Process View</i> .....  | 62        |
| 3.3.2    | <i>Physical and Implementation Views</i> .....                               | 63        |
| <b>4</b> | <b>BLUEPRINTS SPECIFICATION</b> .....  | <b>64</b> |
| 4.1      | INTRODUCTION.....  | 64        |
| 4.2      | TECHNICAL INTEGRATION BLUEPRINTS ACROSS THE STAR FUNCTIONAL DOMAINS.....     | 64        |
| 4.2.1    | <i>Overview</i> .....  | 64        |

|          |   |           |
|----------|---|-----------|
| 4.2.2    | <i>Defending a Poisoning Attack</i> .....                           | 64        |
| 4.2.3    | <i>Defending an Evasion Attack</i> .....                            | 64        |
| 4.2.4    | <i>Management and Configuration of Data Sources</i> .....           | 65        |
| 4.2.5    | <i>Security Policy Management</i> .....                             | 65        |
| 4.2.6    | <i>Human Centred Digital Twin</i> .....                             | 65        |
| 4.2.7    | <i>Explainable Artificial Intelligence</i> .....                    | 66        |
| 4.2.8    | <i>Active Learning for Human Robot Collaboration</i> .....          | 66        |
| 4.2.9    | <i>Feedback Provision</i> .....                                     | 66        |
| 4.2.10   | <i>Trusted Reconfiguration for Mobile Robot</i> .....               | 66        |
| 4.2.11   | <i>Management and Configuration of Analytics Processors</i> .....   | 67        |
| 4.2.12   | <i>Validating the Integrity of Industrial Data</i> .....            | 67        |
| 4.2.13   | <i>Security Policy Definition and Enforcement</i> .....             | 67        |
| 4.2.14   | <i>Reliable Data Generation for Simulated Reality</i> .....         | 68        |
| 4.2.15   | <i>Configuration of Risk Assessment and Mitigation Engine</i> ..... | 68        |
| 4.2.16   | <i>Data Probes Configuration</i> .....                              | 68        |
| 4.2.17   | <i>Real-Time Data Monitoring</i> .....                              | 68        |
| 4.3      | REGULATORY COMPLIANCE BLUEPRINTS – COMPLIANCE TO AI ACT.....        | 69        |
| 4.3.1    | <i>Overview</i> .....   | 69        |
| 4.3.2    | <i>Minimal-Risk and No-Risk Systems</i> .....                       | 69        |
| 4.3.3    | <i>Limited-Risk Systems</i> .....                                   | 70        |
| 4.3.4    | <i>High-Risk Systems</i> .....                                      | 71        |
| <b>5</b> | <b>CONCLUSIONS</b> .....  | <b>73</b> |
|          | <b>REFERENCES</b> .....   | <b>74</b> |

# Table of Figures

FIGURE 1: THE TWO ITERATIONS OF THE STAR ARCHITECTURE..... 14

FIGURE 2: HIGH LEVEL REFERENCE MODEL ..... 17

FIGURE 3: STAR FUNCTIONAL MODULES AND LOGICAL VIEW OF THE ARCHITECTURE..... 19

FIGURE 4: INSTANTIATION OF THE SECURITY MODULES OF THE STAR ARCHITECTURE ..... 21

FIGURE 5: DETAILED ARCHITECTURE OF THE ACTIVE LEARNING SYSTEM FOR HUMAN ROBOT COLLABORATION ..... 23

FIGURE 6: DETAILED LOGICAL ARCHITECTURE OF THE HUMAN CENTRED DIGITAL TWIN (HDT) ..... 25

FIGURE 7: INFORMATION FLOW FOR A DEFENDING A POISONING ATTACK ..... 27

FIGURE 8: INFORMATION FLOW FOR A DEFENDING AN EVASION ATTACK..... 28

FIGURE 9: PROCESS VIEW OF A DATA SOURCE MANAGEMENT USE CASE ..... 28

FIGURE 10: PROCESSOR PERSISTENCE TO DISTRIBUTED LEDGER NETWORK ..... 29

FIGURE 11: METADATA VALIDATION USING THE STAR BLOCKCHAIN ..... 30

FIGURE 12: PROBE DATA STORAGE SEQUENCE DIAGRAM ..... 31

FIGURE 13: PROCESS VIEW OF A SECURITY POLICY MANAGEMENT USE CASE ..... 31

FIGURE 14: PROCESS VIEW OF THE HUMAN CENTRIC DIGITAL TWIN OPERATION ..... 32

FIGURE 15: PROVISION OF COUNTERFACTUALS INFORMATION BY THE STAR XAI..... 33

FIGURE 16: FEATURES RANKING OPERATIONS BY THE STAR AI..... 34

FIGURE 17: ACTIVE LEARNING MODULE OPERATION IN SUPPORT OF HUMAN ROBOT INTERACTION ..... 35

FIGURE 18: OPERATION OF THE FEEDBACK MODULE..... 36

FIGURE 19: HIGH LEVEL VIEW OF THE NLP OPERATION IN THE CONTEXT OF THE STAR IMPLEMENTATION ..... 37

FIGURE 20 A COMPLETE GIT BRANCHING MODEL..... 40

FIGURE 21: DEPLOYMENT DIAGRAM FOR THE CYBERSECURITY MODULES OF THE STAR ARCHITECTURE (I.E., MODULES DEVELOPED IN WP3)..... 43

FIGURE 22: DLSDR DEPLOYMENT DIAGRAM ..... 44

FIGURE 23: RMS DEPLOYMENT DIAGRAM..... 45

FIGURE 24: AI CYBER DEFENCE DEPLOYMENT DIAGRAM..... 46

FIGURE 25: SSPM PHYSICAL VIEW..... 47

FIGURE 26 OLISTIC DEPLOYMENT DIAGRAM..... 48

FIGURE 27 THE AL SERVICE CAN BE DEPLOYED AS AN ON PREMISE OR CLOUD APPLICATION ..... 49

FIGURE 28: PHYSICAL VIEW OF THE STAR XAI COMPONENT/MODULES ..... 50

FIGURE 29: SAFETY ZONE DETECTION & AMR FLEET OPTIMIZER LOGICAL VIEW WITH PHYSICAL AND DEPLOYMENT VIEWS CONSIDERATIONS..... 51

FIGURE 30: SAFETY ZONE DETECTION LOGICAL VIEW WITH PHYSICAL AND DEPLOYMENT VIEWS CONSIDERATIONS ..... 51

FIGURE 31: HIGH-LEVEL PHYSICAL VIEW OF SIMULATED REALITY MODULE..... 52

FIGURE 32: HUMAN DIGITAL TWIN CORE INFRASTRUCTURE DEPLOYMENT SHOWCASE ..... 53

FIGURE 33: HIGH LEVEL PROCESS VIEW AND INFORMATION FLOWS FOR THE HUMAN ROBOT COLLABORATION USE CASE ..... 57

FIGURE 34: HIGH-LEVEL IMPLEMENTATION VIEW OF THE QUALITY CONTROL/MANAGEMENT USE CASE 57

FIGURE 35: PRODUCTION CELL..... 58

FIGURE 36: VISION SYSTEM COMPONENTS..... 58

FIGURE 37: QUALITY CONTROL ALGORITHM IMPLEMENTATION BASED ON PYTHON ..... 58

FIGURE 38: PROCESS VIEW FOR AS-IS SCENARIO ..... 60

FIGURE 39: PROCESS VIEW FOR TO-BE SCENARIO ..... 60

FIGURE 40: PHYSICAL VIEW FOR AS-IS SCENARIO ..... 61

FIGURE 41: PHYSICAL VIEW FOR TO-BE SCENARIO ..... 61

FIGURE 42: SOFTWARE COMPONENTS OF THE AMR FLEET OPTIMIZER ..... 62

FIGURE 43: OVERVIEW OF IBER UC PROCESSES (SEE ALSO DELIVERABLE D6.1)..... 63

FIGURE 44: PRELIMINARY IMPLEMENTATION VIEW OF THE AGILE MANUFACTURING USE CASE(S) AT IBER  
..... 63

FIGURE 45: OVERVIEW OF THE RISK LEVELS OF THE AI ACT ..... 69

FIGURE 46: REQUIREMENTS OF MINIMAL OR NO RISK SYSTEMS..... 70

FIGURE 47: REQUIREMENTS FOR LIMITED RISKS SYSTEMS ..... 71

FIGURE 48: REQUIREMENTS FOR HIGH-RISK SYSTEMS ..... 72

## List of Tables

|   |    |
|---|----|
| TABLE 1: GUIDE FOR INDUSTRIAL DEPLOYMENTS AT THE CLOUD/EDGE/FAR EDGE.....                           | 37 |
| TABLE 2: EDGE/CLOUD DEPLOYMENT CONSIDERATIONS FOR THE MAIN COMPONENTS OF THE STAR ARCHITECTURE..... | 38 |
| TABLE 3: ENVISAGED IMPLEMENTATION TECHNOLOGIES.....   | 54 |
| TABLE 4: STAR-BLPR-1 - POISONING ATTACK DEFENCE .....   | 64 |
| TABLE 5: STAR-BLPR-2 - EVASION ATTACK DEFENCE.....  | 65 |
| TABLE 6: STAR-BLPR-3 – MANAGEMENT AND CONFIGURATION OF INDUSTRIAL DATA SOURCES .....                | 65 |
| TABLE 7: STAR-BLPR-4 – SECURITY POLICY MANAGEMENT.....  | 65 |
| TABLE 8: STAR-BLPR-5 –HUMAN CENTRED DIGITAL TWIN.....   | 65 |
| TABLE 9: STAR-BLPR-6 – EXPLAINABLE ARTIFICIAL INTELLIGENCE .....                                    | 66 |
| TABLE 10: STAR-BLPR-7 – ACTIVE LEARNING FOR HUMAN ROBOT COLLABORATION .....                         | 66 |
| TABLE 11: STAR-BLPR-8 – PROVISION OF FEEDBACK IN HUMAN IN THE LOOP SCENARIOS .....                  | 66 |
| TABLE 12: STAR-BLPR-9 – TRUSTED RECONFIGURATION OF MOBILE ROBOT.....                                | 67 |
| TABLE 13: STAR-BLPR-10 – MANAGEMENT AND CONFIGURATION OF INDUSTRIAL DATA SOURCES.....               | 67 |
| TABLE 14: STAR-BLPR-11 – MANAGEMENT AND CONFIGURATION OF INDUSTRIAL DATA SOURCES.....               | 67 |
| TABLE 15: STAR-BLPR-12 – SECURITY POLICY DEFINITION AND ENFORCEMENT BLUEPRINT.....                  | 67 |
| TABLE 16: STAR-BLPR-13 – DATA GENERATION USING SIMULATED REALITY .....                              | 68 |
| TABLE 17: STAR-BLPR-14 – CONFIGURATION OF RISK ASSESSMENT AND MITIGATION ENGINE.....                | 68 |
| TABLE 18: STAR-BLPR-15 – MANAGEMENT AND CONFIGURATION OF INDUSTRIAL DATA SOURCES.....               | 68 |
| TABLE 19: STAR-BLPR-16 – MANAGEMENT AND CONFIGURATION OF INDUSTRIAL DATA SOURCES.....               | 69 |
| TABLE 20: STAR-REG-BLPR-1 – SUPPORTING THE DEPLOYMENT OF AI SYSTEMS OF MINIMAL RISK ..              | 70 |
| TABLE 21: STAR-REG-BLPR-2 – SUPPORTING THE DEPLOYMENT OF AI SYSTEMS OF LIMITED RISK ...             | 71 |
| TABLE 22: STAR-REG-BLPR-3 – SUPPORTING THE DEPLOYMENT OF AI SYSTEMS OF LIMITED RISK ...             | 72 |

## Definitions, Acronyms and Abbreviations

| Acronym/<br>Abbreviation | Title   |
|--------------------------|---|
| <b>ACDS</b>              | AI Cyber-Defense Strategies                           |
| <b>AI</b>                | Artificial Intelligence                               |
| <b>AL</b>                | Active Learning                                       |
| <b>API</b>               | Application Programming Interface                     |
| <b>ART</b>               | Adversarial Robustness Toolbox                        |
| <b>BDVA</b>              | Big Data Value Association                            |
| <b>CE</b>                | Community Edition                                     |
| <b>CERT</b>              | Computer Emergency Response Team                      |
| <b>CLI</b>               | Command Line Input                                    |
| <b>CoAP</b>              | Constrained Application Protocol                      |
| <b>CPPS</b>              | Cyber Physical Production System                      |
| <b>CPS</b>               | Cyber Physical System                                 |
| <b>DAIRO</b>             | Data AI and Robotics                                  |
| <b>DL</b>                | Deep Learning   |
| <b>DLSDR</b>             | Distributed Ledger Services for Data Reliability      |
| <b>DLT</b>               | Distributed Ledger Technologies                       |
| <b>DoA</b>               | Description of Action                                 |
| <b>DPT</b>               | Data Provenance & Traceability                        |
| <b>EP</b>                | European Parliament                                   |
| <b>ERP</b>               | Enterprise Resource Planning                          |
| <b>GUI</b>               | Graphical User Interface                              |
| <b>HDT</b>               | Human Centered Digital Twin                           |
| <b>HTTP</b>              | HyperText Transfer Protocol                           |
| <b>IEEE</b>              | Institute of Electrical and Electronics Engineers     |
| <b>IETF</b>              | Internet Engineering Task Force                       |
| <b>IIAF</b>              | Industrial Internet Architecture Framework            |
| <b>IIC</b>               | Industrial Internet Consortium                        |
| <b>IIoT</b>              | Industrial Internet of Things                         |
| <b>IIRA</b>              | Industrial Internet Consortium Reference Architecture |
| <b>IISF</b>              | Industrial Internet Security Framework                |
| <b>IoT</b>               | Internet of Things                                    |
| <b>ISO</b>               | International Organization for Standardization        |
| <b>JDBC</b>              | Java Database Connectivity                            |
| <b>LIME</b>              | Local Interpretable Model Agnostic Explanations       |
| <b>MES</b>               | Manufacturing Execution Systems                       |
| <b>MQTT</b>              | Message Queue Telemetry Transport                     |
| <b>ML</b>                | Machine Learning                                      |
| <b>MPPE</b>              | Multi-Purpose Processing Engine                       |
| <b>NLP</b>               | Natural Language Processing                           |
| <b>OPA</b>               | Open Policy Agent                                     |
| <b>PLM</b>               | Product Lifecycle Management                          |
| <b>PPKB</b>              | Production Processes Knowledge Base                   |

|             |                                       |
|-------------|---------------------------------------|
| <b>RA</b>   | Reference Architecture                |
| <b>RAME</b> | Risk Assessment and Mitigation Engine |
| <b>REST</b> | Representational State Transfer       |
| <b>RFC</b>  | Request for Comments                  |
| <b>RL</b>   | Reinforcement Learning                |
| <b>RM</b>   | Reference Model                       |
| <b>RMS</b>  | Runtime Monitoring System             |
| <b>SC</b>   | Smart Contract                        |
| <b>SPR</b>  | Security Policies Repository          |
| <b>SR</b>   | Simulated Reality                     |
| <b>SSPM</b> | Security Policy Manager               |
| <b>STT</b>  | Speech To Text                        |
| <b>TCP</b>  | Transport Control Protocol            |
| <b>TTS</b>  | Text To Speech                        |
| <b>UI</b>   | User Interface                        |
| <b>UML</b>  | Unified Modelling Language            |
| <b>WIRM</b> | Worker Intention Recognition Module   |
| <b>WP</b>   | Work Package                          |
| <b>VM</b>   | Virtual Machine                       |
| <b>XAI</b>  | Explainable Artificial Intelligence   |

# 1 Introduction

## 1.1 Scope and Purpose

One of the main objectives of the H2020 STAR project is to provide a Reference Architecture (RA) for trusted Artificial Intelligence (AI) systems. The goal of this architecture is to boost the development of the STAR platform and use cases, while at the same time serving as blueprint for trusted manufacturing systems in the Industry 5.0 era. STAR has already released the first version of its reference architecture as part of deliverable D2.6, which can be considered as the first/earlier version of the present deliverable. The purpose of the present deliverable is to provide an updated version of this reference architecture, considering:

- Feedback regarding the STAR architecture from STAR partners that have been consulting and using it as part of their efforts to develop STAR systems and industrial use cases.
- Implementation and deployment information regarding the STAR systems, notably information that was not known and available at the time when D2.6 was released.
- Updates and revisions to the technical modules that comprise the STAR platform, notably modules for AI cyber-security, the safety of AI-based systems in production lines, as well as modules for human-robot collaboration (e.g., active learning). These technical modules are under development in workpackages WP3, WP4 and WP5 of the project.

The present deliverable enhances and completes the initial version of the architecture that was presented in D2.6. Specifically, it provides information about the deployment and implementation views of the architecture. Moreover, it includes more detailed information about how the STAR systems and use cases leverage the structuring principles and technical components of the architecture.

## 1.2 Deliverable Methodology

### 1.2.1 4+1 Views Methodology

As outlined in D2.6, the development, documentation, and presentation of the STAR architecture is based on the popular 4+1 views methodology for describing software systems architectures [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 initial version of the STAR architecture in D2.6 focused on the logical and process views of the above-listed methodology. Specifically, D2.6 presented:

- A logical view of the architecture, including the main STAR modules, their functionalities and the structuring principles that drive their integration.
- Process view of the STAR systems, including the main interactions and information flows between the STAR systems.
- Preliminary information about the implementation and deployment of the STAR modules.

This version provides more detailed information about the implementation and deployment aspects of the architecture that was missing in the previous version. Moreover, more detailed views of the use cases and their alignment to the architecture is given.

### 1.2.2 Alignment to Reference Architectures and Regulatory Compliance

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 served as a starting point for the definition of the STAR Reference Architecture. They provided 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 earlier version of this deliverable (D2.6) extended these existing architectures and combined their building blocks towards introducing blueprints for trusted AI systems. Specifically, D2.6, analysed and considered 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.

This deliverable does not include information about these reference architectures. Interested readers should consult D2.6. However, this version attempts an initial alignment of the STAR architecture with emerging regulatory requirements stemming from the AI Act proposal of the European Parliament and the Council of Europe. Therefore, the deliverable makes a brief reference to the AI Act and the way it mandates a risk-based classification of AI systems. This is also used in order to illustrate how the STAR reference architecture and its technical modules could serve as blueprints for developing regulatory compliant systems.

### 1.2.3 Validation and Evolution

The STAR architecture is developed and validated in two iterations. The first iteration was specified in deliverable D2.6 and was driven by the project's activities during the first semester of STAR's lifetime. This deliverable reflects the second iteration of the deliverable, which incorporates inputs from the on-going technical implementation activities of the project in WP3, WP4 and WP5. This enables our project to receive feedback from the actual, implementation, deployment and use of the first version of the architecture. Figure 1 illustrates the methodology for the specification and validation of the STAR-RA.

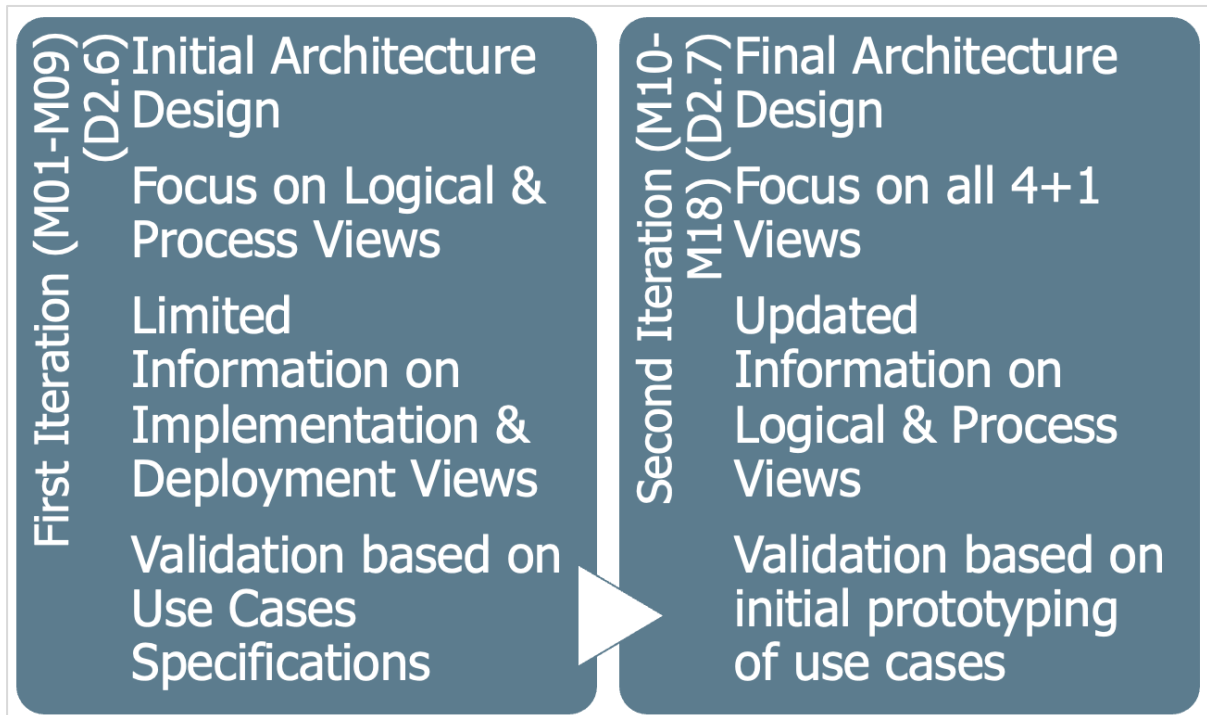


Figure 1: The two Iterations of the STAR Architecture

### 1.3 Updates and Enhancements to Previous Version

Overall, the present version of the deliverable contains the following main changes with respect to the earlier version (D2.6):

- The present deliverable does not analyse the state of the art in reference architectures for industrial systems. Interested readers are advised to consult deliverable D2.6.
- The logical view of the reference architecture has been updated.
- The deliverable includes more information about the implementation view and the practical implementation of the architecture.
- The deliverable includes more information about the physical view and the physical deployment of the architecture.
- Views associated with the project’s cases have been updated.
- The present deliverable includes information about the alignment of the architecture to the emerging AI regulation i.e., the AI of the European Parliament (EP) and the Council of Europe.

### 1.4 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 considered 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, including the latest updates to D2.1 following revisions requested by the reviewers of the project. Following the technical review of the project in January 2022, D2.1 has been further updated with additional KPIs and benchmarks. This updated version of D2.1 has been consulted and used as background material for D2.1.

