Reference Architecture for AI REDGIO 5.0 - M9

D4.1

Person responsible / Author:	Nenad Stojanovic (NISSATECH), Jan van Deventer(LTU), Nefeli Bountouni (SUITE5), Gabriele De Luca (ENG)
Deliverable N.:	**D4.1**
Work Package N.:	**4**
Date:	30/09/2023
Project N.:	101092069
Classification:	Public
File name:	D4.1_Reference Architecture for AI REDGIO 5.0_M9 v.1.0
Number of pages:	55

This project has received funding from the European Union's Horizon Europe Research and Innovation Program under Grant Agreement N. 101092069

Status of deliverable

Action	By	Date (dd.mm.yyyy)
Submitted (author(s))	Nenad Stojanovic (NISSATECH)	20/11/2023
Responsible (WP Leader)	Nenad Stojanovic (NISSATECH)	20/11/2023
Approved by Peer reviewer	Sergio Gusmeroli (POLIMI)	20/11/2023

Revision History

Date (dd.mm.yyyy)	Revision version	Author	Comments
23/07/2023	v. TOC	Nenad Stojanovic (NISSATECH)	TOC
06/09/2023	v. 01	Nenad Stojanovic (NISSATECH)	First Version for contributions
20/09/2023	v. 05	Jan van Deventer (LTU)	Arrowhead contribution
27/09/2023	v. 06	Nefeli Bountouni (SUITE5)	Data space contribution
05/10/2023	v. 07	Nenad Stojanovic (NISSATECH)	RA draft
05/10/2023	v. 08	Nenad Stojanovic (NISSATECH)	RA draft
02/11/2023	v. 08	Nenad Stojanovic (NISSATECH)	Requirements draft
10/11/2023	v. 09	Nenad Stojanovic (NISSATECH)	Complete draft
20/11/2023	v. 1	Nenad Stojanovic (NISSATECH)	Refinement

Author(s) contact information

Name	Organisation	E-mail	Tel
Nenad Stojanovic	NISSATECH	Nenad.Stojanovic@nissatech.com	
Jan van Deventer	LTU	jan.van.deventer@ltu.se	
Nefeli Bountouni	SUITE5	nefeli@suite5.eu	

Table of Contents

TABLE OF CONTENTS **3**

FIGURES **4**

TABLES **4**

1. EXECUTIVE SUMMARY **5**

1.1. INTRODUCTION 6
1.2. SCOPE OF THE DELIVERABLE 6
1.3. CONTRIBUTIONS TO OTHER WPS AND DELIVERABLES 6
1.4. STRUCTURE OF THE DOCUMENT 6

2. REFERENCE ARCHITECTURE BACKGROUNDS **7**

2.1. BACKGROUNDS AND RELEVANT INITIATIVES 7
2.2. REFERENCE ARCHITECTURAL MODEL INDUSTRIE 4.0 (RAMI 4.0) 7
2.3. ASSET ADMINISTRATION SHELL 10
2.4. BDVA BIG DATA VALUE ASSOCIATION 12
2.5. INDUSTRIAL INTERNET REFERENCE ARCHITECTURE (IIRA) 15
2.6. OPEN DEI 18
2.7. ARROWHEAD FRAMEWORK 20
2.7.1. The local automation cloud approach 20
2.7.2. Architecture fundamentals 22
2.7.3. Security in local clouds 22
2.7.4. Automation support core systems 23
2.7.5. Data encryption 23
2.7.6. An example: SMEI5.0 SHOP FLOOR USE CASE 23
2.8. EDGE COMPUTING 24
2.8.1. IBM Edge computing reference architecture 25
2.8.2. IBM edge architecture principles 26
2.8.3. IBM Edge computing reference architecture 27
2.9. REAL-TIME LOCATION SYSTEMS (RTLS) IN MANUFACTURING 28
2.9.1. Introduction 28
2.9.2. RTLS structure 29
2.9.3. RTLS for Manufacturing 29
2.9.4. A possible scenario 30
2.10. MODERN DATA PLATFORMS 31
2.10.1. Data Storage and Processing 33
2.10.2. Data Ingestion 34
2.10.3. Data Transformation and Modeling 35
2.10.4. Business Intelligence (BI) and Analytics 36