- **D2.2 Reference Scenarios and Use Cases for AI in Manufacturing:** The reference scenarios of these deliverables have been considered as an initial input for the validation of the architecture as part of the 4+1 methodology. However, the present version of the deliverable has also considered the more detailed specification of the project's use cases in deliverable D6.1. Hence, the scenarios' view of the present deliverable (D2.7) sheds light on the validation of the architecture against the STAR use cases considering the most up to date information about the project's use cases.
- **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., D2.7. 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.
- **D2.5 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. D2.6 has considered the information contained in D2.4 in terms of data models and data collection specifications. Likewise, the development of these deliverables has considered relevant work undertaken as part of deliverable D2.5, which extends and enhances the work carried out in D2.4.
- **D3.2 Decentralized Reliability for Industrial Data and Distributed Analytics:** The present deliverable details the logical interactions of the components that comprise the decentralized data provenance infrastructure of the project. It, therefore, supports the development of D3.2, much in the same way, the earlier version of the architecture deliverable (D2.6) influenced the development of the first prototype of the project's decentralized infrastructure for industrial data reliability (i.e. D3.1). However, D3.2 is expected to provide additional information and technical details (e.g., components and APIs specifications) than what is illustrated in the present deliverable.
- **D3.4 Cyber-defence Mechanisms against Poisoning and Evasion Attacks:** Similar to the case of D3.2 above, D3.4 will benefit from the architecture specifications of the cyber-defence mechanisms that are provided in this deliverable. Specifically, the present deliverable illustrates some logical interactions between the main components of the cyber-defence systems, which will serve as a basis for more detailed specifications and prototyping in the context of D3.4. Overall, D2.7 serves as input for D3.4, in a way similar to how the earlier version of this deliverable (D2.6) influenced the first version of the cyber-defence mechanism of the project (D3.3).
- **D4.1 & D4.2 Library of XAI algorithms:** The explainable AI components of the STAR project are 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 benefit from the project's XAI library. It also provides information about the interactions of the XAI modules/algorithms with other STAR technologies in-line with the STAR architecture. As such it is related to deliverables D4.1 and D4.2, which provide the detailed specifications and prototype implementations of STAR's XAI mechanisms. The earlier version of the present deliverable (D2.6) influenced D4.1, while the current version (D2.7) will provide input to the development of D4.2.

- **D5.1 Digital Models for Human Centric AI-based Production:** The STAR digital models play an important role in the specification, design and implementation of the human-centric digital twins of the project, which are used in the scope of the human robot collaboration developments of STAR. Therefore, these models are integral elements of the implementation view and the implementation information of the STAR architecture.
- **D5.3 Digital Twins for Security and Safety - Initial version and D5.4 Digital Twins for Security and Safety - Final version:** This deliverable defines the digital twins of the project and their implementation. These digital twins are used in the specification of some of the process interactions and blueprints of this deliverable. The previous version of the deliverable (D2.6) served as input for D2.6, while the present version will provide inputs to D5.4.
- **D6.1 Report on Pilot Sites Preparation-Initial version & D6.2 Report on Pilot Sites Preparation-Final version:** These two deliverables describe the activities undertaken at the three STAR pilot sites towards ensuring that the production lines will be ready to deploy STAR. Information from these deliverables has been consulted in order to develop the scenarios views of the architecture in Section 3.

## 1.5 Document Structure

The deliverable is structured as follows:

- Section 2 following this introductory session presents the updated logical and process views of the STAR architecture. It also provides additional information on the implementation and deployment of the STAR systems, including details on implementation and physical views of the architecture.
- Section 3 is devoted to the presentation of the project's use cases as a validation view of the architecture. Specifically, it discusses how the use cases of the project can be supported by the specified logical, physical and implementation views of the architecture.
- Section 4 updates the list of blueprints that have been presented in deliverable D2.6. New blueprints are driven by the project's use cases, as well as by regulatory compliance.
- Section 5 is the final and concluding section of the deliverable.

## 2 STAR Reference Architecture

### 2.1 High Level Reference Model

In D2.6 we introduced a high-level reference model for trusted AI systems in general, including the STAR platform developments. This model clusters the functionalities of the STAR platform in three main categories or functional domains according to the terminology of the Industrial Internet Reference Architecture (IIRA) of the Industrial Internet Consortium (IIC). These three domains are illustrated in Figure 2 and are as follows:

- **Cybersecurity Domain:** This domain includes the functionalities that are destined to boost the cybersecurity and cyber resilience of AI systems in production lines. In the scope of the STAR platform, these functionalities ensure the reliability and security of industrial data, as well as of the AI algorithms that are trained and executed based on these data. The functionalities of this domain 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. With reference to the STAR platform, this domain includes the project's active learning functionalities, as well as production knowledge bases and functionalities that enable interactions between humans and machines.
- **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 (Automatic Mobile Robots) in industrial sites.

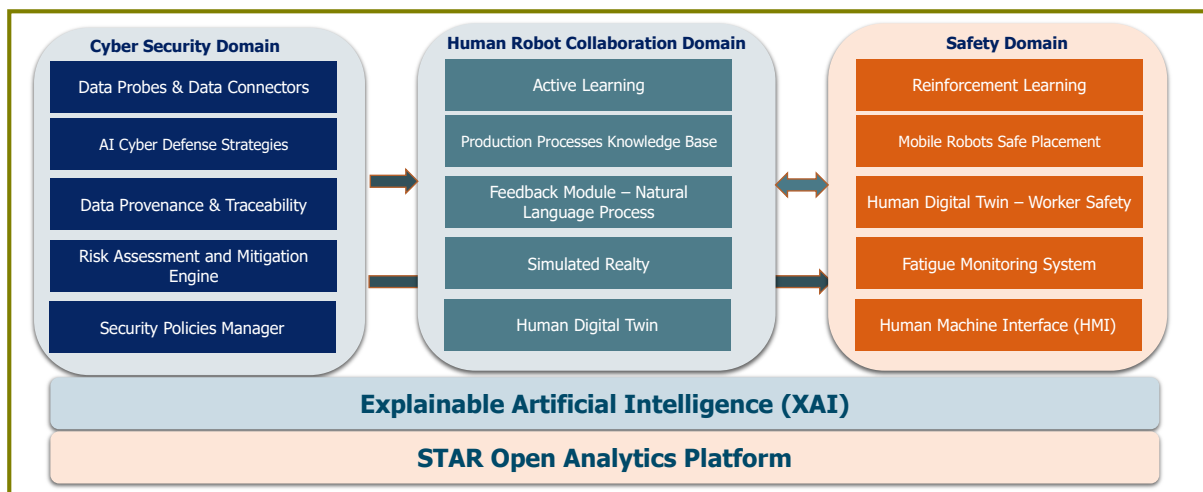


Figure 2: High level Reference Model

As illustrated in Figure 2, the functionalities of all domains depend on Explainable AI (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 views of the architecture.

## 2.2 Logical View

### 2.2.1 Overview

A logical view of the STAR architecture is presented in Figure 3. 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 for alleviating 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-defence strategies that protect and defend AI systems from malicious security attacks. STAR focuses primarily on defences 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 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 hand, 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 trusted and reliable industrial data. This boosts and reinforces their trustworthiness. Moreover, they can be integrated with the STAR's cyber-defense strategies to ensure that they cannot be tampered or compromised by malicious parties. This is yet another step to strengthening the trustworthiness of STAR systems for safe AMR operations. 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 integrating the project's systems for trusted AI. However, the STAR functionalities do not represent an "all-or-nothing" value proposition. It is possible for manufacturers and Industry 4.0 systems integrators to opt for adopting and implementing only parts of the STAR architecture i.e., specific modules of the logical architecture. 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.

As illustrated in Figure 3 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).

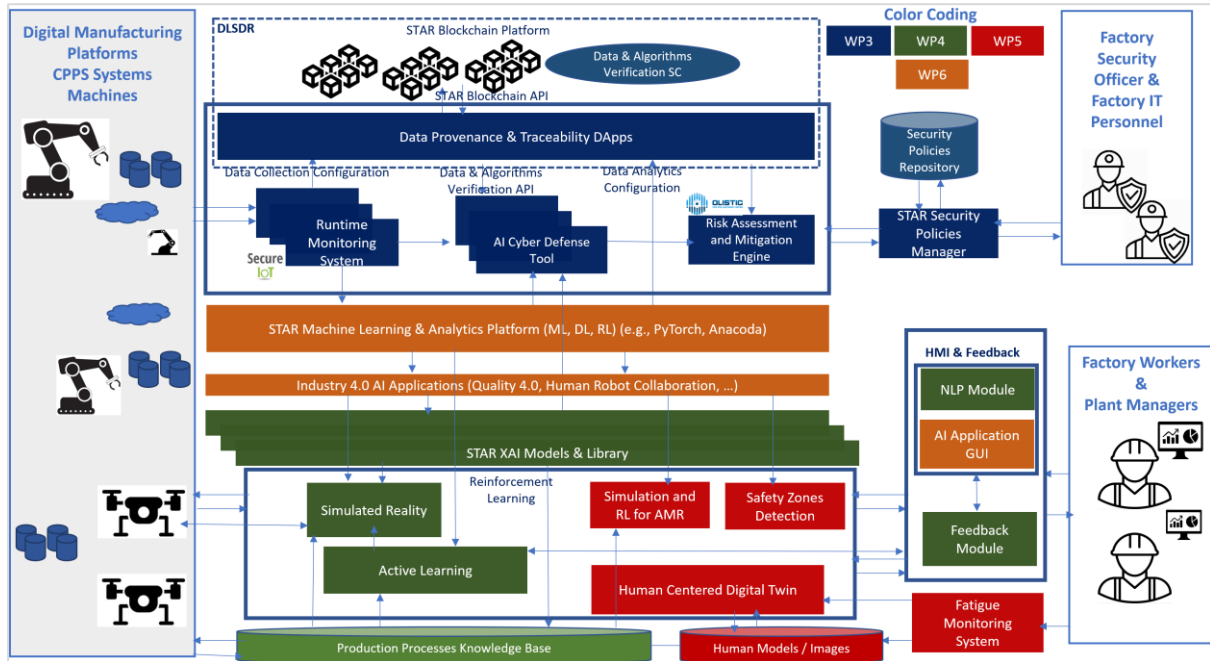


Figure 3: STAR Functional Modules and Logical View of the Architecture

The following paragraphs describe the various modules and building blocks of the architecture by providing a summary of their core functionality. Additional technical details are provided in deliverables of the technical workpackages where the various modules are implemented, including WP3, WP4 and WP5. In the current deliverable, a summary of the functionalities of the various components is provided, along with complementary architectural views such as process, implementation, and deployment views.

### 2.2.2 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 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. Any 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 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.

### 2.2.3 Industry 4.0 AI Applications

This main building block represents different types of AI-based industrial applications such as machine learning (ML) 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 platforms, AI applications can be data sources, which contribute data to the operation of STAR's data driven systems.

### 2.2.4 Runtime Monitoring System (RMS)

RMS building block is a data collection solution which offers a real time data collection, transformation, filtering, and management service to facilitate data consumers (e.g., AI Cyber Defence Module and Security Policy Manager) with accessing the required data. The building block can be applied in IoT environments supporting solutions in various domains (e.g., Industrial, Cybersecurity, etc.). For example, the solution may be used to collect security related data (e.g., network, system, solution proprietary, etc.) from monitored IoT systems and store them to detect patterns of abnormal behaviour by applying simple (i.e., filtering) or more elaborate (i.e., deep learning) data processing mechanisms. The solution features specialized probes, that may be deployed to the monitored IoT system, or polling services for acquiring data from shopfloor sources (e.g., CPPS systems and digital manufacturing platforms).

### 2.2.5 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.

### 2.2.6 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 enable

decentralized applications that provide information about the metadata to interested STAR modules such as the cyber-defence strategies module.

### 2.2.7 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 the 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 a similar 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-defense 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).

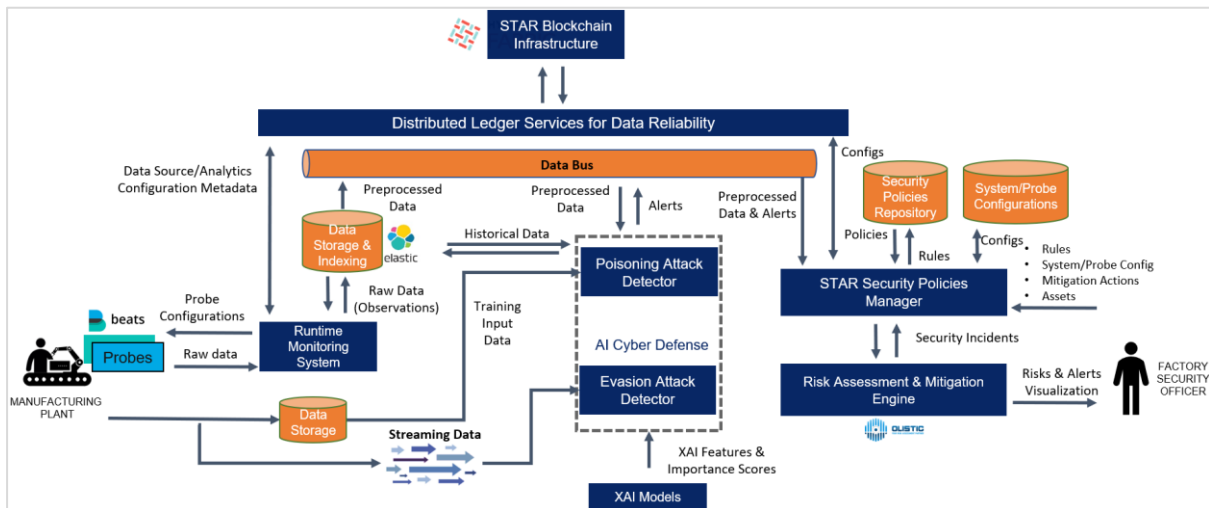


Figure 4: Instantiation of the Security Modules of the STAR Architecture

Figure 4 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 Runtime Monitoring System (RMS), which collects monitoring data from the deployed probes, as well as with the STAR

XAI models and library. Furthermore, Figure 4 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.

### 2.2.8 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).

### 2.2.9 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)).

### 2.2.10 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 to or provide datasets for training and executing AI algorithms such as the data probes and the data connectors.

### 2.2.11 STAR XAI (Models & Library)

This module provides and executes Explainable Artificial Intelligence models and algorithms. Like 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.

### 2.2.12 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 facilitate humans to assess the appropriateness and correctness of the simulated data that are generated by the SR.

### 2.2.13 Active Learning (AL)

This module provides a placeholder for AL systems i.e., AI systems that 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 5, 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].

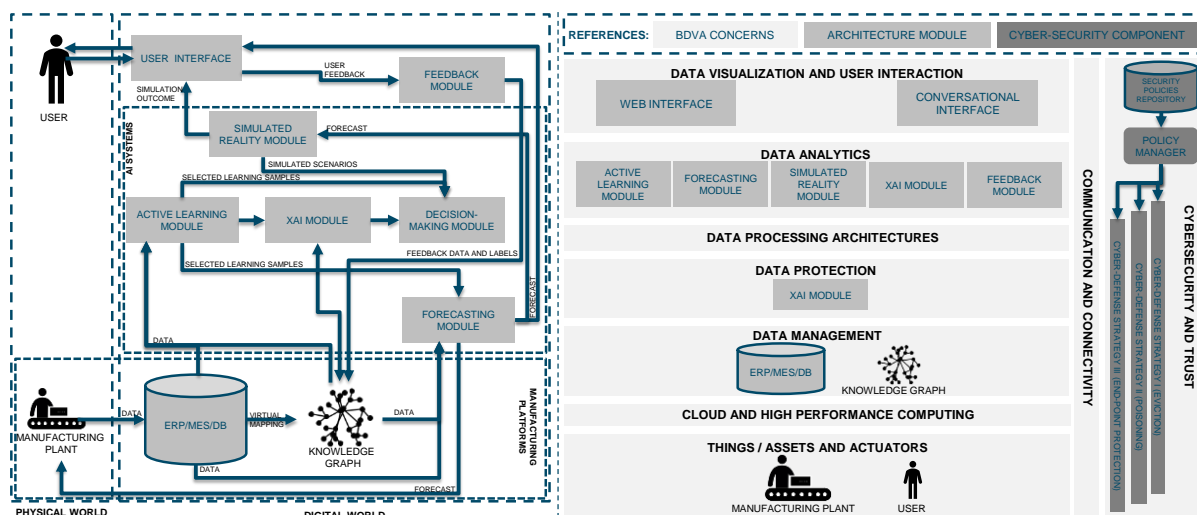


Figure 5: Detailed Architecture of the Active Learning System for Human Robot Collaboration

### 2.2.14 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 the AL module whenever the AL consults the authority. This helps accelerating knowledge acquisition.

### 2.2.15 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.

### 2.2.16 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 6:

- **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 connections and 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 features 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 publish data on the IoT 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 logic.

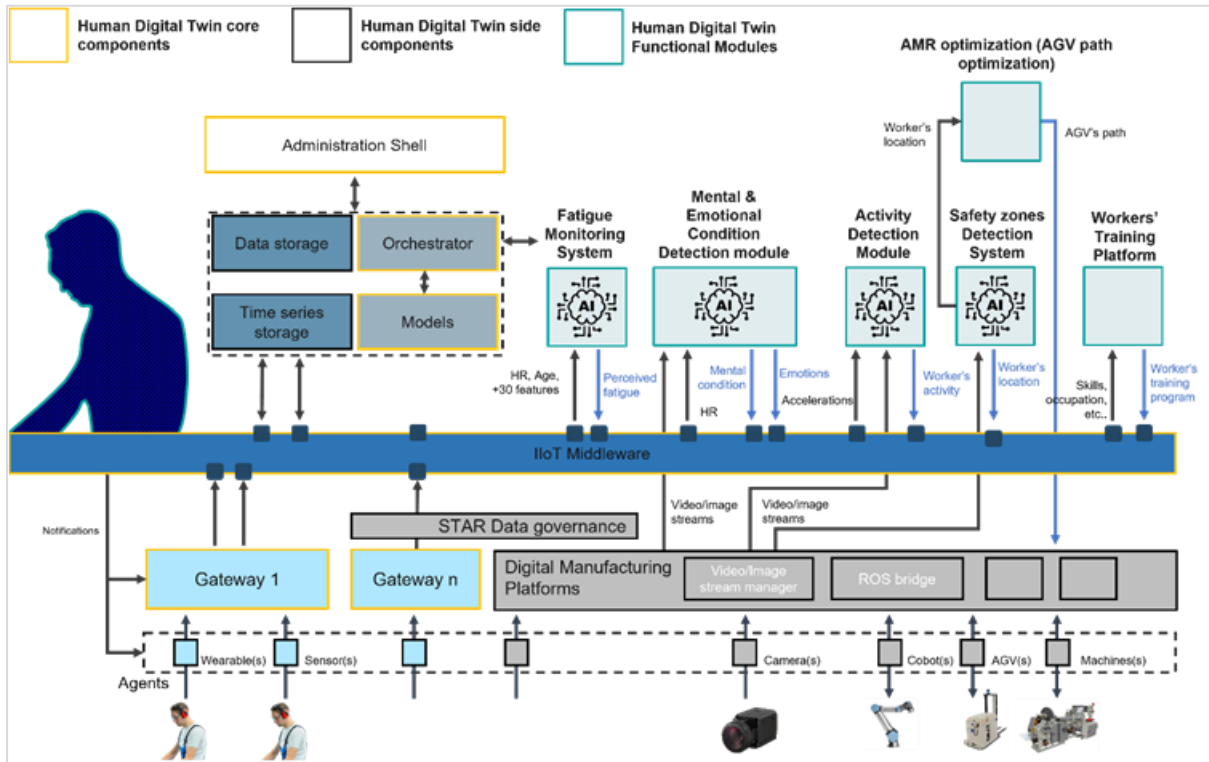


Figure 6: Detailed Logical Architecture of the Human Centred Digital Twin (HDT)

### 2.2.17 Human Models – Human Digital Images

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

### 2.2.18 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).

### 2.2.19 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, the NLP modality is used for the interaction between workers and the AL module.

## 2.2.20 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.

## 2.3 Process Views

### 2.3.1 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.

### 2.3.2 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 entails 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 7 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. 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).

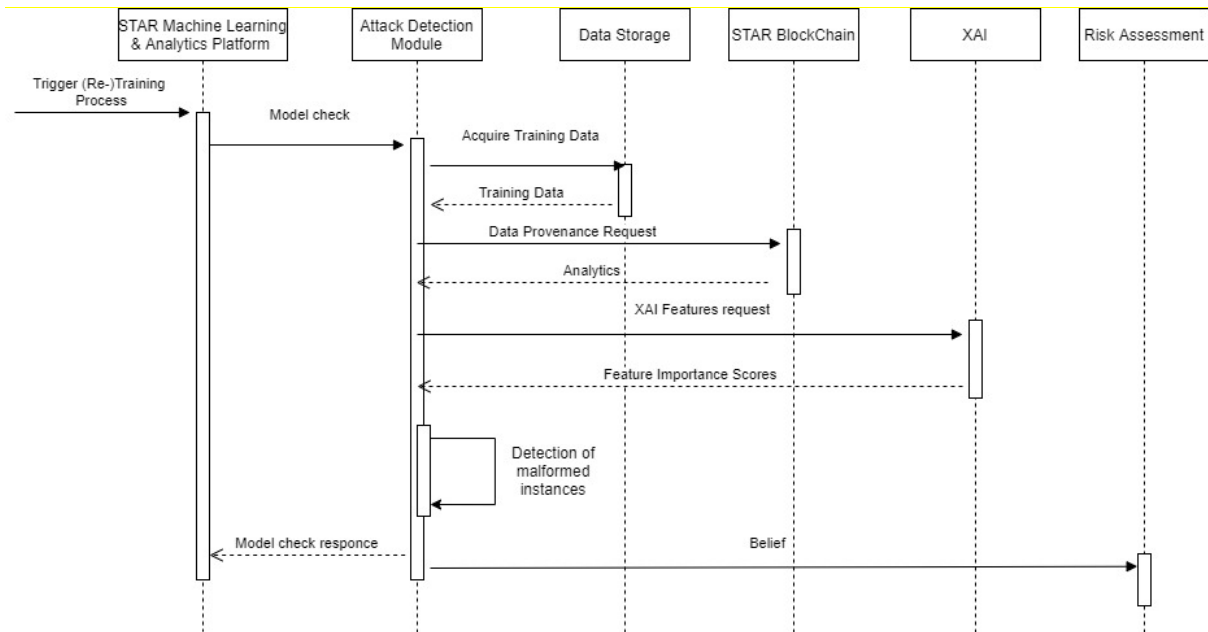


Figure 7: Information Flow for a Defending a Poisoning Attack

### 2.3.3 Defending an Evasion Attack

Figure 8 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 by 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.

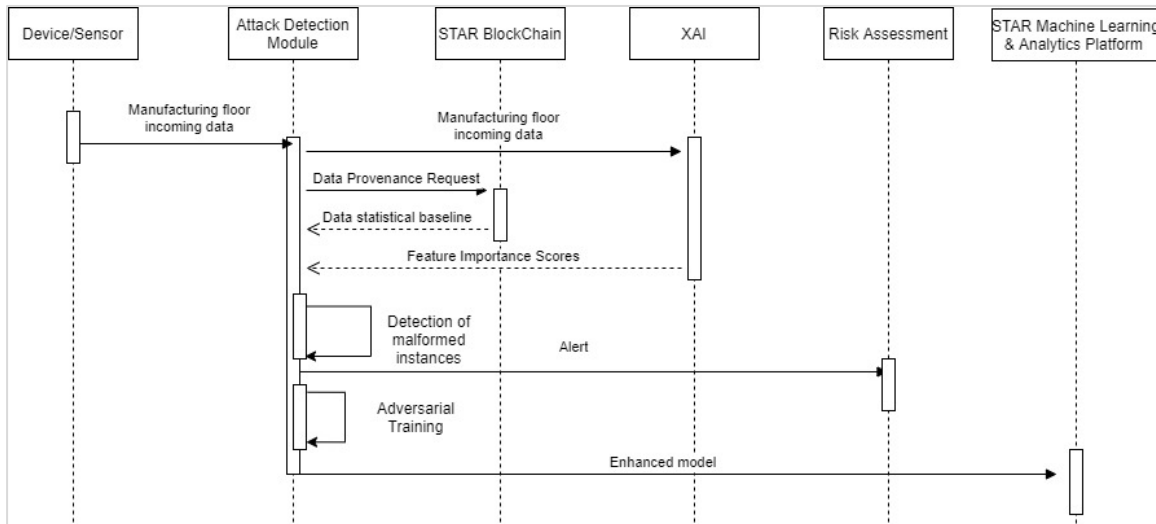


Figure 8: Information Flow for a Defending an Evasion Attack

### 2.3.4 Dynamic Management and Configuration of Data Sources

Figure 9 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 ensure 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.

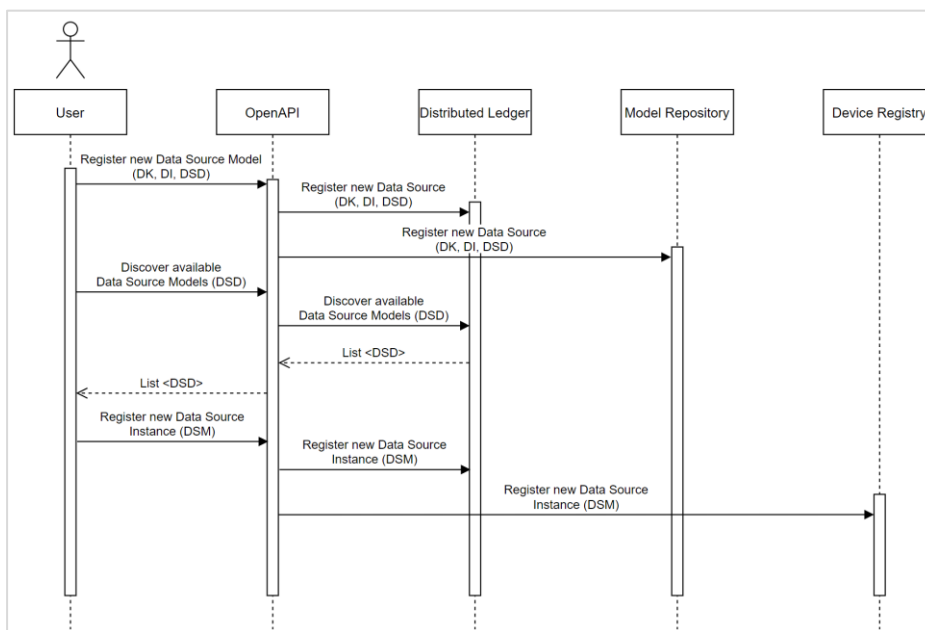


Figure 9: Process View of a Data Source Management Use Case

### 2.3.5 Dynamic Management and Configuration of Analytic Processors

In the same logic as section 2.3.4 above Figure 10 below provides the high-level sequence of interactions for a user/client to persist a Processor configuration to the distributed ledger network and model/registry repositories of the STAR solution.

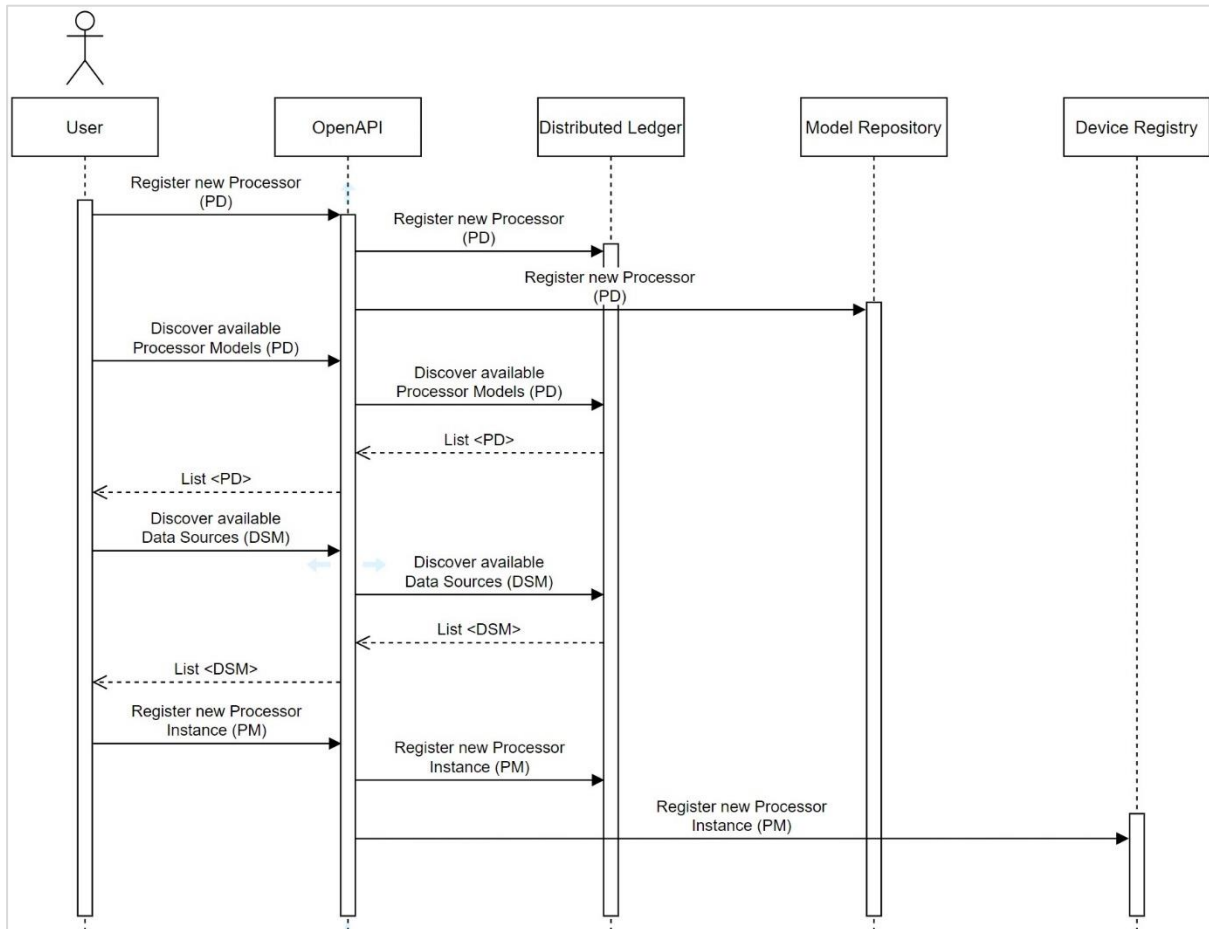


Figure 10: Processor persistence to Distributed Ledger network

As we can see in Figure 10 above the Processor configuration is persisted in parallel to the Distributed Ledger network and the conventional repositories. This is because the Distributed Ledger is used auxiliary to the existent repositories and persistence mechanisms for provenance and validation of the specified data sources.

### 2.3.6 Metadata Validation using the STAR Blockchain

Once the provenance and traceability information have been persisted on the blockchain (see sections 2.3.4 and 2.3.5 above), they can be queried by any organization/service of the STAR platform in need to verify their authenticity. Such queries might serve different reasons like data audits or cross-verifications before the client service proceeds to some form of actuation on the field. The sequence of interactions following a query is depicted in Figure 11 below.

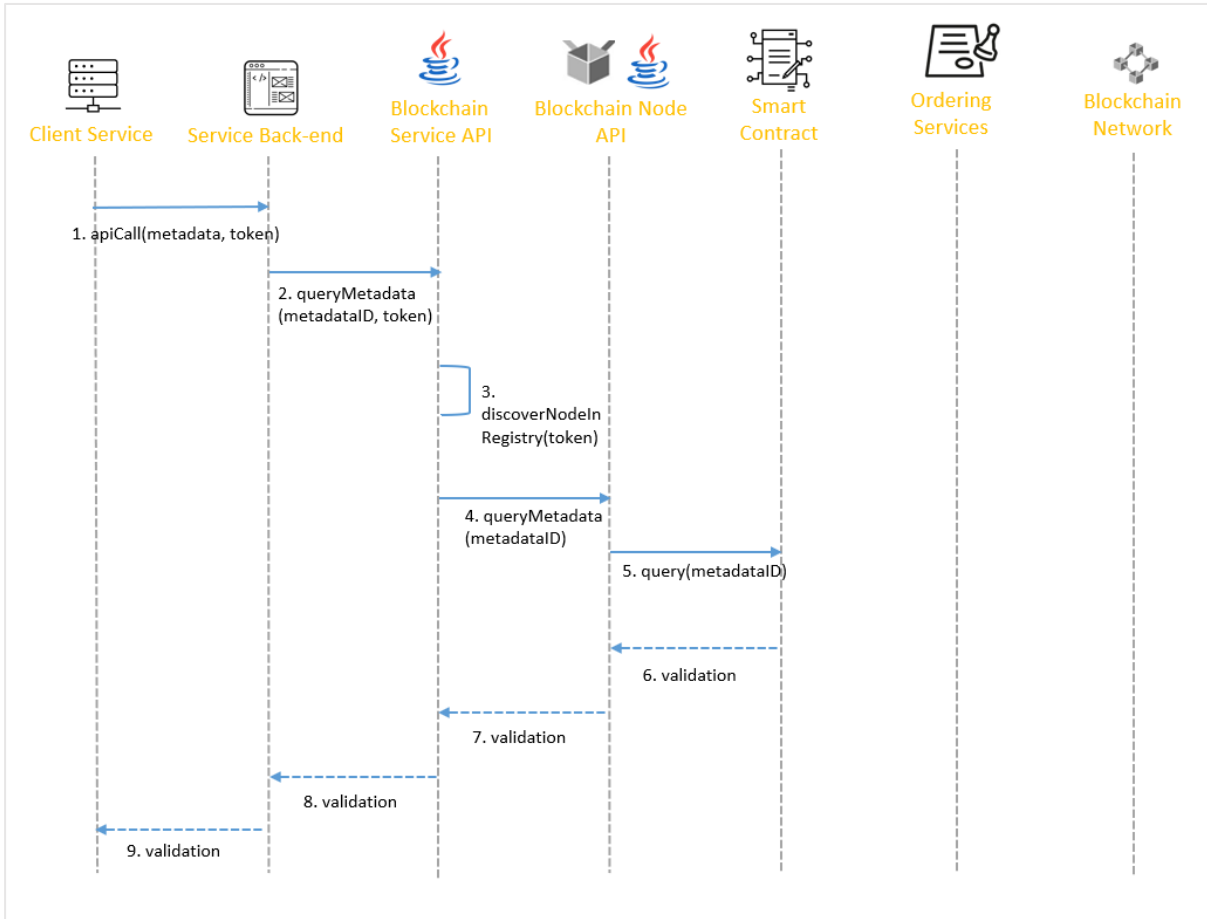


Figure 11: Metadata Validation using the STAR Blockchain

### 2.3.7 Probe Data Transformation and storage

In Figure 12 we can find a sequence diagram providing the flow of initiating and storing measurements from the monitored system. We see that the User initiates the process (starts the Probe) through the probe management wrapper. Then it is performed a continuous loop of the Probe collecting the required measurements, pushing them to the data routing wrapper and persisting them to the Monitoring Data Storage and Data bus.

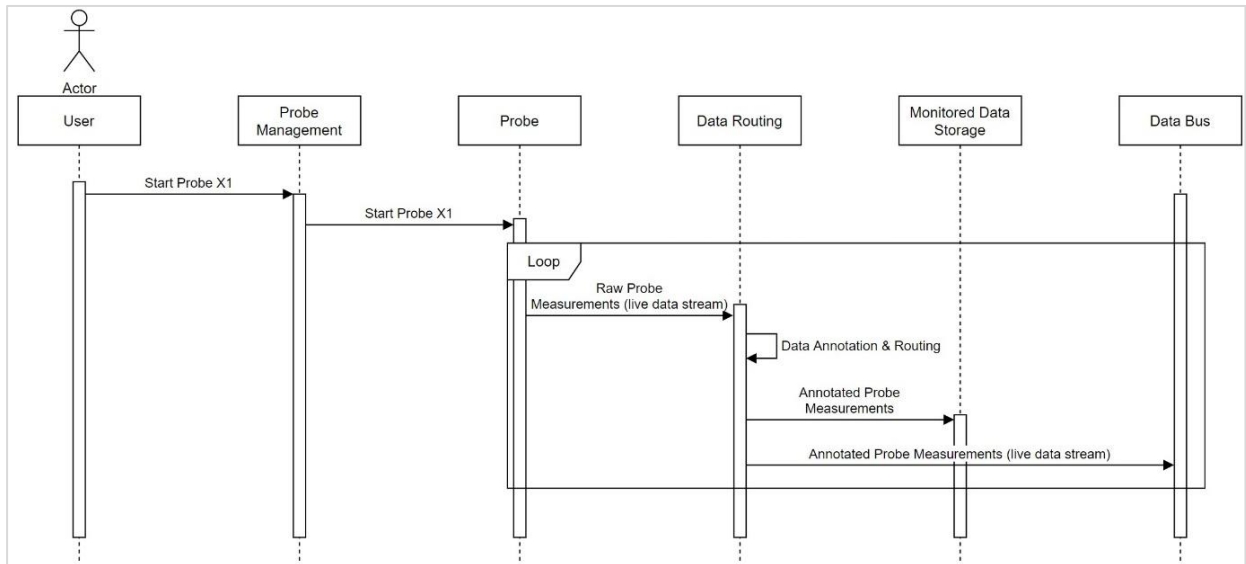


Figure 12: Probe data storage sequence diagram

### 2.3.8 Security Policy Management

Figure 13 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).

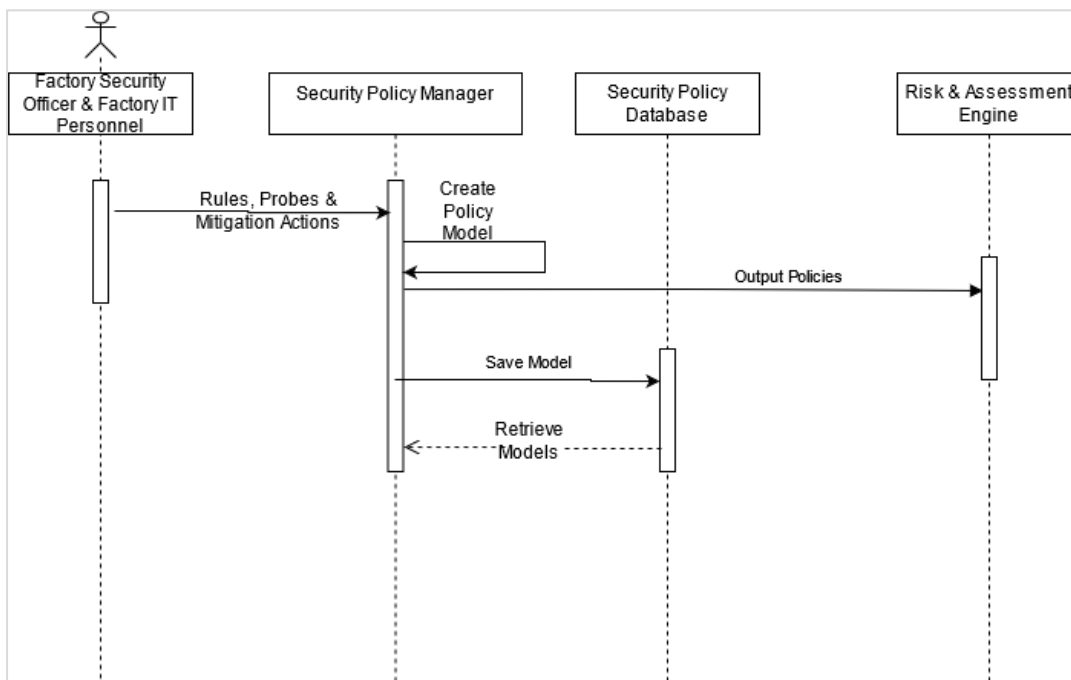


Figure 13: Process View of a Security Policy Management Use Case

### 2.3.9 Human Centric Digital Twin

The sequence diagram depicted in Figure 14 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 Functional 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.

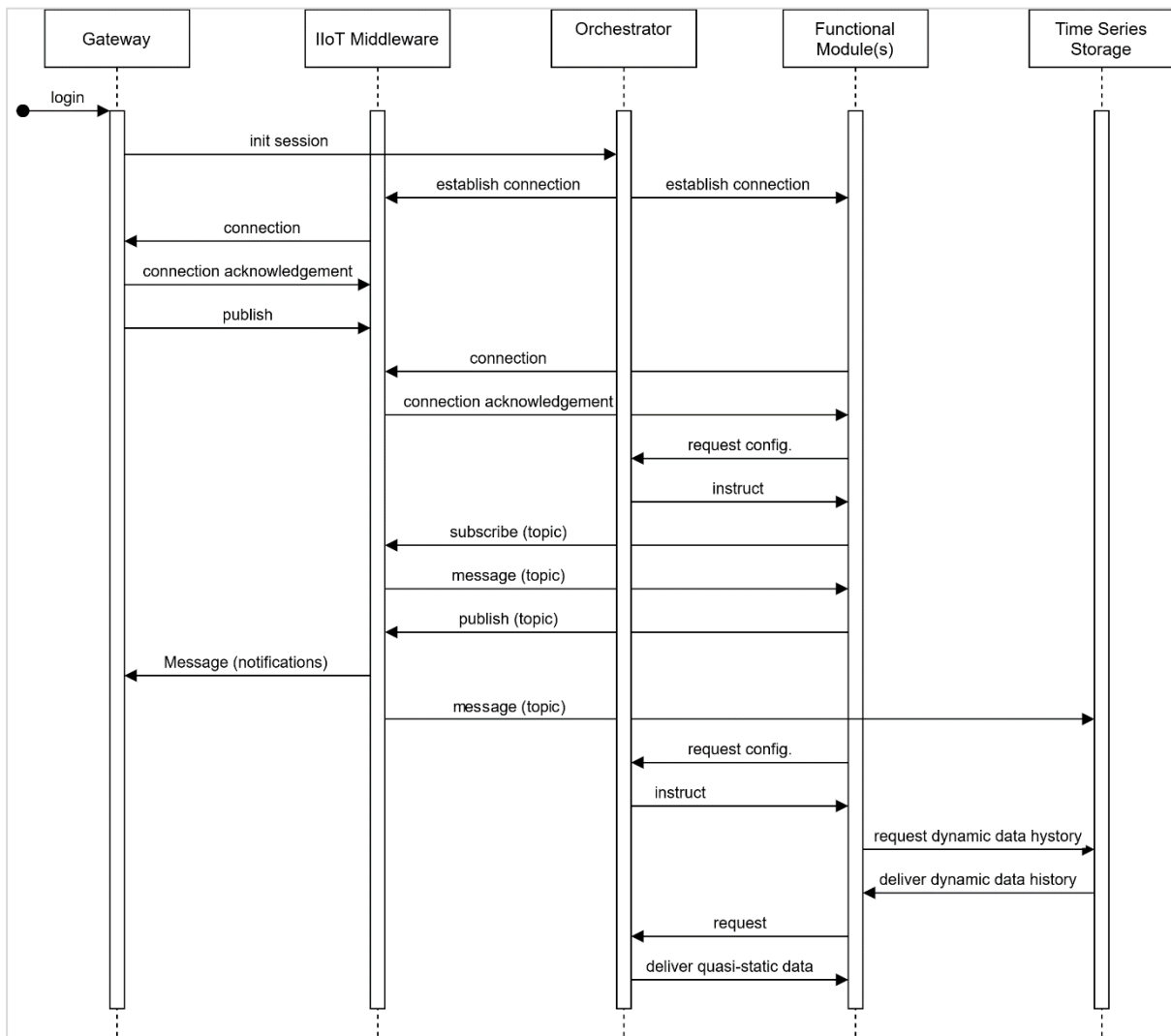


Figure 14: Process View of the Human Centric Digital Twin Operation

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.

### 2.3.10 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 15 illustrates how STAR XAI will produce counterfactual logic, while Figure 16 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.