3. AI REDGIO 5.0 REFERENCE ARCHITECTURE **37**

3.1. ARCHITECTURE REQUIREMENTS AND SPECIFICATIONS 37
3.1.1. Inputs from Experiments 37
3.2. DESIGN PRINCIPLES FOR AI REDGIO 5.0 RA 45
3.3. AI REDGIO 5.0 REFERENCE ARCHITECTURE 47
3.4. REFERENCE IMPLEMENTATION 49

4. CONCLUSIONS **55**

Figures

Figure 1 - The "automation pyramid" 8
Figure 2 - Digitalization trend towards RAMI 4.0 8
Figure 3 - RAMI 4.0 9
Figure 4 - AAS common layers and logical view 11
Figure 5 - AAS exemplary information structure 12
Figure 6 - Big Data Value Reference Model 13
Figure 7 - Functional view of IIRA 15
Figure 8 - Functional decomposition of Control and Monitoring domain 16
Figure 9 - Functional decomposition of System Management domain 16
Figure 10 - Functional decomposition of Information, Application and Business domains 17
Figure 11 - Interrelations between RAMI4.0 and IIRA 17
Figure 12 - OPEN DEI Reference Architecture Framework 19
Figure 13 – Local Cloud structure 21
Figure 14 – System of Systems 21
Figure 15 – System of Systems 24
Figure 16 – Edge computing layers 25
Figure 17 – Edge computing layers: data view 26
Figure 18 – Edge computing infrastructure 27
Figure 19 – Edge computing infrastructure 30
Figure 20 – The position of Data Observation 32
Figure 21 – The position of Data Observation 33
Figure 22 - Layered Databus Architecture 46
Figure 23 - AI REDGIO 5.0 Reference Architecture 47
Figure 24 - AI REDGIO 5.0 Reference Implementation 50

Tables

Table 1: User requirements for Industry 5.0 Data4AI Platform & Data Spaces 40
Table 2: User requirements for Industry 5.0 EDGE AI Toolkit & AI-On-demand Platform 45

1. Executive summary

The **goal of AI REDGIO 5.0** project is to **move artificial intelligence (AI) and machine learning (ML) to the "edge"** which means closer to the source of manufacturing information. This "local cloud" depicts the possibility to have an efficient computing infrastructure locally, which can be of a big importance for manufacturing small and medium-sized enterprises (SMEs). AI REDGIO 5.0 will lower the complexity and cost barriers for manufacturing enterprises (notably SMEs) to develop, deploy and fully leverage the cloud/edge computing paradigm for the implementation of applications like predictive maintenance, quality management, Zero Defect Manufacturing, Lifecycle Assessment, Intelligent Asset Management, Human Robot Collaboration and Digital Twins, through providing:

- **free open source**, and **user friendly tools** for developing end-to-end **pipeline for edge ML/AI applications**;
- **cloud/edge AI deployment patterns and blueprints** for different manufacturing applications.

During the project implementation **7 SMEs** are monitored and supported in their experiments deployment and **14 Didactic Factories** around Europe are involved for the implementation of didactic experiments addressed to SMEs.

The present deliverable is the result of the Task 4.1 "T4.1 Reference Architecture for AI REDGIO 5.0", which main goal is to collect requirements, define design principles and create reference architecture for Industry 5.0 Data4AI Platform.

This deliverable addresses the definition of the AI REDGIO 5.0 Reference Architecture aligned to the state-of-the-art Industry 4.0 Reference Architectures and Reference Models from previous I4MS and AI-related initiatives. Starting from the requirements collected in the D2.2 and D2.3, the design phase has considered several scenarios coming from real needs described in the trials.

Furthermore, a Reference Implementation, based on open-source components, is defined in order to lower the barriers for the adopters, acting as a baseline for the experiment deployments.

The document has been elaborated by the main partners involved in the WP4, in strict synergy with the WP4 task leaders.

2. Introduction

2.1. Scope of the deliverable

The present deliverable is the result of the Task 4.1 “Reference Architecture for AI REDGIO 5.0”, which main goal is to collect requirements, define design principles and create reference architecture for Industry 5.0 Data4AI Platform.