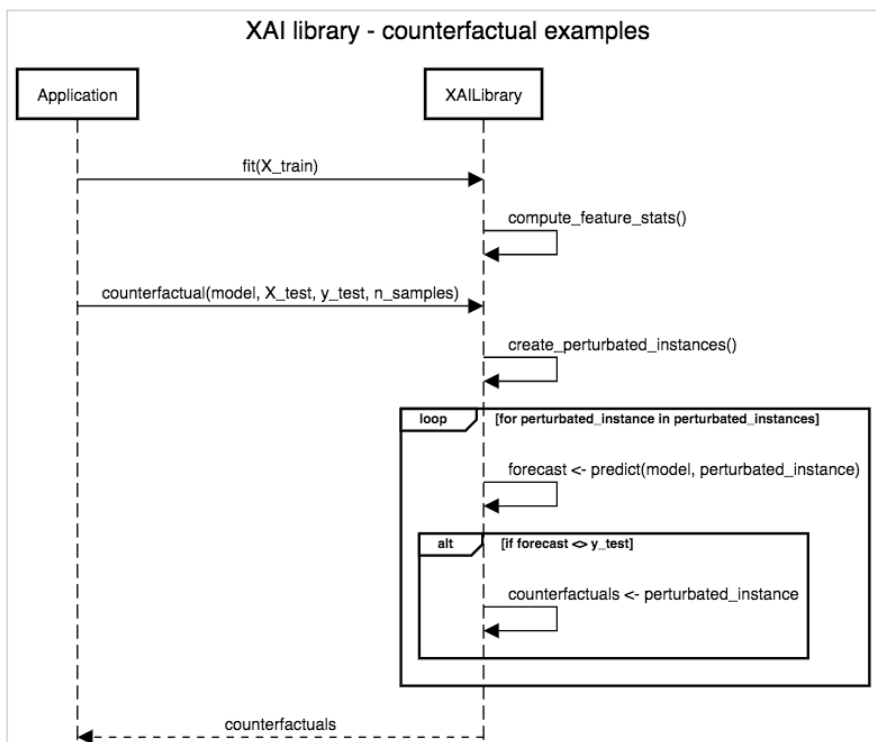


Figure 15: Provision of Counterfactuals Information by the STAR XAI

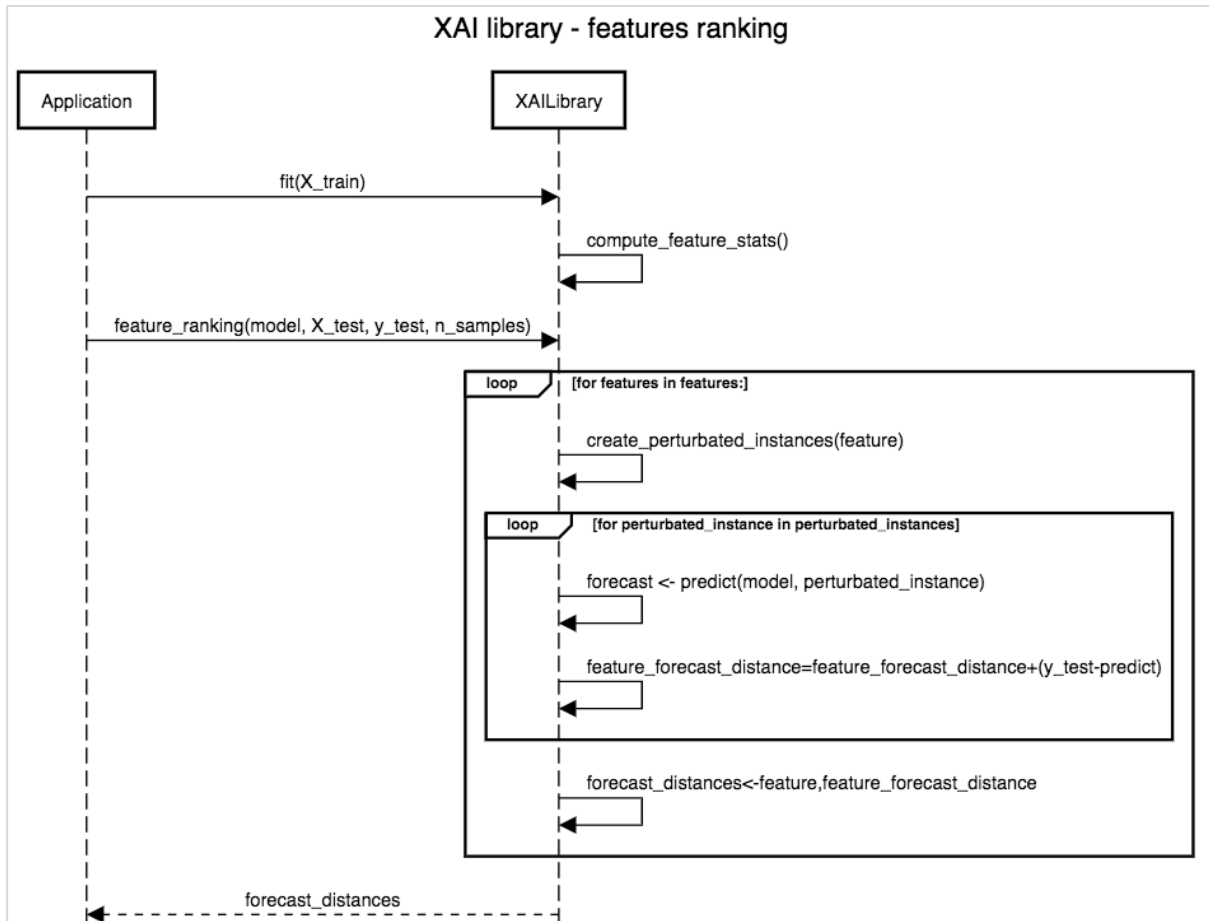


Figure 16: Features Ranking Operations by the STAR AI

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.

### 2.3.11 Active Learning for Human Robot Interactions

Figure 17 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).

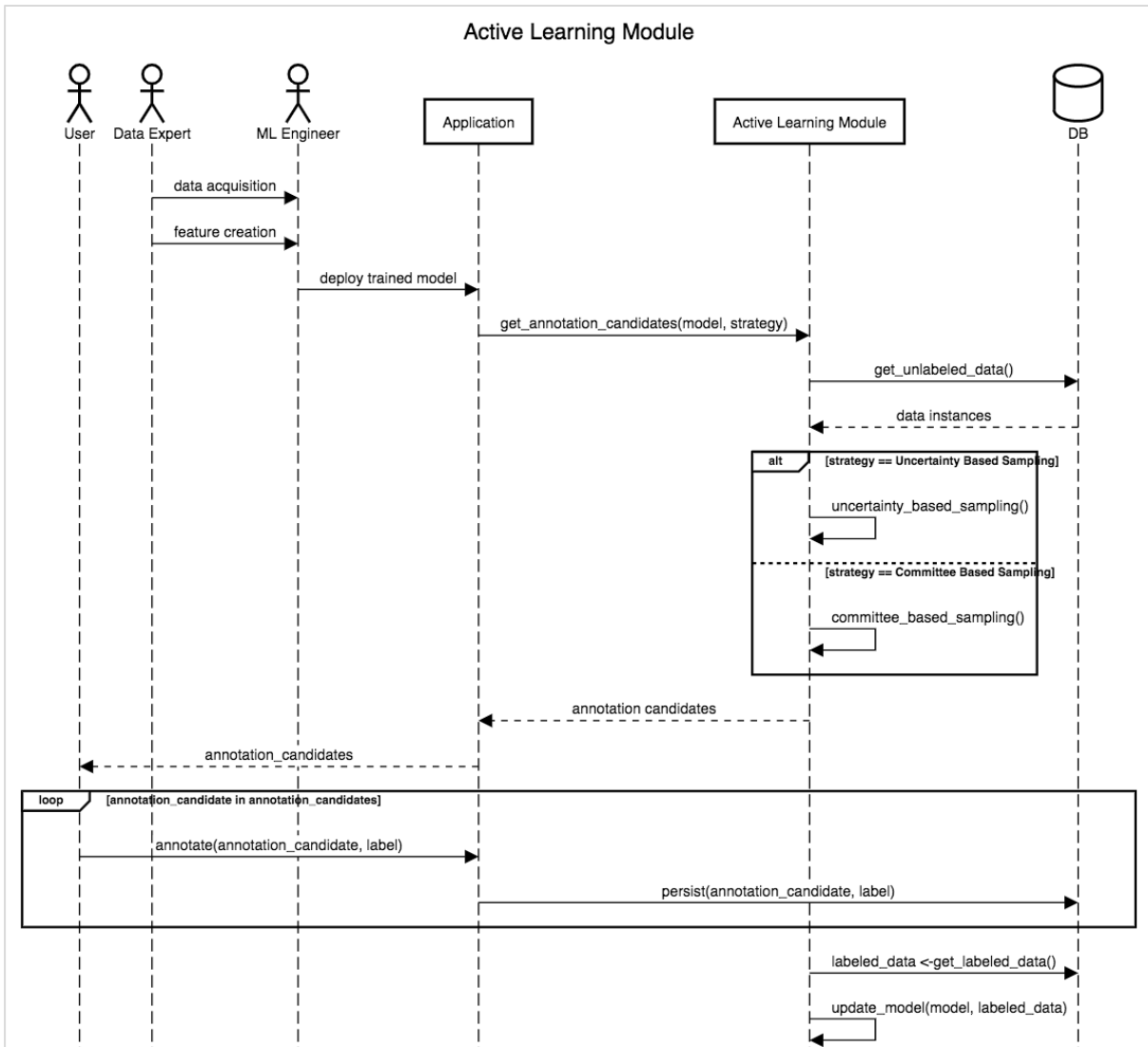


Figure 17: Active Learning Module Operation in support of Human Robot Interaction

### 2.3.12 Feedback Module Operation

The UML diagram of Figure 18 presents the operation of the feedback module, including interactions between the end-user, analyst and the Industry 4.0 (or Industry 5.0) 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.

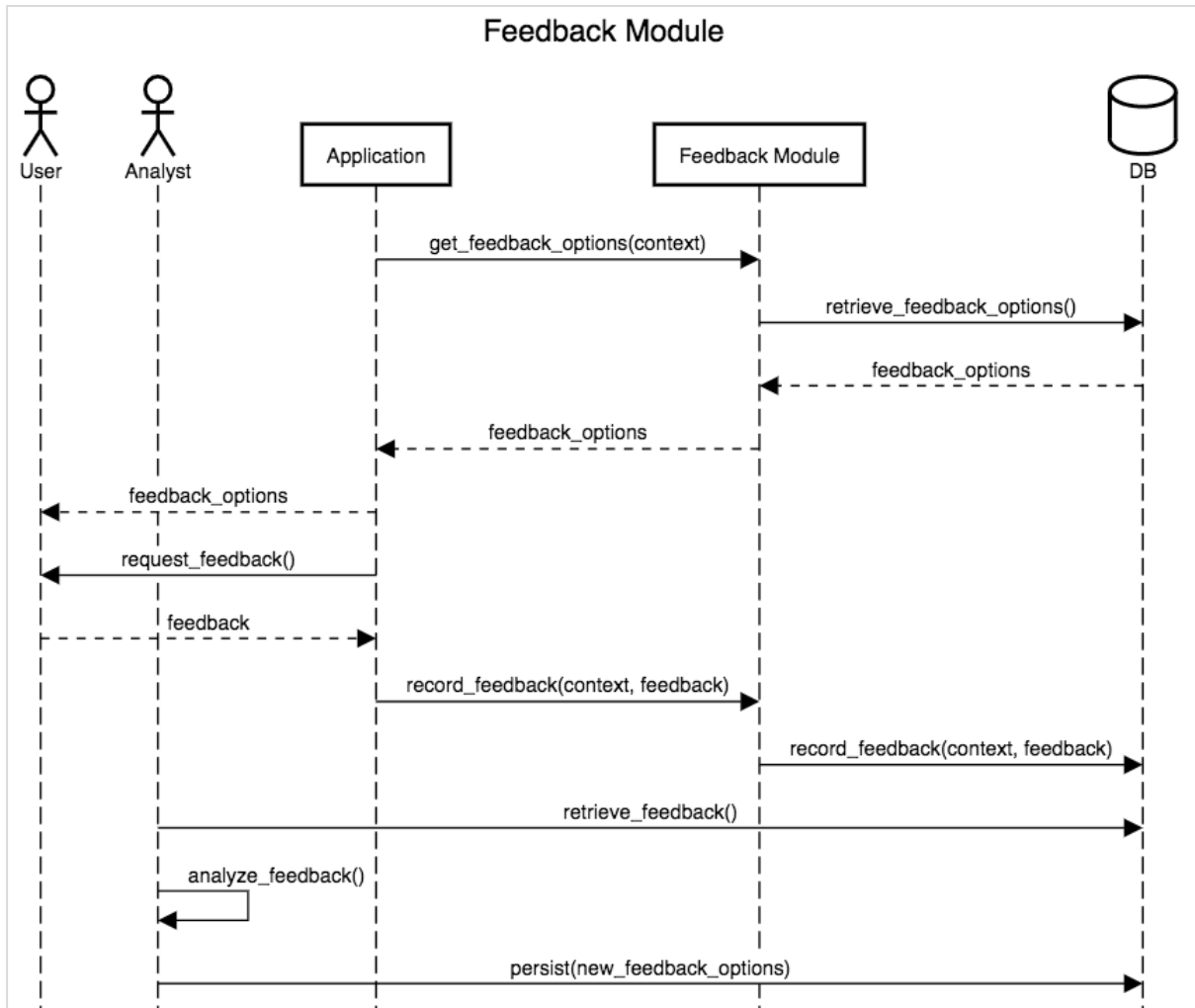


Figure 18: Operation of the Feedback Module

### 2.3.13 NLP Interaction

The NLP module provides one of the most user-friendly ways of interacting with AI systems and robots. Its operation is illustrated at a high level in Figure 19. The diagram illustrates how the NLP module provides feedback to JSI’s CuriousCat system, which will support the implementation of the AL system.

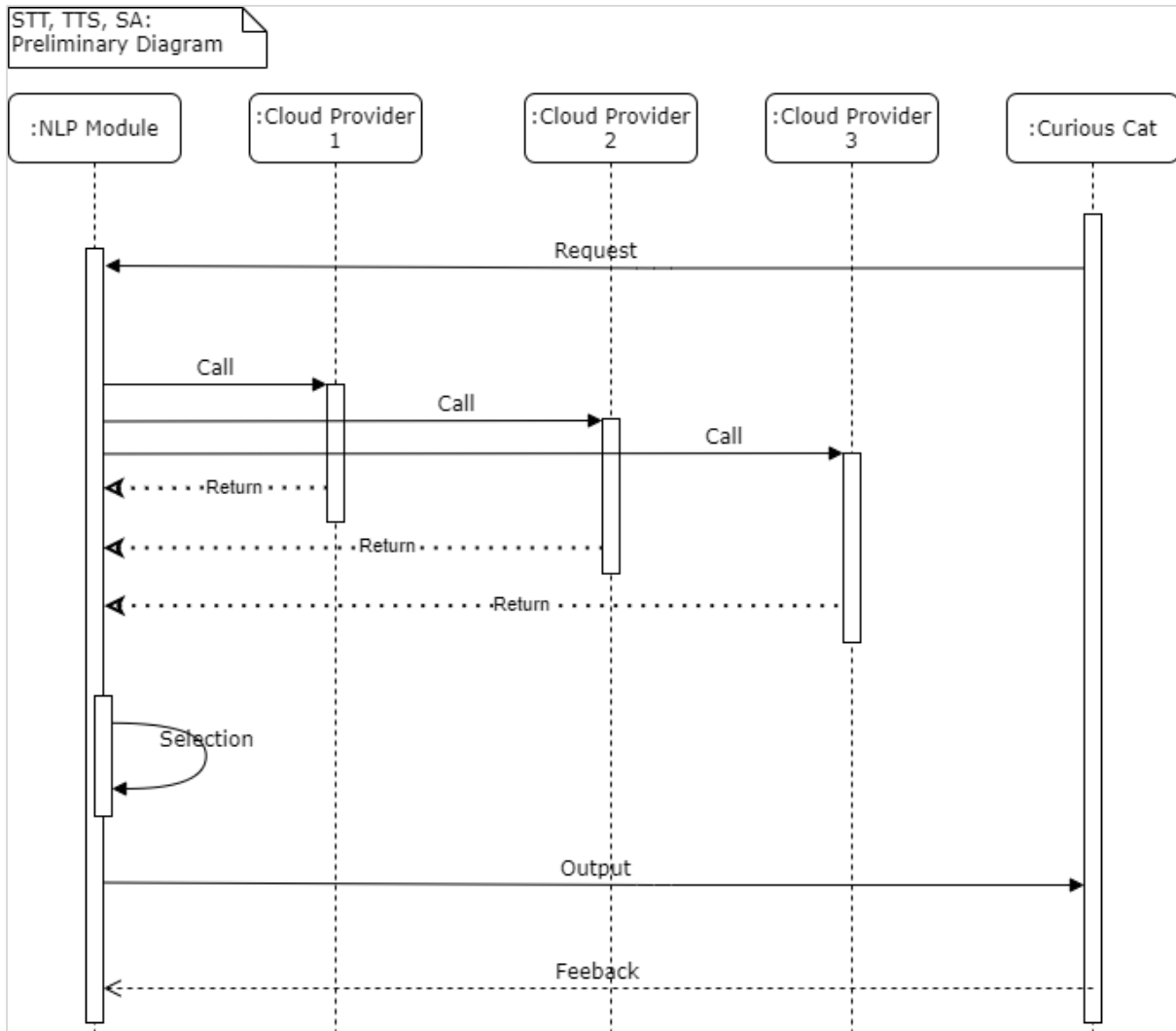


Figure 19: High Level View of the NLP Operation in the context of the STAR Implementation

## 2.4 Physical View(s) of STAR Architecture

### 2.4.1 Deployment Overview

The physical deployment of the STAR system is generally 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 far edge (e.g., an embedded device or machinery if applicable).

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

| Features                 | Cloud | Edge   | Far Edge |
|--------------------------|-------|--------|----------|
| 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 labelled 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   |

### 2.4.2 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.

### 2.4.2.1 Software Packaging with Docker images.

For the STAR software packaging, we have considered Docker<sup>1</sup> images which is currently the dominant technology 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 way the 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 which 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 a checklist on how to "dockerise" any application. Specifically, the following steps are suggested:

- 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.

### 2.4.2.2 Container Tool with Docker Compose.

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 will facilitate its installation. Additionally,

---

<sup>1</sup> <https://docs.docker.com/>

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 Compose<sup>2</sup>.

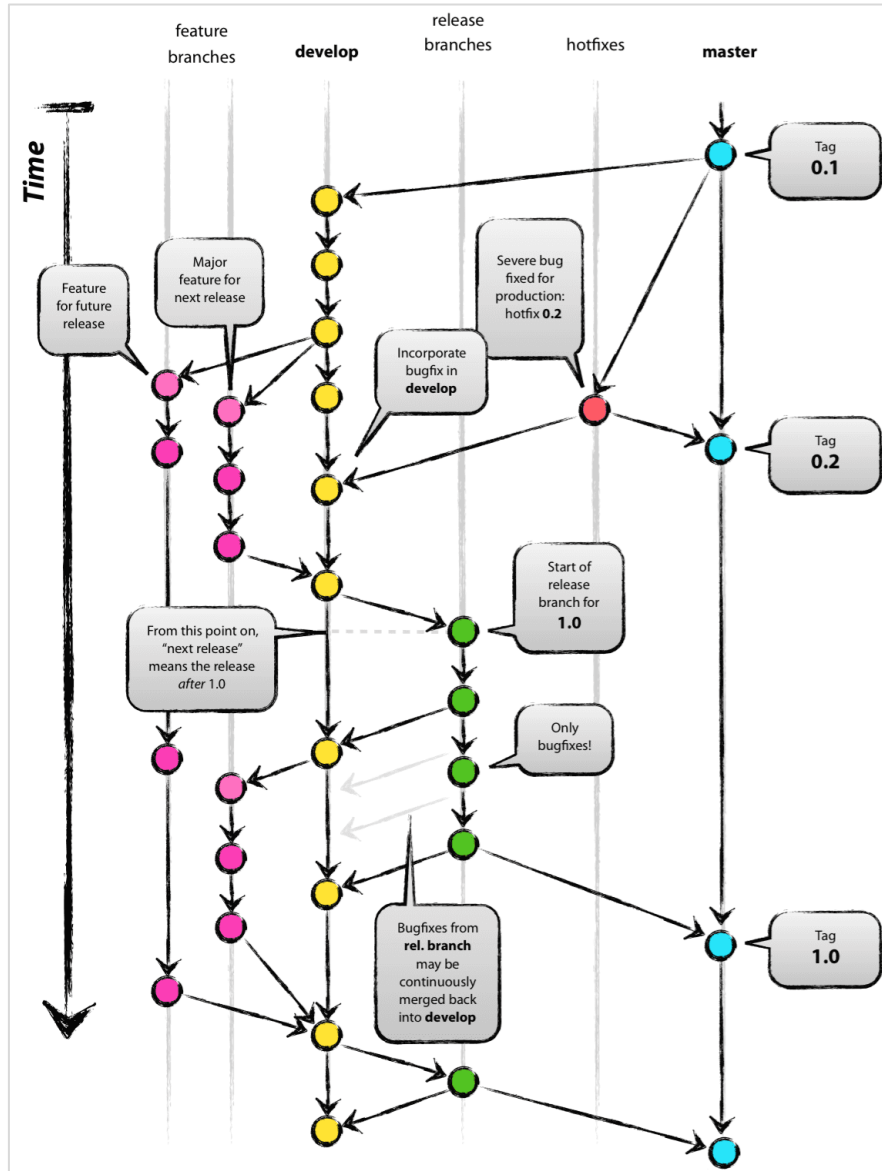


Figure 20 A Complete Git branching model<sup>3</sup>

### 2.4.2.3 Code Management with GitLab

STAR component's code management is based on two popular open-source technologies, Git and GitLab<sup>4</sup>. 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

<sup>2</sup> <https://docs.docker.com/compose/gettingstarted/>

<sup>3</sup> <https://nvie.com/posts/a-successful-git-branching-model/>

<sup>4</sup> <https://gitlab.com/>

several built-in features. It allows the creation of 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"<sup>5</sup> and a complete version of which is shown in Figure 20.

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.

#### 2.4.2.4 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 [Kisler21].

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.

#### 2.4.2.5 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 Portainer<sup>6</sup>.

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

---

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

<sup>6</sup> <https://www.portainer.io/products-services/portainer-community-edition/>

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.
- 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 Deployment<sup>7</sup> documentation. General documentation along with user and configuration guides can be found in Portainer's Documentation<sup>8</sup>.

#### 2.4.2.6 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 an 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 Keycloak<sup>9</sup>. 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 page<sup>10</sup>. Moreover, all the information related with the Keycloak functionalities, deployment and usage can be found at Keycloak's documentation<sup>11</sup>.

#### 2.4.3 Physical Views of STAR Cybersecurity Modules

In this section, we describe the physical view (i.e., the deployment diagram) of the STAR security and data governance for AI Systems in Manufacturing infrastructure. Figure 21 below depicts the complete WP3 deployment diagram combining all the individual components which comprise the solution. As we can see the solution utilizes a minimum of 7 VMs (Application Servers) to offer it with different resources allocated for each of the components. In the subsections below we describe individually the diagram per component.