This deliverable is the first release of the Reference Architecture. Second release is planned for M27.

2.2. Contributions to other WPs and deliverables

Indicators defined in this deliverable are related to each WP.

Furthermore, requirements collected in the context of WP2 and detailed into D2.2 and D2.3, will represent the input for the refinement phase fitting the needs of all AI REDGIO 5.0 clusters composed of 21 experiments for implementing different use cases.

The project will define an open Reference Architecture for AI scenarios totally implementable with open-source components.

During the project implementation **7 SMEs** are monitored and supported in their experiments deployment and **14 Didactic Factories** around Europe are involved for the implementation of didactic experiments addressed to SMEs.

The Reference Architecture and the proposed Reference Implementation will be validated in the context of WP6 during the implementation of the use cases.

2.3. Structure of the Document

The rest of the deliverable is structured as follows:

- Section 2: backgrounds in terms of reference architectures, briefly described, to design AI REDGIO 5.0 RA in the following section. Standards and protocols useful to develop the platform properly.
- Section 3: AI REDGIO 5.0 Reference architecture designed and structured following the design principles Definition of Reference Implementation empowering the already know tools for Industry 5.0 with AI tools.
- Section 4: description of the achieved results and next steps can be reached from this checkpoint

3. Reference Architecture Backgrounds

3.1. Backgrounds and relevant initiatives

In this Section the main solutions and initiatives will be described, paying a particular attention on the relevant Reference Architectures in relation to data-driven manufacturing. These will represent the baseline for designing the AI REDGIO 5.0 Reference Architecture.

The AI REDGIO 5.0 project aims to define a Reference Architecture for describing a large variety of industrial scenarios, where Artificial Intelligence will have effects on the existing processes and edge processing on its realization.

Currently, the Industrial Internet Consortium (IIC) and the Working Group for Industry 4.0, have been working to provide recommendations and guidelines following the structure of the Industrial Internet Reference Architecture (IIRA) and the Reference Architecture Model for Industry 4.0 (RAMI 4.0), respectively.

Thus, from these starting points, the objective of this document is to give a data-driven approach for its reference architecture capable of driving the industrial sector transformation into a networked, data driven environment, empowered at the same time with the concepts of AI that will supports the entire scenario, adopting several patterns for gateways, edge architectures and databuses at different layers.

Open initiatives such as FIWARE and additional open-source components will be able to support the AI REDGIO 5.0 platform, defining high value applications for manufacturers to optimize production systems and value chains, and radically improve sustainability, productivity, innovation and customer service using the advances made in big data and data analytics through both, the data-in-motion and the data-at-rest.

Architectural IIoT systems where connected machines, or industrial systems, exchanging data, analysis, and actions to improve production and performance, and reduce defects or failures, are the main features of the framework. A huge amount of data is collected and analyzed, putting them at the disposal of the final user who is in charge of work with the tools allowing the factories to upgrade their performance.

3.2. Reference Architectural Model Industrie 4.0 (RAMI 4.0)

The idea behind “Industry4.0” is to enable significantly more interaction between shop-floor devices and high-level business systems, building on the previous generation of industrial monitoring and control systems. As a result, production flexibility is increased, and the International Society of Automation 95 (ISA 95)[1] traditional rigorous hierarchical approach has changed . This hierarchy is shown in the pyramid of Figure 1 and the structure expresses the typical model of production systems automation before the Industry4.0 paradigm, when each level of the pyramid relies on the layer below, with the lower levels providing the data and control necessary for the higher levels to monitor, make decisions, and take action. This hierarchical structure allows a clear separation of responsibilities and enables the integration of different systems and technologies. The pyramidal layers are built on a "layer 0", which includes the hardware (sensors, actuators) involved in the production process that are responsible for the direct control and the collection of data from the field devices, and then build up layer 1 the Programmable Logic Controller (PLC) layer which is responsible for the control of the equipment, the collection of data, and the monitoring of alarms and events, layer 2 (SCADA layer), layer 3 (Manufacturing Execution System, MES, layer) and layer 4 (Enterprise Resource Planning, ERP, layer). The IEC 62264 is an international standard established by the International Electrotechnical Commission, which serves as a complementary standard to ISA 95. It provides a comprehensive framework for the integration and communication of enterprise and control systems in the context of Industry 4.0.