<sup>7</sup> <https://portainer.readthedocs.io/en/stable/deployment.html>

<sup>8</sup> <https://portainer.readthedocs.io/en/stable/#>

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

<sup>10</sup> <https://www.keycloak.org/getting-started/getting-started-docker>

<sup>11</sup> <https://www.keycloak.org/documentation>

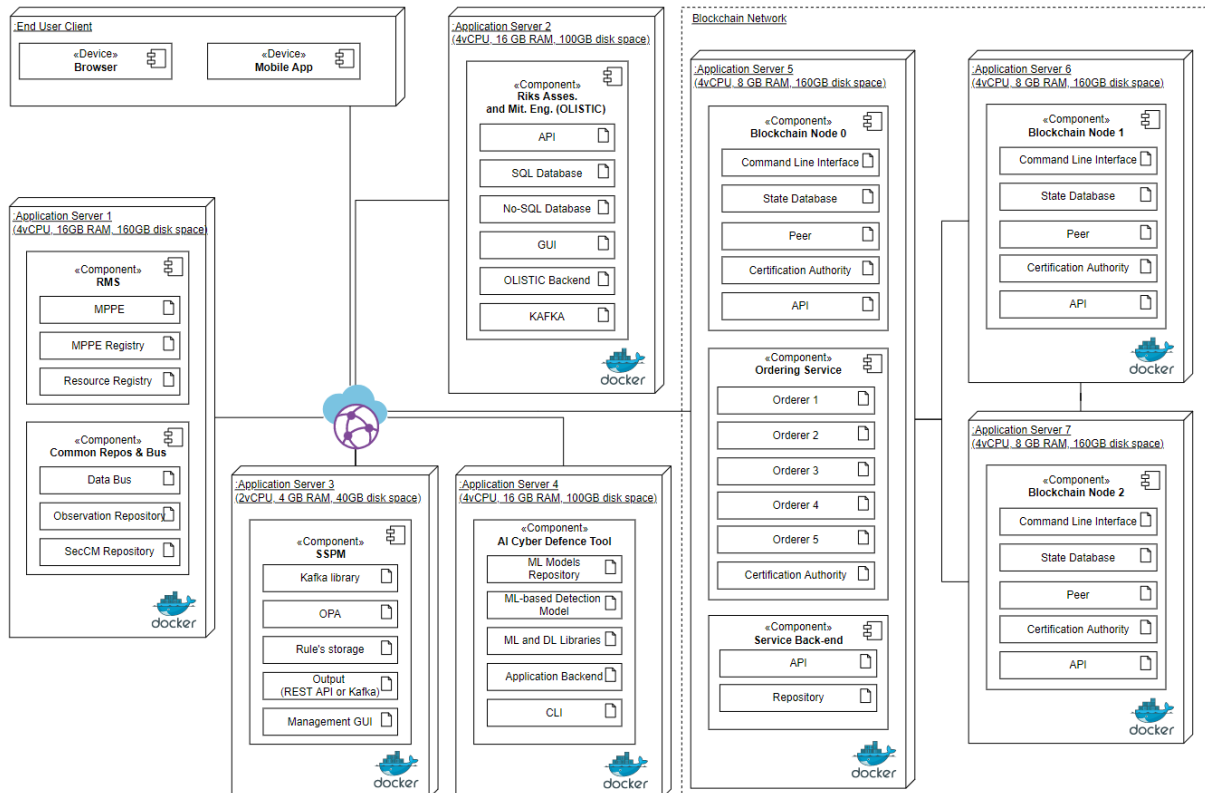


Figure 21: Deployment Diagram for the Cybersecurity Modules of the STAR Architecture (i.e., modules developed in WP3)

### 2.4.3.1 Physical View of DLSDR Infrastructure

The deployment plan for the Distributed Ledger Services for Data Reliability solution, based on the needs of a STAR MVP proof-of-concept architecture, it has been assumed that only three blockchain stakeholders participate on the network, holding peer roles. To stay faithful to the decentralization principles and to showcase how to handle future scaling, the components belonging to each organization have been deployed in different virtual machines, whose networking and orchestration functionalities are being handled by formulating a Docker Swarm cluster. Note that more implementation and deployment details about the DLSDR are provided in STAR D3.1. This distribution of the containerized components into machines is visualised in Figure 22 and Figure 21.

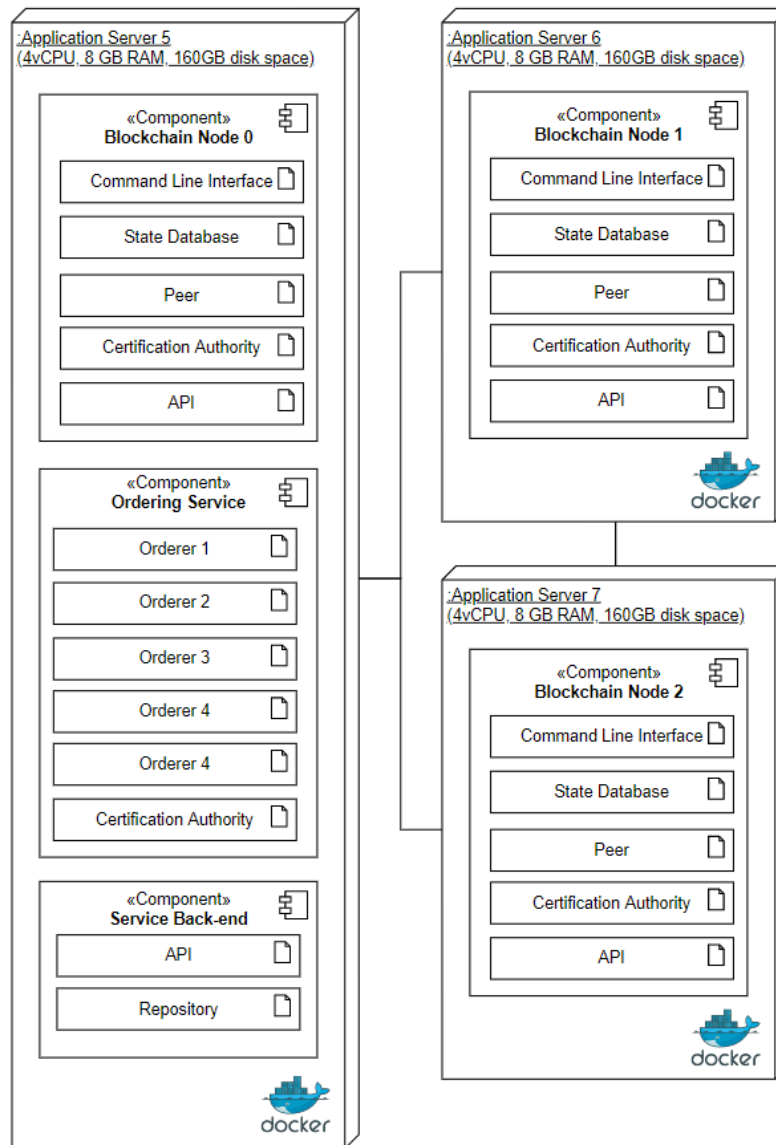


Figure 22: DLSDR Deployment Diagram

### 2.4.3.2 Physical View of Runtime Monitoring System (RMS)

For the deployment of the RMS, we have identified one application server (see Figure 23), which will host all the different RMS components but also the common repositories and data buses of the STAR Cybersecurity layer. The Application server exposes its interface as a service and can be accessed by external devices like Browsers or other third party components (see Figure 21).

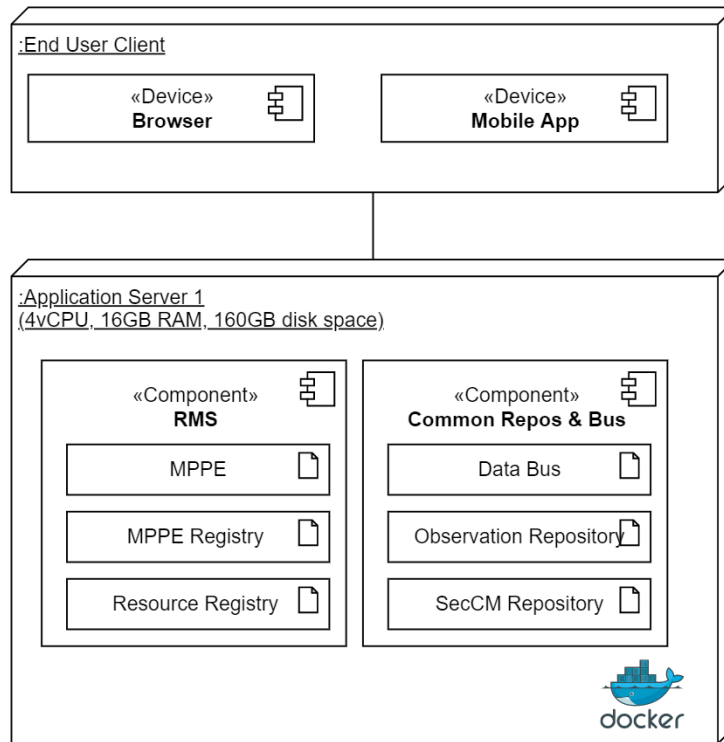


Figure 23: RMS Deployment Diagram

### 2.4.3.3 Physical View of Cyber-Defence Strategies

For the deployment of the AI Cyber Defence tool that enables STAR to apply AI cyber defence strategies against poisoning and evasion attacks, we adopt the deployment approach illustrated in the deployment shown in Figure 24. According to the figure, the tool is a dockerised application which is composed of core elements that operate in a synergistic manner to deliver a detection service for poisoning and evasion attacks. More specifically, a backend python-based application aims to the orchestration of all the individual sub-components and is also responsible for enabling the communication with the rest of the STAR components, either by retrieving information or by sharing the alerts upon the detection of possible attacks. The backend handles an ML-based Detection model, which is responsible for detecting the attacks. The model is supported by the ML and DL libraries which are used for data pre-processing and for empowering the models’ training and online operations. The ML/DL models are saved in a persistent volume to enable the change of configuration depending on the use case to be validated. The user can interact and acquire information on the operational status through a Command Line Interface. The sharing of the alerts is supported by the Data Bus deployed on the application server offering the common repositories and repos for WP3 tools.

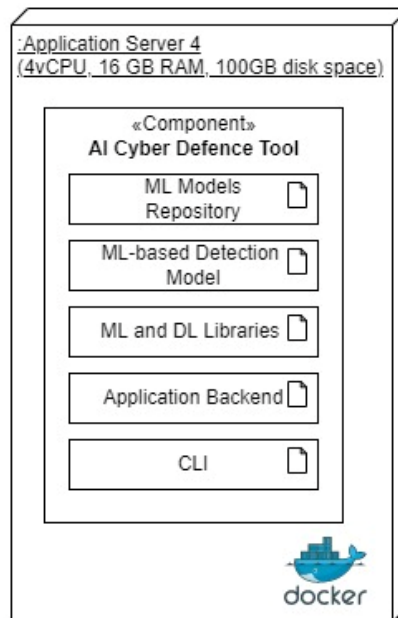


Figure 24: AI Cyber Defence Deployment Diagram

#### 2.4.3.4 Physical View of Policy Manager

Following paragraphs present the physical view of the security policy manager (SSPM). The main modules of the tool are reported, namely the backend modules: the Kafka library; the Open Policy Agent (OPA) engine for the policies creation; the repositories of the rules related to the policies created and the output, which will be conveyed or through REST API or through another Kafka queue, this will be decided in the next phases of the deployment. This decision is based on the necessity to keep (or not) a history of the outputs.

The last block is the representation of the frontend and the GUI, which features will be defined in the next project phases.

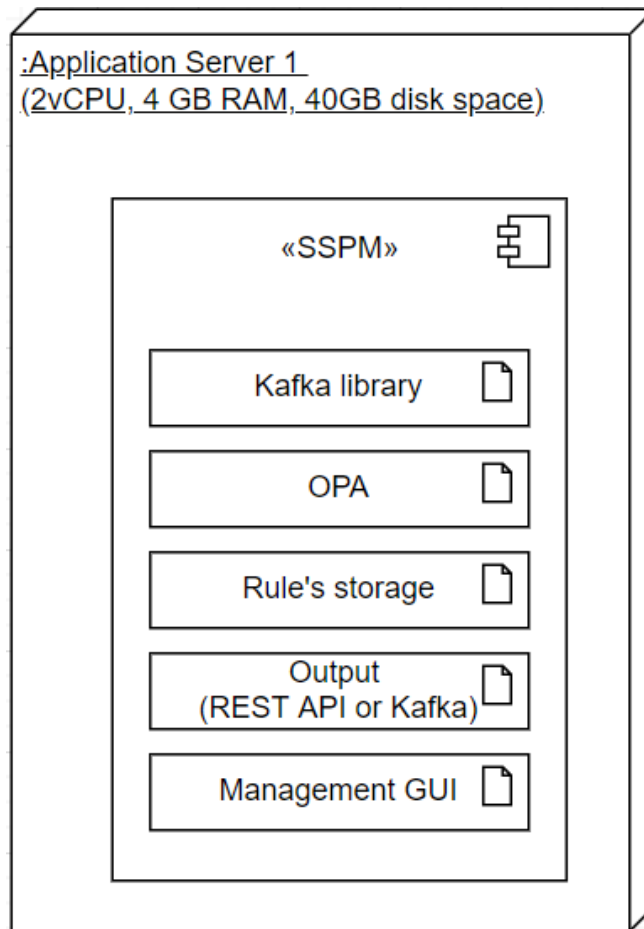


Figure 25: SSPM Physical view

#### 2.4.3.5 Physical View of Risk Assessment and Mitigation Engine (OLISTIC)

The Risk Assessment and Mitigation Engine is the STAR technical artifact that will enable a factory security officer to manage the security events detected on the assets comprising the production lines of the factory. To do so, UBITECH’s Risk Management tool, namely OLISTIC, is deployed in order to offer this STAR service. As can be seen in Figure 26, a dedicated application server is used to host OLISTIC. The latter comes as a complete tool that consists of the Frontend (GUI) to enable the security officer to interact with the tool and get vital information for the security status of the monitored systems, as well as a backend that orchestrates the risk assessment operations and the communication with other STAR tools. A complete API is provided in order to enable data exchange with other tools, while both SQL and No-SQL solutions are used for internal storage. The risk assessment output is shared with other components through Kafka. All the above are offered via a dockerised application deployed on the Application Server 2, as illustrated in Figure 26.

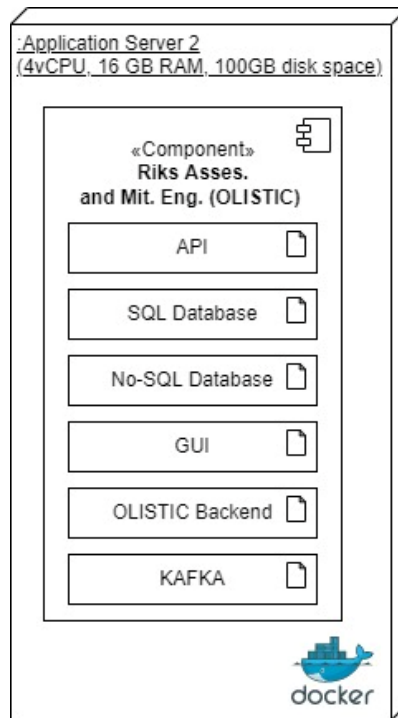


Figure 26 OLISTIC Deployment Diagram

## 2.4.4 Physical Views of STAR Active Learning and XAI Modules

The source code regarding the Active Learning and XAI modules is managed in private git repositories. In the future, we expect to expose the best machine learning models and the active learning flow through a specific set of microservices. The services will offer the machine learning models functionality through a REST API. Furthermore, in line with the technology stack described above, we expect to containerize them in Docker images. The Docker images will be published and stored in the JFrog Artifactory binaries repository. We will ensure traceability between the source code and published Docker images by including the commit hash as part of the released artifact version number. A Docker compose file will be used to manage the required dependencies against external services, databases and the required infrastructure supporting our services. The services are deployed either on premise or in the Cloud.

### 2.4.4.1 Physical View of Active Learning Module

For the development and deployment of the Active Learning module, we adopt the deployment and ecosystem management technologies described above. We envision the Active Learning service as a use case agnostic service that provides a catalog of active learning strategies. The strategies can be approach-specific (e.g., methods can differ whether they are suitable for supervised or unsupervised machine learning). In the scope of the STAR project, we expect to mainly develop active learning strategies suitable for supervised machine learning. We depict the interaction between a Gateway Service requesting some specific functionality from the STAR platform, a particular service implementing machine learning models, and the Active Learning service if required by the aforementioned ML-based service.

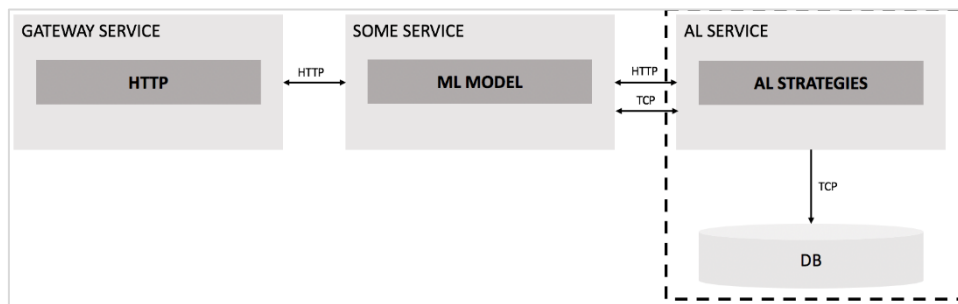


Figure 27 The AL service can be deployed as an on premise or cloud application

The Active Learning service offers a wide range of strategies, which aim to be use case agnostic. It interfaces via the HTTP or TCP protocols with client services, and via TCP with the database, where relevant information regarding each call is stored.

#### 2.4.4.2 Physical View of the Explainable AI (XAI) Modules

The XAI models and library, will be deployed in a containerized microservices way. Specifically, it is proposed to be deployed to a Kubernetes environment as illustrated in the physical view of Figure 28.

Within the context of this deliverable, the first version of PoC (Prototype\_v1) is introduced. Accordingly, the PoC will not implement all the proposed XAI components (i.e., XAI for timeseries, images, text, and tabular data) (see D4.1), but just a selected subset.

In more detail, the blueprint testbed implementation is based on the following assumptions:

- The PoC version (Prototype\_v1) does not fully represent the XAI component, since some of the specific services are still under development (i.e., XAI for timeseries, XAI for text data and XAI for tabular data).
- Data migrated to the testbed are owned by pilots and it is to be discussed in which way will be ingested or retrieved by XAI.
- Regarding the Data Management and the communication with other components of the platform both Kafka queue and zookeeper, JDBC and REST API connections will be used in terms of data handling.
- The ML/DL models should be saved in the persistent volume provided by the testbed in a way that may be shared among the different components of the platform or shared database/objectstore will be leveraged.

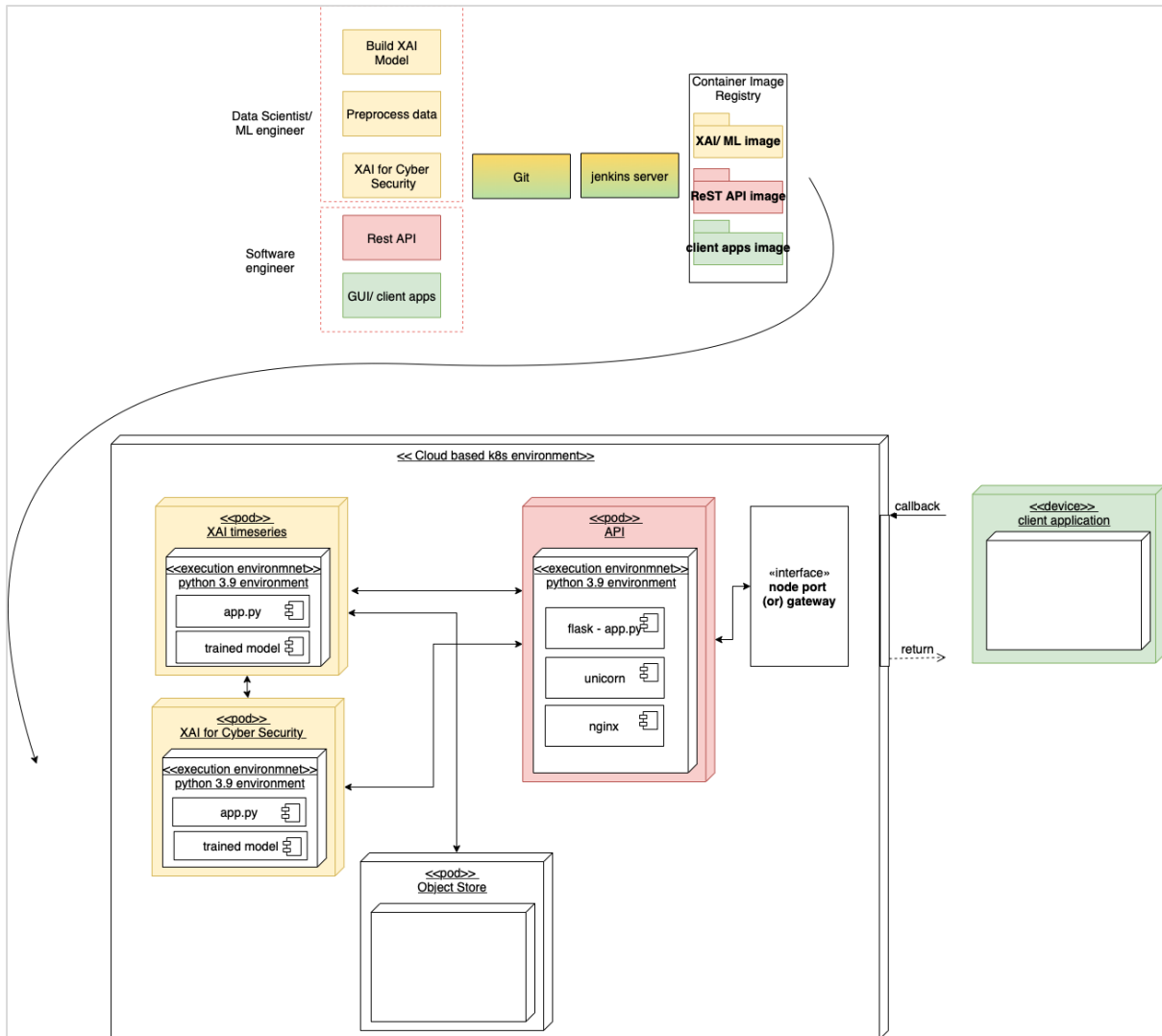


Figure 28: Physical View of the STAR XAI Component/Modules

## 2.4.5 Physical Views of STAR Reinforcement Learning Modules (WP4/WP5)

### 2.4.5.1 Physical View of Safety Zones Detection Module

The solution to ensure safe movement of AMRs among workers relies on two modules:

1. Safety Zone Detection.
2. AMR Fleet Optimizer.

The first one relies on cameras deployed in the factory (i.e., the DFKI / SmartFactory lab in WP6 Pilot), to extend the perceptions of AMRs, which allow planning of the overall fleet movements thanks to the second module, taking into account workers movement.

The Safety Zones Detection System is composed of several dockers, and a Docker compose file will be used to manage these containers. The communication between the two components will be based on the Human Centric Digital Twins (see Figure 29 and section 2.4.6)

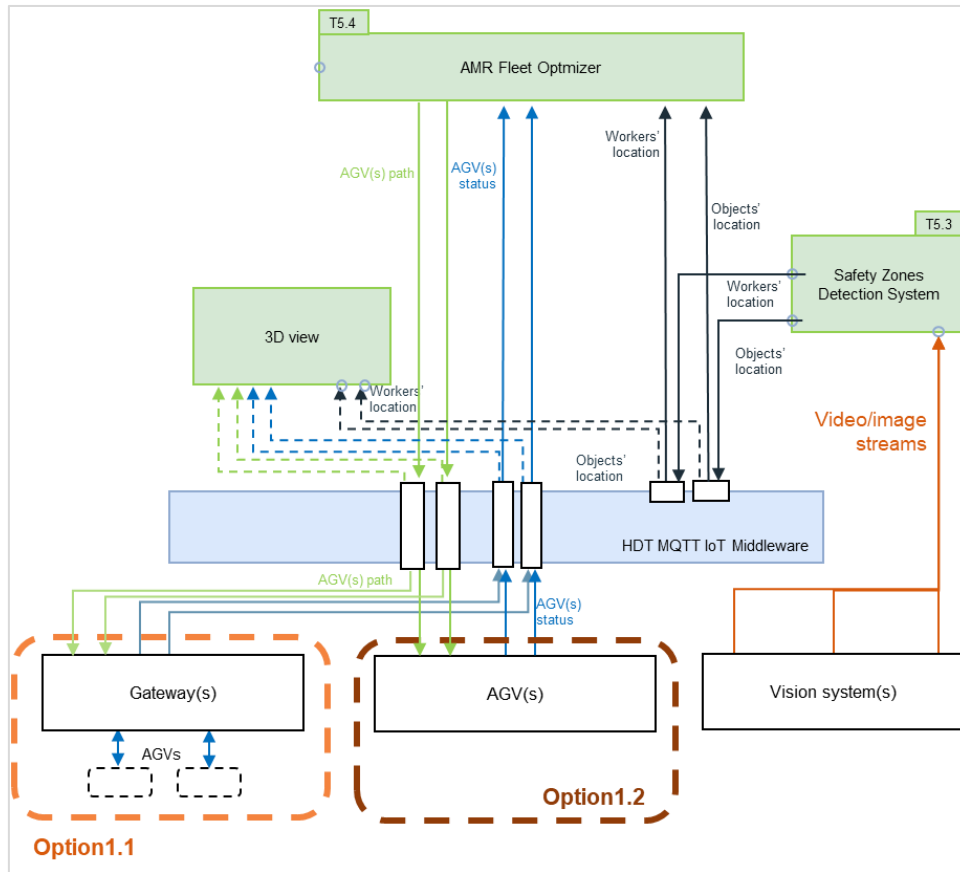


Figure 29: Safety Zone Detection & AMR Fleet Optimizer Logical View with Physical and Deployment Views Considerations

More deeper, the Safety Zones Detection System exploits video footprints as input and will deliver the spatial heatmaps (worker location and object location) as results of the analytics. The elements extractor engine merges two deep learning algorithms, either for the skeleton reconstruction in order to follow the human gesture and pose and the other for the detection and classification of the non-static object in the scene, with a background subtraction module.

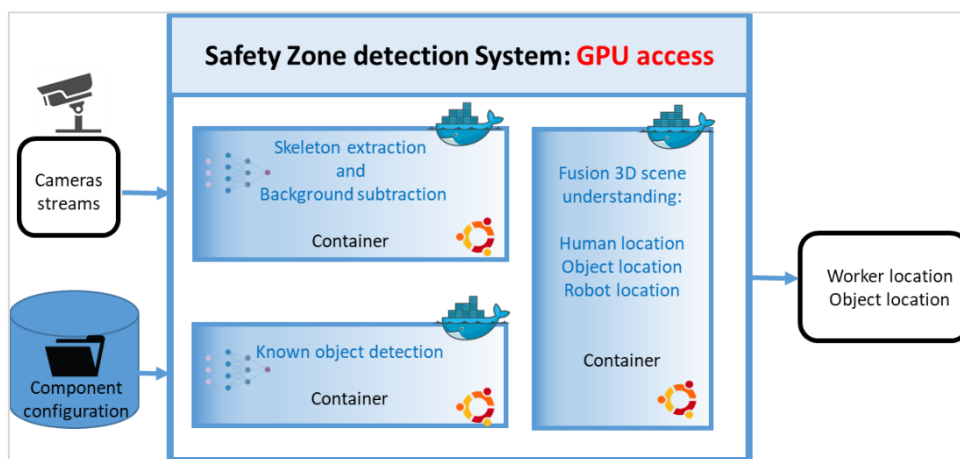


Figure 30: Safety zone detection Logical View with Physical and Deployment Views Considerations

### 2.4.5.2 Physical View of Simulated Reality Module

Currently, the scope of the simulated reality component is to assist in the Automated Quality Inspection use-case of the project where it will generate simulated images to balance defect datasets that suffer from skewed class data. Its scope may be enhanced to cover other use-cases such as object recognition for worker perception and shop-floor resource allocation simulation. Although the deployment diagram provided is an attempt to cover all of these, it is more tailored to the use-case currently under focus (i.e., Quality Inspection) and could be updated to accommodate the other use-cases if need be.

The component will consist both of one (or more to cover additional use-cases) batch jobs, which will be long running jobs such as the training of a Generative Adversarial Network (GAN) for data augmentation, or running a heavy simulation (robotic, resource allocation) to produce synthetic scenarios. The output of these batch jobs will be consumed by live services that will deliver synthetic data upon request, potentially in a way that can also categorize the data as easy or difficult to classify through the Confidence Assessment subcomponent.

All subcomponents will need access to a common data store or shared filesystem containing the Input Dataset and Auxiliary Data needed including weights for pre-trained models. Their outputs will consist of trained generator models that can produce synthetic data upon request (e.g., a trained GAN). These need to be accessible through the Data Serving services (e.g., through a Shared File System). Data Serving should return synthetic data fitting a certain use-case specification upon request together with a potential confidence assessment. This can take place through a REST API.

In terms of infrastructure, the batch jobs and ideally also the services will need access to a GPU which will enable machine learning models built in Tensorflow and Pytorch to run much faster, both during training and inference. Their code should be packaged in Docker containers as this will make managing program specific Python dependencies easier.

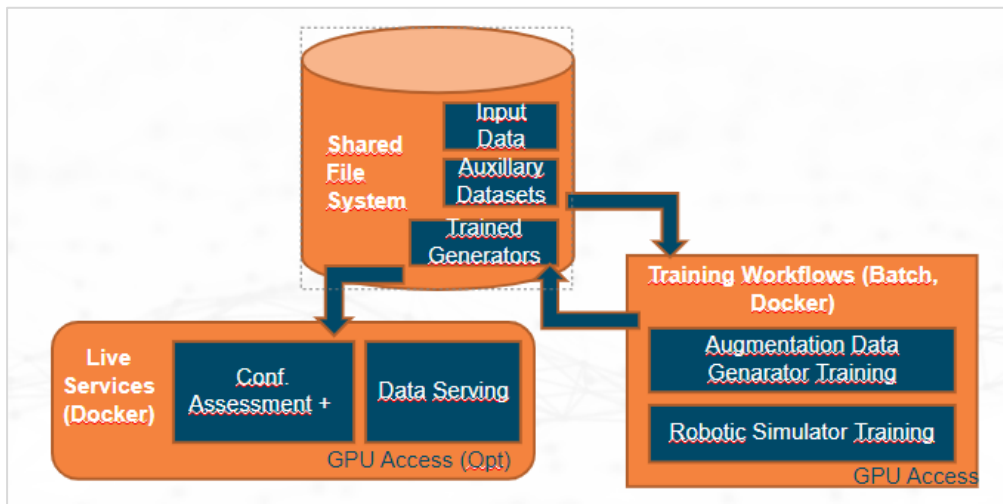


Figure 31: High-Level Physical View of Simulated Reality Module

### 2.4.6 Physical Views of STAR Human Centric Digital Twins

The Human Digital Twin Core Infrastructure (HDT) can be found in the hdt-core repository hosted at Gitlab, by the star-ai project <https://gitlab.com/star-ai/human-digital-twin-core-infrastructure/hdt-core>.

The Docker images of HDT services are hosted at a private Gitlab instance, which requires a token for downloading images. Instructions on how to log into the registry are available within the repository. Access token is provided by SUPSI after a specific request. This repository contains different Compose files to setup different environments:

- dev: the development environment.
- demo: the demo environment, used to demonstrate the HDT during the first technical review of the STAR project in January 2022.

Figure 32 depicts the HDT Core dev deployment to showcase how custom gateways/modules can be integrated with the SUPSI’s HDT.

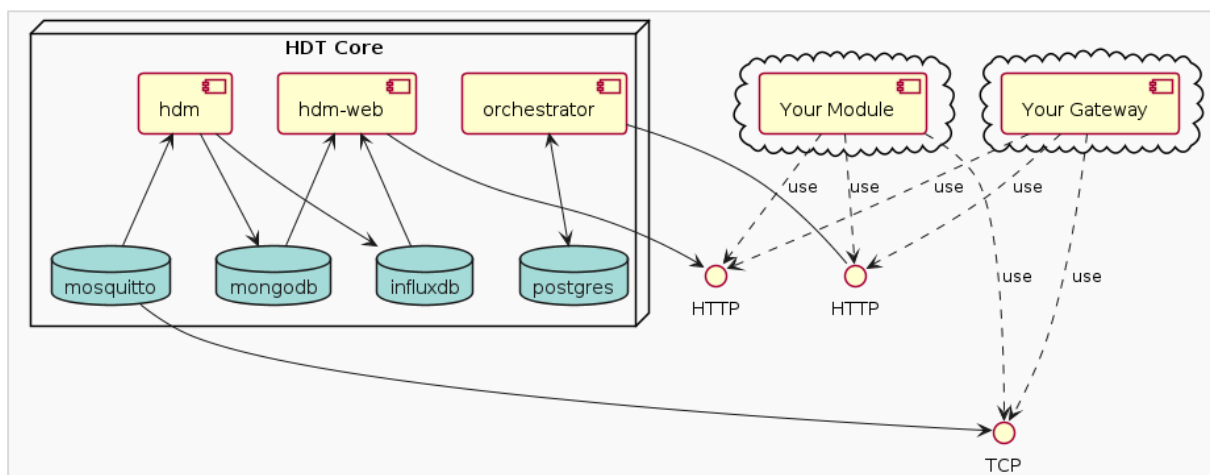


Figure 32: Human Digital Twin Core Infrastructure deployment showcase

The HDT and its components are deployed as cloud applications. Specific instances will be deployed to support the different use-cases of the project (E.g., integration with the R2M’s worker training platform, integration in Use Case #2).

## 2.5 Implementation View of STAR Systems

### 2.5.1 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.

## 2.5.2 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   | Programming Language      | Interfaces to other Modules         | Database                    |
|--|--|---------------------------|-------------------------------------|-----------------------------|
| RMS  | ElasticStack,<br>Kafka                             | Java                      | REST API<br>Apache Kafka<br>Pub/Sub | Elastic Search,<br>Mongo DB |
| STAR Blockchain (DLSDR)                                    | Hyperledge Fabric                                  | Java                      | REST API                            | MySQL, CouchDB              |
| AI Cyber Defense Strategies                                | Adversarial Robustness Toolbox (ART) <sup>12</sup> | Python                    | REST API                            | MySQL                       |
| Risk Assessment and Mitigation Engine (RAME) <sup>13</sup> | Olistic.io / Spring Framework                      | Java                      | REST API<br>Apache Kafka<br>Pub/Sub | MySQL/MongoDB               |
| 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                            |                             |

<sup>12</sup> <https://adversarial-robustness-toolbox.org/>

<sup>13</sup> The implementation with leverage Ubitech’s Olistic.io product

|  |       |        |                             |                           |
|--|-------|--------|-----------------------------|---------------------------|
| Production Processes Knowledge Base    | Linux | Python | REST API                    | noSQL                     |
| Human Digital Twin Core Infrastructure |       | Java   | REST API, Eclipse Mosquitto | Influx, Mongo DB, Postgre |
| Fatigue Monitoring System              |       | Python | REST API                    | Influx, MongoDB           |
| Feedback Module                        | Linux | Python | REST API                    | noSQL                     |

## 3 STAR Architecture Validation – Scenarios Views

To validate the various modules of the architecture, we herewith present the mapping of some of the project’s manufacturing use cases to architecture elements. These use cases are associated with the manufacturing functionalities offered in STAR. Views for the non-functional requirements like cyber-defence requirements addressed in the use cases have been presented in the previous section. Note that the development, deployment and validation of the use cases is a work in progress in WP6 and that relevant specifications are updated based on inputs from the technology development and the co-creation workshops. Therefore, the functionalities of the use cases and their mapping on the STAR architecture are subject to changes and updates that could take place in the following months. Such changes will be documented in relevant WP6 deliverables.

### 3.1 Use Case #1 Human Robot Collaboration for Quality Management

Flexible visual quality inspection is critical to ensure the delivery of good products in the factory. Normally, visual quality inspection systems are trained based on extensive datasets and can be easily optimized due to the mass-production of products. However, when moving to lower-volume production, these extensive datasets are often not available. Therefore, flexible visual quality inspection systems that can be trained based on small, incomplete datasets are needed.

The goal of this use-case is to implement a system that will make setting up automated quality inspections easier & faster by applying AL. The expected result contains a system that can be implemented in the factory to be used to set up an automated quality control for a new product easily (meaning with little to no data) where the production personnel are able to check the products (OK/NOK) and transfer that knowledge to the quality inspection algorithm by means of AL so that an operator can provide his/her input for the first  $x$  products after which the system is able to take over the quality control after a certain amount of input.

By relating this input created by the operators to HDT tools like fatigue monitoring we want to be able to ensure correct inputs and decrease the amount of wrong labelling while also ensuring the (mental) well-being of the operators doing the labelling.

#### 3.1.1 Process View

A high-level process view with relevant information flows for the use case is illustrated in Figure 33. At the heart of the process lies the Active Learning system that uses the operator’s input and information from the HDT to the quality control and optimization algorithm.

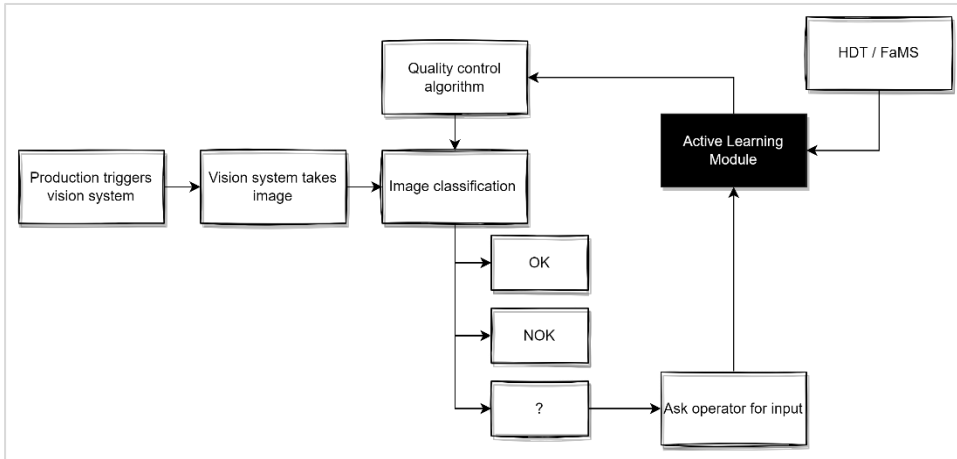


Figure 33: High Level Process View and Information Flows for the Human Robot Collaboration Use Case

### 3.1.2 Implementation View

In line with the presented process view, Figure 34 illustrates an implementation view of the quality control system. The vision system that processes the images is deployed within a production cell, while other modules (e.g., quality control algorithms) are hosted in proper containers.

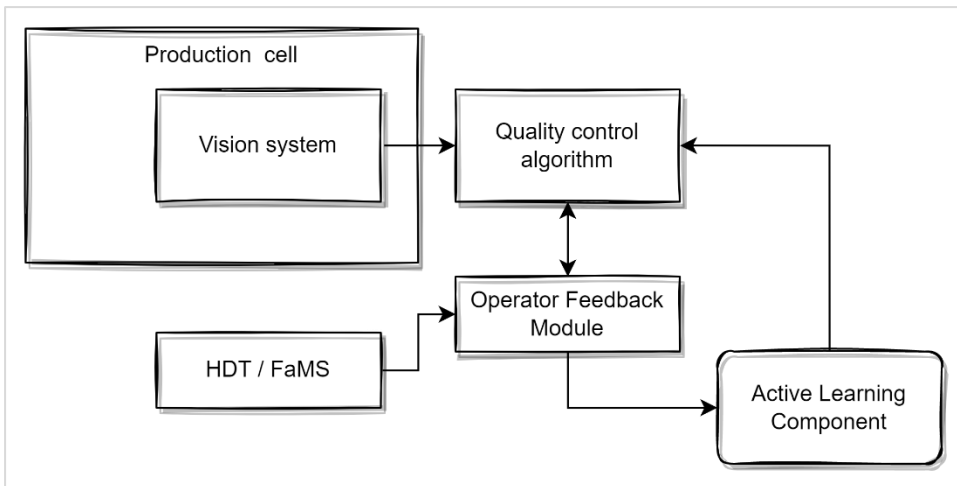


Figure 34: High-Level Implementation View of the Quality Control/Management Use Case

The following figures illustrate the current implementation of some of the components of the use case. Specifically:

- Figure 35 illustrates a snapshot of the production cell implementation.
- Figure 36 illustrates the main modules that comprise the visual quality inspection system.
- Figure 37 illustrates the implementation of the quality control algorithm in a pilot environment.



Figure 35: Production Cell

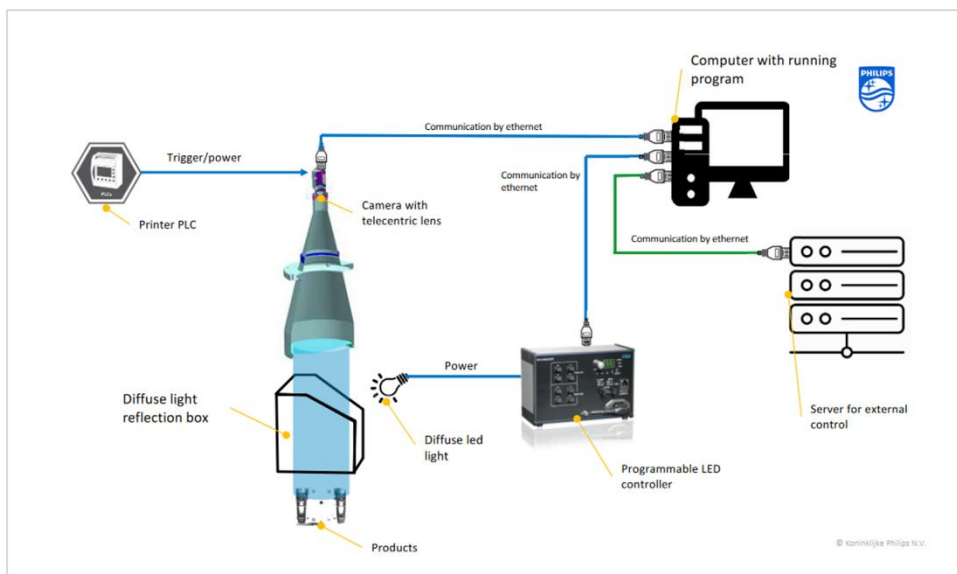


Figure 36: Vision System components

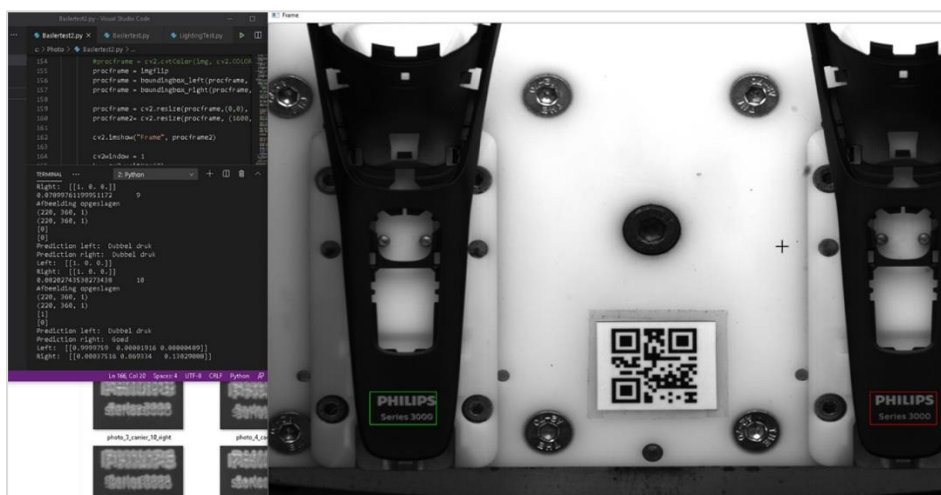


Figure 37: Quality Control Algorithm Implementation based on Python

## 3.2 Use Case #2 Human Behaviour Prediction and Safety Zones Detection

In the DFKI's use case, we utilize the Smartfactory demonstration which presents the latest technologies from the industry domain building industry-standard demonstrators. One new additional feature for our demonstrators would be considering safety during the collaboration between workers and autonomous robots. Although in the as-is scenario (Figure 31), the autonomous mobile robot assures progress to fulfil the safety aspect, this system is not optimal. In the as-is scenario, the robot utilizes its built-in laser scanner to create a map of the current environment as well as the static objects. After the map is created, the stations are manually defined, and their coordinates are stored by the robot. Later, the robot is programmed to deliver the product between the defined coordinates related to the modules. If an object is detected in the direction of moving, it waits for a specific amount of time and then continues to its target when the object moves out of the path. However, if the object does not move within this time span, the robot must be manually resumed. This progress could be slow while there is a lack of foresight to detect the possible obstacles/workers in the navigation path. Moreover, any change in the layout for which the robot created the map needs the generation of a new map of the new layout. Moreover, the as-is scenario does not consider human presence. The STAR project for this use-case plans to improve the situation and the intelligence of the robot. To tackle these difficulties and reach the aforementioned goals and optimize the progress with providing safety, we initiated three use cases. Three use cases derived from the STAR project are:

1. Human intention recognition.
2. Robot reconfiguration based on the dynamic layout.
3. Dynamic path planning using both first and second use cases.

The first use case plans to detect human activities and predict their next actions, which then will be combined with robot navigation to create a safer environment. For this matter, DFKI created a typical worker workflow, happening during normal daily work. The behaviour of more than 10 participants was recorded, who were supposed to follow the same or similar flow. The recordings were made using two wrist sensors which are then utilized as the data for a neural network model to be trained and then detect the activity they are currently performing.

The second use case is to dynamically update the navigation route of the mobile robot, by considering human and/or other (non-)moving objects in the environment. This use case will also enable easier reconfiguration of the robot in case the layout of the environment (including the production stations) changes. The layout is actively monitored by two stationary ceiling-mounted RGB cameras and humans, as well as the objects in the layout, are detected. In case of any change, the new coordinates of the stations, where the robot should navigate, are automatically updated without any barrier to updating the map of the robot.

Finally, as the third use case, these two use cases shall be combined to have a safe environment for the workers and the hardware equipment. The newly received coordinates of the stations will be used to set the robot's destinations. The speed of the robot and the objects in the layout will also be considered to create a collision-free navigation path for the robot. The human's current and next possible activities are also important aspects to consider during the decision. The points of interest for the robot will be sent as separate data to achieve the original goals of the use case.

### 3.2.1 Process View

The as-is and to-be scenarios are depicted in Figure 38 and Figure 39, respectively.

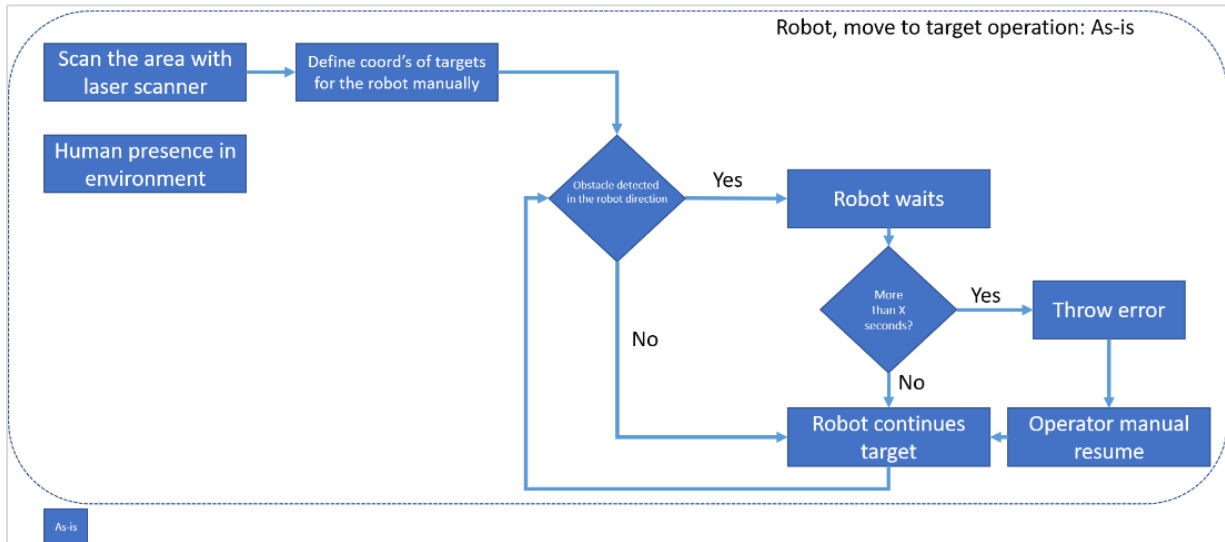


Figure 38: Process view for As-Is scenario

In Figure 39 Light blue blocks belong to the as-is scenario, the green boxes are new additions to the first use case, dark blue boxes are new features introduced by the second use case, and finally, the orange boxes are the functionalities that combine both first and the second use case.

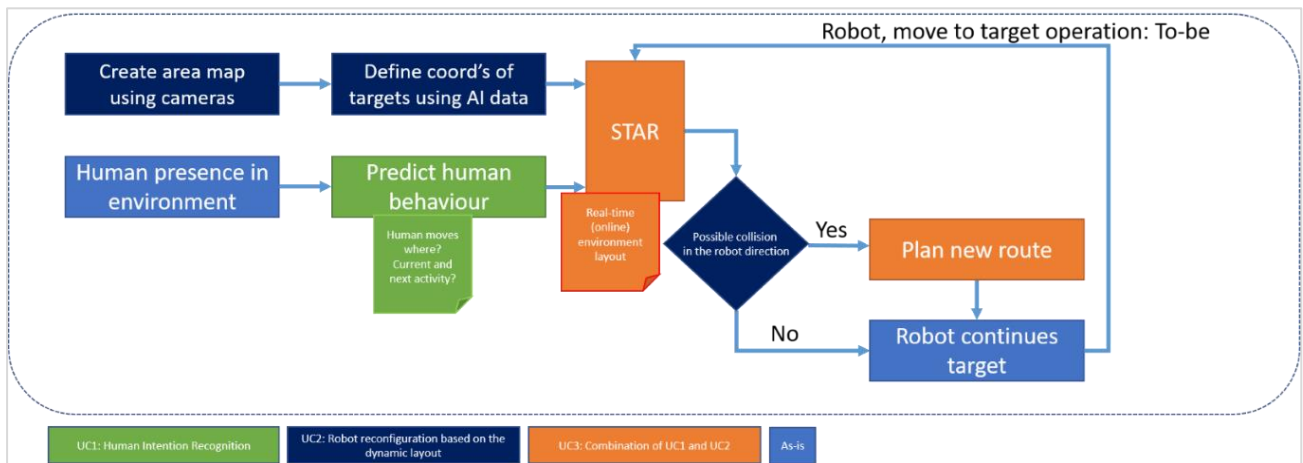


Figure 39: Process view for To-Be scenario

### 3.2.2 Physical View

In the two figures, Figure 40 and Figure 41, the physical view for as-is and to-be scenarios for the DFKI use-cases are presented, respectively. In the as-is scenario, we fulfil the safety aspect using the built-in robot's sensors. In the to-be scenario considering the two stationary cameras, we can provide different zones around the robot and consider the safety aspect for the mobile robot in the environment, which brings foresight in obstacles/workers detection.

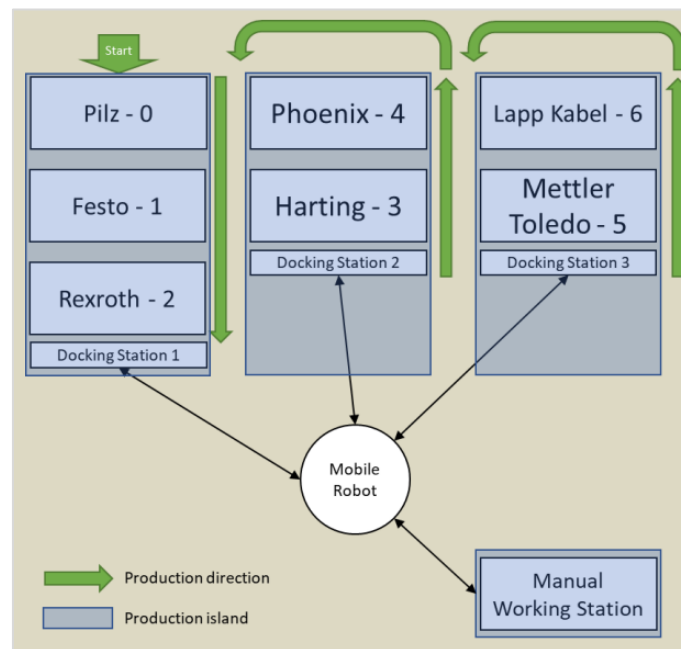


Figure 40: Physical view for As-Is scenario

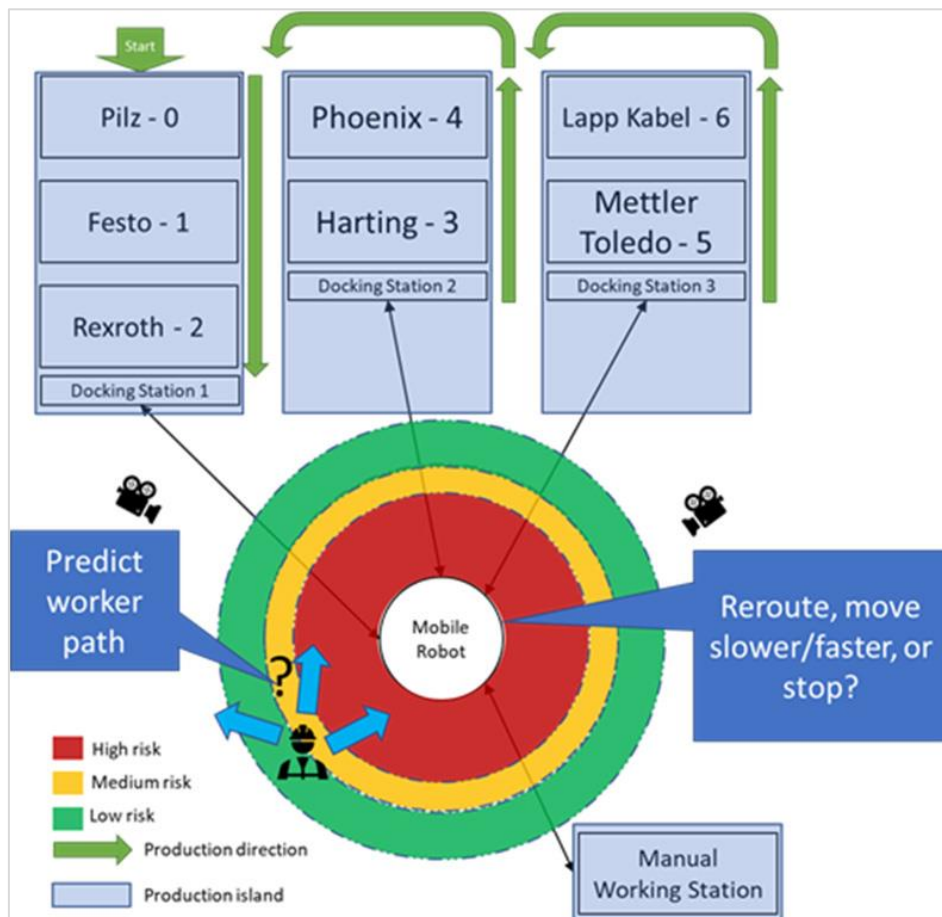


Figure 41: Physical view for To-Be scenario

### 3.2.3 Implementation View

The optimisation of the AMR fleet in co-habitation with workers relies on two modules as described in section 2.4.5.1:

1. Safety Zone Detection.
2. AMR Fleet Optimizer,

The first is based on the implementation of perception algorithms using static cameras deployed in the factory (DFKI lab in WP6 Pilot), the second will allow to control of robot path using a reinforcement learning approach that takes into account the human and object localisation of the safety zone detection module. Figure 42 shows the software components enabling the world description on which the fleet planner is based. Note that these components leverage several modules of the STAR logical architecture, including Reinforcement Learning modules, as well as various probes and ML (Machine Learning) models.

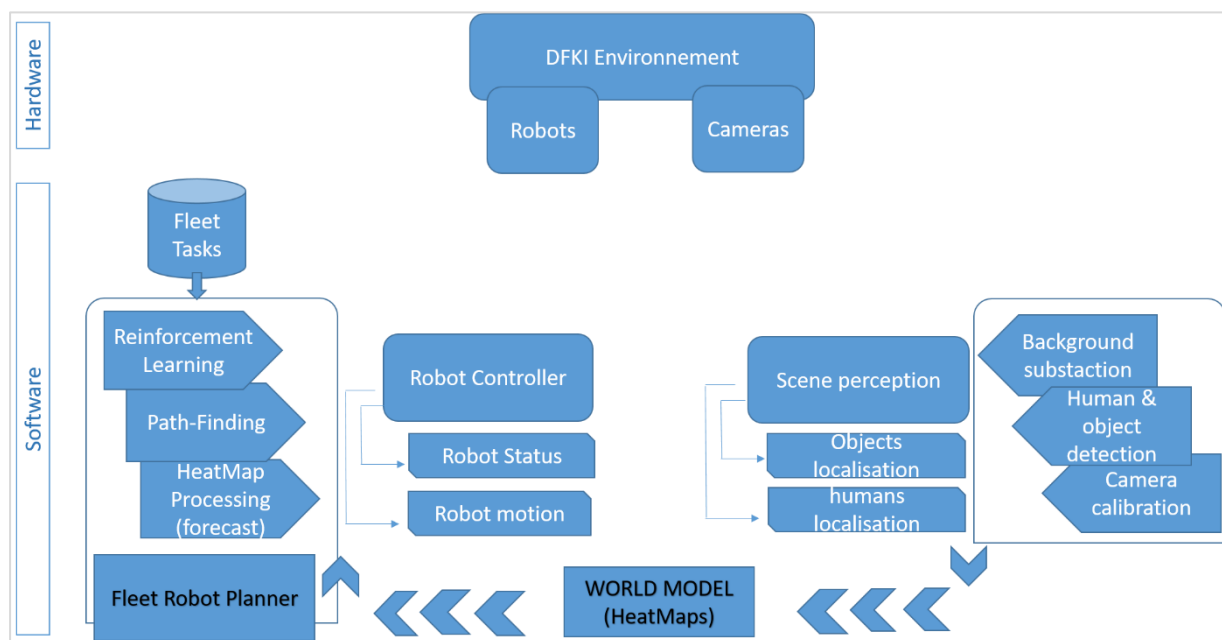


Figure 42: Software Components of the AMR Fleet Optimizer

## 3.3 Use Case #3 Human Centred AI for Agile Manufacturing

### 3.3.1 Process View

As illustrated in other deliverables (e.g., D6.1), STAR is implementing and deploying four AI-based use cases in IBER’s factory, namely:

1. UC#1: Production processes simulations for accelerated decisions and safe processes.
2. UC#2: Production planning optimization.
3. UC#3: Training of Employees for reduction of human errors.
4. UC#4: Agile production management system data integrity and reliability.

The four (sub) use cases are illustrated in the following figure (Figure 43):

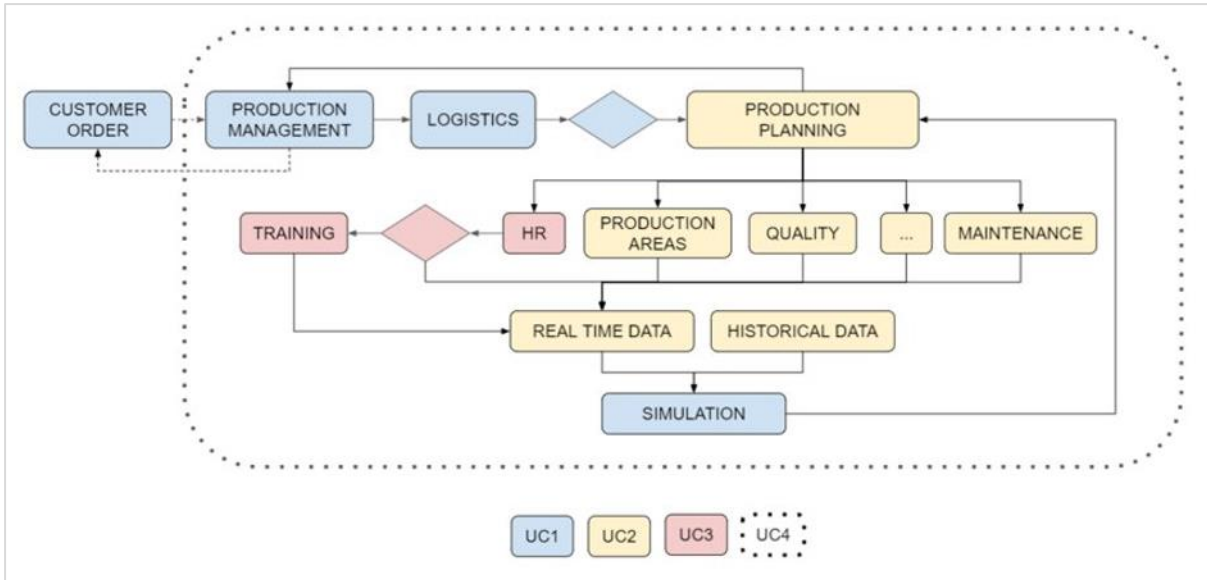


Figure 43: Overview of IBER UC Processes (see also deliverable D6.1)

### 3.3.2 Physical and Implementation Views

Figure 44 illustrates a preliminary implementation view of some of the IBER use cases in-line with the STAR logical architecture. Appropriate probes are implemented and deployed in proper containers (i.e., dockirized), as a means of ensuring historical (e.g., ERP based) and real-time (e.g., sensor/IoT based) data acquisition. Relevant data are persisted in a dockerized data management infrastructure (i.e., database, data lake). The implementation of the AI use cases (notably of UC1 and UC3) leverage the STAR open analytics platform and the XAI models, which drive the implementation of appropriate AI/ML models.

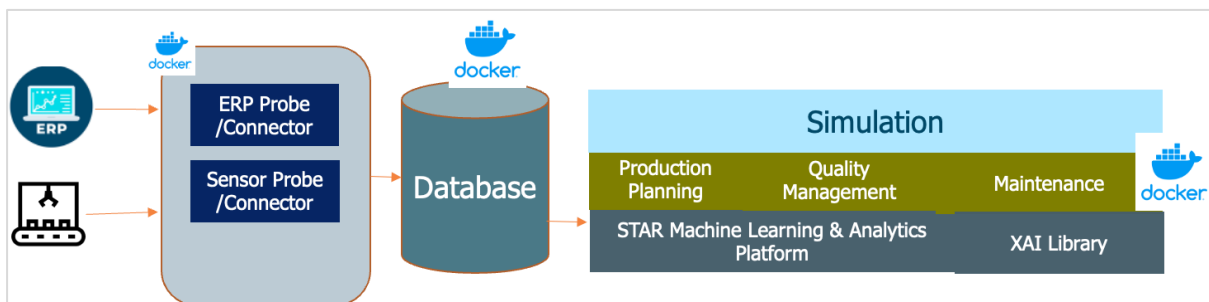


Figure 44: Preliminary Implementation view of the Agile Manufacturing Use Case(s) at IBER

## 4 Blueprints Specification

### 4.1 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. The blueprints are aimed at serving as proven ways to implement trusted data processing and AI functionalities for industrial applications. Rather than having to read, browse and understand the entire STAR reference architecture and its lower level technical details, interested parties (e.g., solution integrators, manufacturers, researchers in industrial automation and digital manufacturing) could consult the STAR blueprints as practical ways for enhancing the trustworthiness and regulatory compliance of their work.

### 4.2 Technical Integration Blueprints across the STAR Functional Domains

#### 4.2.1 Overview

The following paragraphs illustrate blueprints that indicate how to implement trusted AI solutions with an emphasis on the technical implementation of cybersecurity, safety and human robot collaboration use cases.

#### 4.2.2 Defending a Poisoning Attack

The following table illustrates the defending poisoning attack blueprint.

| Code                            | STAR-BLPR-1   |
|---------------------------------|---|
| <b>Title</b>                    | Poisoning Attack Defence  |
| <b>Scope/Purpose</b>            | 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. |
| <b>STAR Components Involved</b> | STAR Analytics Platform, DLSDR, STAR Blockchain, Risk Assessment and Mitigation Engine, XAI Module  |
| <b>UML Diagram</b>              | See Figure 7  |

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

#### 4.2.3 Defending an Evasion Attack

The following table illustrates the defending evasion attack blueprint.

| Code         | STAR-BLPR-2            |
|--------------|------------------------|
| <b>Title</b> | Evasion Attack Defence |

|                                 |   |
|---------------------------------|---|
| <b>Scope/Purpose</b>            | 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 |
| <b>STAR Components Involved</b> | STAR Analytics Platform, DLSDR, STAR Blockchain, Risk Assessment and Mitigation Engine, XAI Module  |
| <b>UML Diagram</b>              | See Figure 8  |

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

#### 4.2.4 Management and Configuration of Data Sources

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

|                                 |   |
|---------------------------------|---|
| <b>Code</b>                     | <b>STAR-BLPR-3</b>  |
| <b>Title</b>                    | Management and Configuration of Data Sources  |
| <b>Scope/Purpose</b>            | Configure a source of data in the manufacturing shopfloor to take advantage of the STAR system and its trusted AI capabilities. |
| <b>STAR Components Involved</b> | Probes Registry, Data Models, DLSDR, STAR Blockchain (Distributed Ledger)   |
| <b>UML Diagram</b>              | See Figure 9  |

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

#### 4.2.5 Security Policy Management

The following table illustrates the security policy management blueprint.

|                                 |   |
|---------------------------------|---|
| <b>Code</b>                     | <b>STAR-BLPR-4</b>  |
| <b>Title</b>                    | Security Policy Management  |
| <b>Scope/Purpose</b>            | Specify, Manage and Enforce policies regarding the operation of STAR AI systems |
| <b>STAR Components Involved</b> | Security Policy Management, Risk Assessment and Mitigation Engine               |
| <b>UML Diagram</b>              | See Figure 13   |

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

#### 4.2.6 Human Centred Digital Twin

The following table illustrates the HDT blueprint.

|                                 |  |
|---------------------------------|--|
| <b>Code</b>                     | <b>STAR-BLPR-5</b>   |
| <b>Title</b>                    | Human Centered Digital Twin  |
| <b>Scope/Purpose</b>            | Enable the Collection and Management of Humans’ Contextual Information for use in Digital Twins Applications |
| <b>STAR Components Involved</b> | Orchestrator, STAR Data Governance (IIoT Middleware) / DLSDR, Gateway/Orchestrator                           |
| <b>UML Diagram</b>              | See Figure 14  |

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

### 4.2.7 Explainable Artificial Intelligence

The following table illustrates the XAI blueprint.

| Code                            | STAR-BLPR-6   |
|---------------------------------|---|
| <b>Title</b>                    | Explainable Artificial Intelligence in Manufacturing  |
| <b>Scope/Purpose</b>            | Implement Counterfactual logic and Features Ranking for AI/ML algorithms in the scope of AI-based Manufacturing Use Cases |
| <b>STAR Components Involved</b> | XAI Models (collection of models & algorithms), AI Algorithms (used in applications)                                      |
| <b>UML Diagram</b>              | See Figure 15 & Figure 16   |

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

### 4.2.8 Active Learning for Human Robot Collaboration

The following table illustrates the human robot collaboration blueprint.

| Code                            | STAR-BLPR-7   |
|---------------------------------|---|
| <b>Title</b>                    | Active Learning for Human Robot Collaboration   |
| <b>Scope/Purpose</b>            | Boost Human Robot Collaboration and Accelerate Robot’s Acquisition based on Active learning |
| <b>STAR Components Involved</b> | STAR AI Algorithms, Active Learning Module  |
| <b>UML Diagram</b>              | See Figure 17   |

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

### 4.2.9 Feedback Provision

The following table illustrates the feedback provision blueprint, including feedback based on the NLP module.

| Code                            | STAR-BLPR-8  |
|---------------------------------|--|
| <b>Title</b>                    | Provision of Feedback during Human Robot Collaboration   |
| <b>Scope/Purpose</b>            | Enable humans to interact with Robots and Cyber physical systems towards providing feedback about their operations |
| <b>STAR Components Involved</b> | Feedback Module, NLP Module  |
| <b>UML Diagram</b>              | See Figure 18, Figure 19, Figure 17  |

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

### 4.2.10 Trusted Reconfiguration for Mobile Robot

The following table illustrates the trusted mobile robot reconfiguration blueprint.

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

|                    |               |
|--------------------|---------------|
| <b>UML Diagram</b> | See Figure 17 |
|--------------------|---------------|

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

#### 4.2.11 Management and Configuration of Analytics Processors

The following table illustrates the management and configuration of the analytics processors blueprint.

|                                 |   |
|---------------------------------|---|
| <b>Code</b>                     | <b>STAR-BLPR-10</b>   |
| <b>Title</b>                    | Management and Configuration of Analytics Processors  |
| <b>Scope/Purpose</b>            | Configure an analytics processor configuration in the manufacturing shopfloor to take advantage of the STAR system and its trusted AI capabilities. |
| <b>STAR Components Involved</b> | MPPE Registry, Data Models, DLSDR, STAR Blockchain (Distributed Ledger)   |
| <b>UML Diagram</b>              | See Figure 10 above   |

*Table 13: STAR-BLPR-10 – Management and Configuration of Industrial Data Sources*

#### 4.2.12 Validating the Integrity of Industrial Data

The following table illustrates the validation of the integrity of the industrial data blueprint.

|                                 |   |
|---------------------------------|---|
| <b>Code</b>                     | <b>STAR-BLPR-11</b>   |
| <b>Title</b>                    | Validating the Integrity of Industrial Data   |
| <b>Scope/Purpose</b>            | Retrieve persisted critical measurements (e.g., Analytics results) from the blockchain by any organization/service of the STAR platform to be validated/ compared with existing data to verify their authenticity in the manufacturing shopfloor. |
| <b>STAR Components Involved</b> | Data Models, DLSDR, STAR Blockchain (Distributed Ledger)  |
| <b>UML Diagram</b>              | See in Figure 11 above  |

*Table 14: STAR-BLPR-11 – Management and Configuration of Industrial Data Sources*

#### 4.2.13 Security Policy Definition and Enforcement

The following table illustrates the security policy definition and enforcement blueprint.

|                                 |   |
|---------------------------------|---|
| <b>Code</b>                     | <b>STAR-BLPR-12</b>   |
| <b>Title</b>                    | Security Policy Manager   |
| <b>Scope/Purpose</b>            | Receive input from RMS, BLCK and XAI modules and compare them with the set security policies related rules, in case of poisoning or evasion attacks threats, the SSPM sends an alert to Olistic |
| <b>STAR Components Involved</b> | Risk Assessment and Mitigation Engine Olistic, RMS, BLCK, XAI   |
| <b>UML Diagram</b>              | See SSPM physical view  |

*Table 15: STAR-BLPR-12 – Security Policy Definition and Enforcement Blueprint*

#### 4.2.14 Reliable Data Generation for Simulated Reality

The following table illustrates the reliable data generation using a simulated reality blueprint.

|                                 |  |
|---------------------------------|--|
| <b>Code</b>                     | <b>STAR-BLPR-13</b>  |
| <b>Title</b>                    | Synthetic Data Generation  |
| <b>Scope/Purpose</b>            | Generate reliable high-fidelity data to balance an input dataset adhering to a provided specification (e.g. number of images per class, generation method) |
| <b>STAR Components Involved</b> | STAR AI Algorithms, Simulated Reality  |
| <b>UML Diagram</b>              | See Simulated Reality Physical View  |

*Table 16: STAR-BLPR-13 – Data Generation Using Simulated Reality*

#### 4.2.15 Configuration of Risk Assessment and Mitigation Engine

The following table illustrates the RAME engine configuration blueprint.

|                                 |  |
|---------------------------------|--|
| <b>Code</b>                     | <b>STAR-BLPR-14</b>  |
| <b>Title</b>                    | Risk Assessment and Mitigation Engine  |
| <b>Scope/Purpose</b>            | Configuring the risk assessment and mitigation engine by adding the necessary assets of the production lines, configuring the threat landscape to be considered in the pilots. Operate in synergy with the security policy manager in order to create attack scenarios to be evaluated based on security incidents received. |
| <b>STAR Components Involved</b> | Security policy manager  |
| <b>UML Diagram</b>              | See Figure 13 above  |

*Table 17: STAR-BLPR-14 – Configuration of Risk Assessment and Mitigation Engine*

#### 4.2.16 Data Probes Configuration

The following table illustrates the data probes configuration blueprint.

|                                 |  |
|---------------------------------|--|
| <b>Code</b>                     | <b>STAR-BLPR-15</b>                                  |
| <b>Title</b>                    | Data Probes and Data Sources Configuration           |
| <b>Scope/Purpose</b>            | Configuring monitoring probes                        |
| <b>STAR Components Involved</b> | Probe Management, Monitoring Probes, Probes Registry |
| <b>UML Diagram</b>              | See Figure 12 above                                  |

*Table 18: STAR-BLPR-15 – Management and Configuration of Industrial Data Sources*

#### 4.2.17 Real-Time Data Monitoring

The following table illustrates the real time data monitoring blueprint.

|                      |   |
|----------------------|---|
| <b>Code</b>          | <b>STAR-BLPR-16</b>   |
| <b>Title</b>         | Real-Time Data Monitoring                                     |
| <b>Scope/Purpose</b> | Initiating and storing measurements from the monitored system |

|                                 |   |
|---------------------------------|---|
| <b>STAR Components Involved</b> | Probe Management, Monitoring Probes, Data Routing, Observation Repository, Data Bus |
| <b>UML Diagram</b>              | See Figure 12 above   |

Table 19: STAR-BLPR-16 – Management and Configuration of Industrial Data Sources

## 4.3 Regulatory Compliance Blueprints – Compliance to AI Act

### 4.3.1 Overview

In April 2021, the European Parliament and the Council of Europe presented an initial proposal for the regulation of AI systems [EP21]. This proposal is the first organized and structured effort to regulate AI systems worldwide. Its importance for systems deployed within Europe is particularly high, given that it lays a basis for future laws within the various EU member states. The proposal establishes a technology-neutral definition of AI systems in EU law, while presenting a risk-based classification of AI systems. The classification proposes to categorize AI systems in four general classes, ranging from unacceptable risk to no risk (i.e., risk free) systems. It also outlines the requirements and obligations associated with the deployment of systems from each one of the envisaged risk levels [Kop21]. For instance, “high-risk” AI systems can be authorised if and only if they meet a number of requirements spanning the areas of transparency, explainability, data quality and more. These obligations are significantly lower for medium and low risk systems.

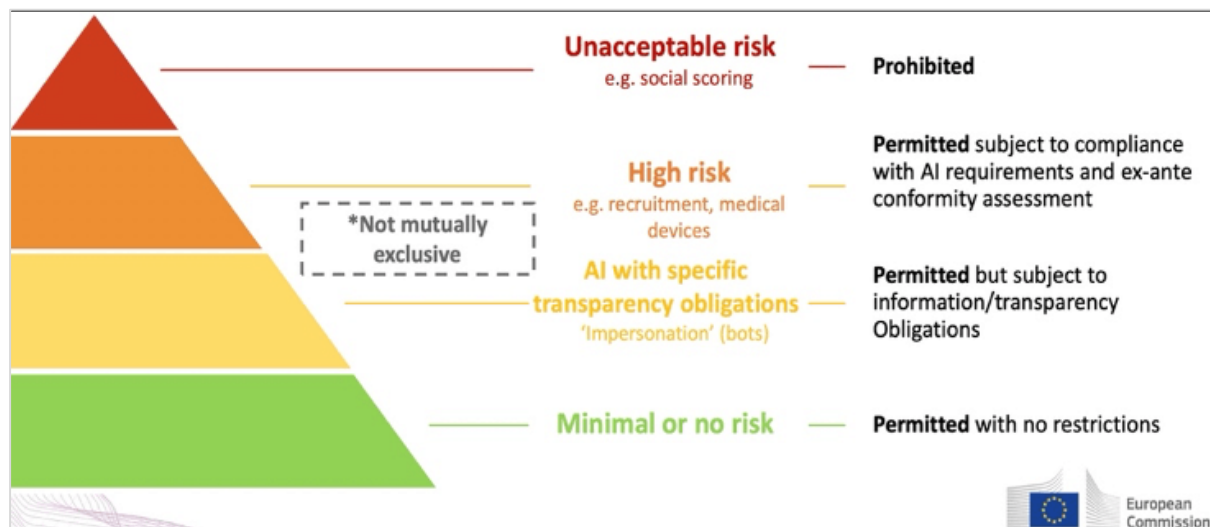


Figure 45: Overview of the Risk Levels of the AI Act

STAR is developing and offering technical components that can help AI deployers and operators to meet regulatory requirements and obligations. Different components can be used to support systems belonging to the different risk classes of the AI Act. The following paragraphs outline how the various STAR components can help in the development of regulatory compliant AI systems considering the mandates of the AI Act.

### 4.3.2 Minimal-Risk and No-Risk Systems

The AI Act specifies that minimal-risk systems (e.g., ML-based calculations and visualization of information about physical assets) can be deployed without essential restrictions.

Specifically, there are no mandatory obligations and compliance with the AI code of conduct, yet recommended is considered voluntary (see Figure 46).

Deployers may therefore choose to deploy one or more STAR components from the different domains of the platform (cybersecurity, human-robot-collaboration, safety), as well as explainable AI components (Table 20).

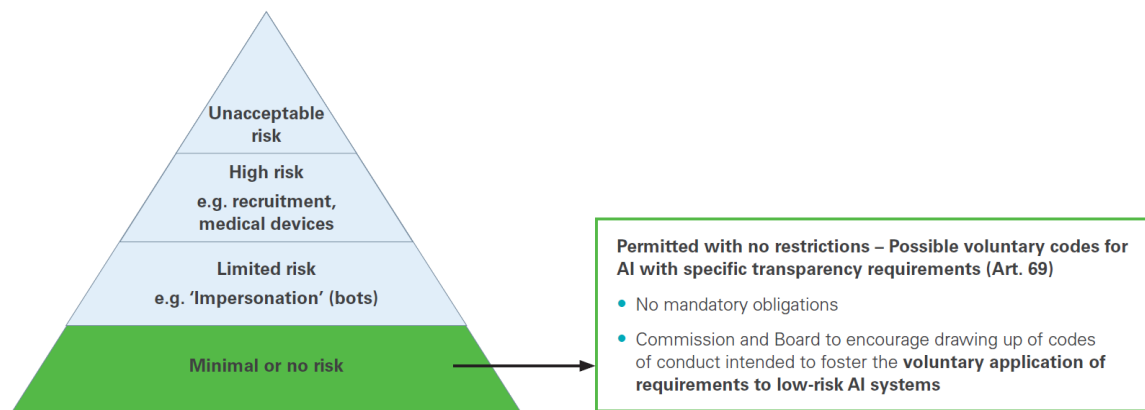


Figure 46: Requirements of Minimal or No Risk Systems

|                                 |  |
|---------------------------------|--|
| <b>Code</b>                     | <b>STAR-REG-BLPR-1</b>   |
| <b>Title</b>                    | Supporting the deployment of AI Systems of Minimal Risk  |
| <b>Scope/Purpose</b>            | Support adherence to Codes of Conduct that mandate transparency of AI system   |
| <b>STAR Components Involved</b> | <b>Optional Components:</b> XAI (for Transparency); DLSDR (for Data Quality); Security Policies Manager (for Increased Cybersecurity); AI Cyber-Defence Strategies (ACDS) (for AI cybersecurity) |
| <b>UML Diagram</b>              | N/A  |

Table 20: STAR-REG-BLPR-1 – Supporting the deployment of AI Systems of Minimal Risk

### 4.3.3 Limited-Risk Systems

When deploying a limited risk system, AI deployers must ensure that they are meeting transparency obligations (see Figure 47). In this direction, humans must be notified of the existence of an AI system component in the loop of the industrial process. This concerns industrial processes with the human in the loop, where AI systems and human interact. It is for example the case of some human centred digital twin applications, where industrial systems collect information about the status of the worker and adapt their operations to it.

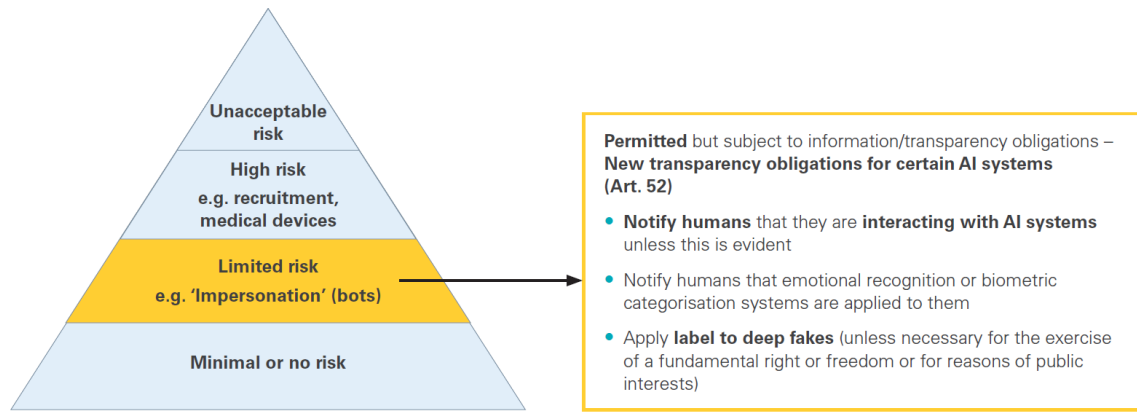


Figure 47: Requirements for Limited Risks Systems

STAR provides XAI components that can help deployed meet the requirements of limited risk deployments. Deployers can optionally use other STAR components to increase the safety, security and overall trustworthiness of the AI system.

|                                 |   |
|---------------------------------|---|
| <b>Code</b>                     | <b>STAR-REG-BLPR-2</b>  |
| <b>Title</b>                    | Supporting the deployment of AI Systems of Limited Risk   |
| <b>Scope/Purpose</b>            | Support the mandatory transparency of AI system; Provide optional support for increasing the security and safety of limited risk AI systems   |
| <b>STAR Components Involved</b> | <b>Mandatory Component:</b> XAI (for Transparency);<br><b>Optional Components:</b> DLSDR (for Data Quality); Security Policies Manager (for Increased Cybersecurity); AI Cyber-Defence Strategies (ACDS) (for AI cybersecurity) |
| <b>UML Diagram</b>              | N/A   |

Table 21: STAR-REG-BLPR-2 – Supporting the deployment of AI Systems of Limited Risk

### 4.3.4 High-Risk Systems

Many AI systems in manufacturing and other industrial environments can be classified as being of high-risk. This is for example the case with STAR’s AMR systems, as well as with other systems involving industrial robots. In the case of High-Risk systems, deployers and operators must comply with a longer list of requirements, including more stringent requirements regarding explainability, transparency, data quality and more (see Figure 48).

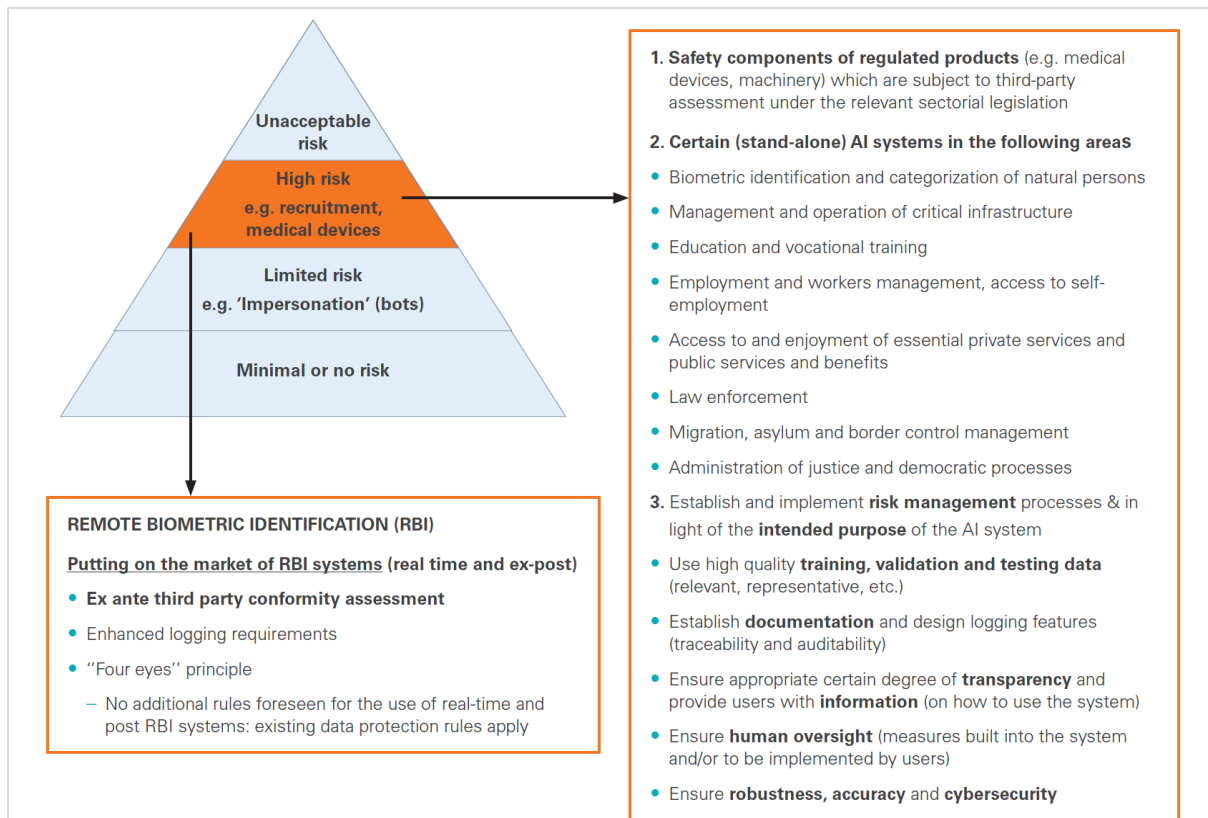


Figure 48: Requirements for High-Risk Systems

To support the qualification, deployment and use of such a high-risk system, STAR provides a host of relevant components that support data reliability, AI algorithms’ reliability, increased cybersecurity, safe human-robot collaboration and more. The use of these systems in a high-risk AI context becomes mandatory rather than optional. This is illustrated in the blueprint of

|                                 |  |
|---------------------------------|--|
| <b>Code</b>                     | <b>STAR-REG-BLPR-3</b>   |
| <b>Title</b>                    | Supporting the deployment of AI Systems of High Risk   |
| <b>Scope/Purpose</b>            | Support the mandatory transparency of AI system; Provide optional support for increasing the security and safety of limited risk AI systems  |
| <b>STAR Components Involved</b> | <b>Mandatory Components:</b> XAI (for Transparency); DLSDR (for Data Quality); Security Policies Manager (for Increased Cybersecurity); AI Cyber-Defence Strategies (ACDS) (for AI cybersecurity); AMR Safety (for Increased Safety) |
| <b>UML Diagram</b>              | N/A  |

Table 22: STAR-REG-BLPR-3 – Supporting the deployment of AI Systems of Limited Risk

## 5 Conclusions

This deliverable has presented the final version of the STAR reference architecture for trusted AI systems, along with a wide range of blueprints for different use cases. The deliverable has extended the work that was carried out as part of the earlier version of the deliverable (i.e., D2.6). The latter work has served as a starting point for the final version of the STAR architecture. At a high level, the STAR architecture comprises functionalities clustered in three complementary domains, including cyber-security, human robot collaboration and safety domains. The functionalities in each one of the domains reinforce functionalities in the other domains. Moreover, the XAI components of the project are used to support functionalities in all three domains.

The architecture has been presented in-line with the popular 4+1 methodology i.e., it has been presented in the form of five views including a logical, process, implementation, physical/deployment and scenarios/use cases views. The views cover different functional areas of the architecture for different use cases, given that STAR platform is not based on an “all or nothing” value proposition. Rather STAR enables manufacturers and integrators of AI solutions in production lines to select subsets of components of the reference architecture in order to meet different sets of industrial requirements.

Even though the deliverable provides views of the architecture that go down to implementation detail, its primary aim is to provide higher level structuring principles and blueprints for the implementation of AI-based industrial systems. In-depth information about the STAR components and their implementation details are provided in other technical deliverables of the project, notably deliverables of WP3, WP4 and WP5 of the project. These technical deliverables complement the technical information of the present deliverable.

The presented blueprints focus on guidelines about the technical implementation of trustworthy AI systems in production lines. However, they also provide guidance on how to use the STAR components in order to implement systems that adhere to the mandates of the AI regulation proposal of the European Parliament and the Council of Europe. This regulatory related guidance complements the ethical management work of the project in WP1 and is considered important for companies that seek to comply with the AI Act and to demonstrate regulatory readiness.

The earlier version of the deliverable (D2.6) has been already considered in the implementation and the integration of the STAR systems in the various technical workpackages of the project. Likewise, this version will be provided as updated input to these workpackages to drive their implementation work. However, our ambition is to disseminate the outcomes of this deliverable to the broader European community of AI in manufacturing, as certain components and blueprints can be of wider interest. We will be doing this dissemination using materials already produced (e.g., the STAR book, blogs on the STAR architecture), as well as additional materials that will be developed and integrated into the STAR market platform.

## References

| Reference        | Name of document   |
|------------------|--|
| [BDVA17]         | European Big Data Value Strategic Research and Innovation Agenda, Version 4.0, October 2017, <a href="https://bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf">https://bdva.eu/sites/default/files/BDVA_SRIA_v4_Ed1.1.pdf</a> 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: <a href="https://docs.docker.com/">https://docs.docker.com/</a> , last accessed: March 2022  |
| [EP21]           | Proposal for a REGULATION OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL LAYING DOWN HARMONISED RULES ON ARTIFICIAL INTELLIGENCE (ARTIFICIAL INTELLIGENCE ACT) AND AMENDING CERTAIN UNION LEGISLATIVE ACTS<br>COM/2021/206 final  |
| [IEEE42010]      | ISO/IEC/IEEE: "ISO/IEC/IEEE 42010:2011 Systems and software engineering -- Architecture description", 2011<br><a href="http://www.iso.org/iso/catalogue_detail.htm?csnumber=50508">http://www.iso.org/iso/catalogue_detail.htm?csnumber=50508</a>                              |
| [IIRAv1.9]       | Industrial Internet Consortium, "The Industrial Internet Reference Architecture v 1.9", available at: <a href="https://www.iiconsortium.org/IIRA.htm">https://www.iiconsortium.org/IIRA.htm</a> , last accessed September 2 <sup>nd</sup> 2021                                 |
| [IISF]           | Industrial Internet Consortium, "The Industrial Internet Security Framework", available at: <a href="https://www.iiconsortium.org/IISF.htm">https://www.iiconsortium.org/IISF.htm</a> , last accessed September 2 <sup>nd</sup> 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: <a href="https://jfrog.com/knowledge-base/what-is-a-container-registry/">https://jfrog.com/knowledge-base/what-is-a-container-registry/</a> . [Accessed August 2021]   |
| [Kop21]          | Mauritz Kop, EU Artificial Intelligence Act: The European Approach to AI, Transatlantic Antitrust and IPR Developments (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 Mladenec, 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, <a href="http://dx.doi.org/10.1561/9781680838770">http://dx.doi.org/10.1561/9781680838770</a> .   |
| [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: <a href="https://hackernoon.com/how-to-dockerize-any-application-b60ad00e76da">https://hackernoon.com/how-to-dockerize-any-application-b60ad00e76da</a> , 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.  |