[1] P. Arcaini, E. Riccobene and P. Scandurra, ""Modeling and Analyzing MAPE-K Feedback Loops for Self-Adaptation," 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems, Florence, Italy, 2015, pp. 13-23, doi: 10.1109/SEAMS.2015.10".

IEC 62264 provides a standardized approach to integration and communication between different systems within a manufacturing organization since it defines a set of functional requirements and protocols for communication between these systems, which is critical for effective industrial control and automation [10].

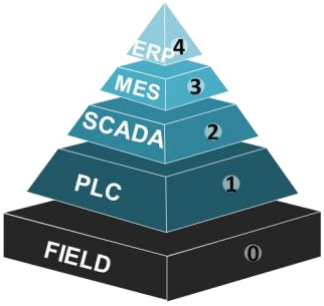

Figure 1 - The "automation pyramid"

ISA-95 automation system alone is not able to address the dynamics of a flexible production environment, and this can be seen by the fact that many businesses are already requesting new models and architectures for dynamic and digitized production as they struggle with high costs when making even simple adjustments to their ISA-95-based production automation systems[1]. In particular, there is a transition from legacy automation technology as defined by ISA-95 to highly distributed Internet of Things (IoT)- based automation systems that fully utilize Internet technologies, thus enabling the implementation of the so-called "Reference Architectural Model Industrie 4.0" (RAMI 4.0), a reference model that provides a common understanding and a common language for Industry 4.0, defining the key components, their relationships and interactions required for Industry 4.0 and outlining the technical requirements and standards needed to implement Industry 4.0 solutions.

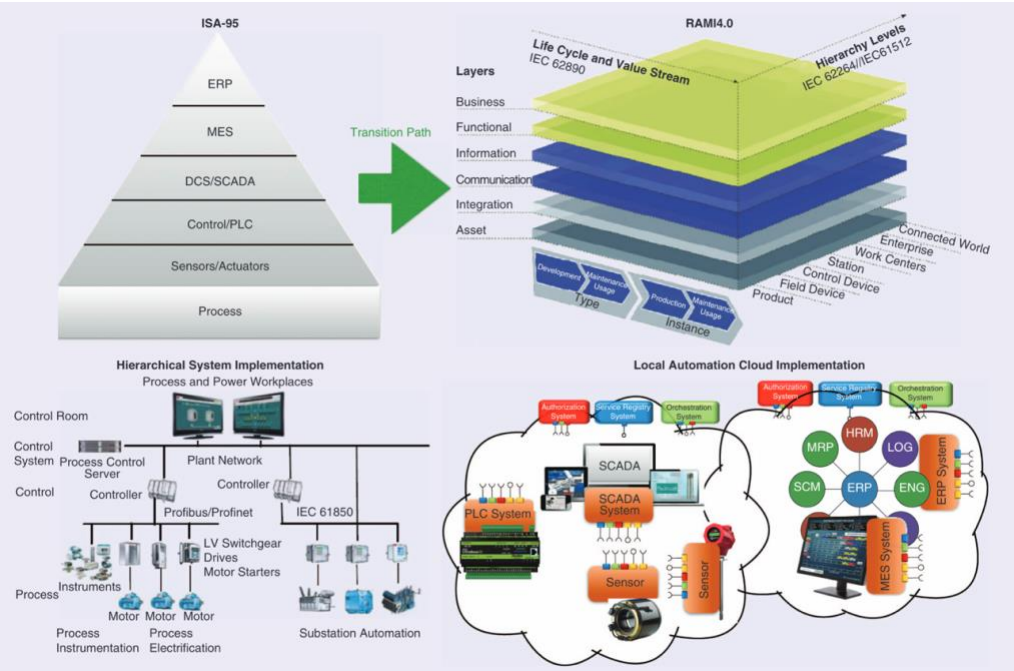

Figure 2 - Digitalization trend towards RAMI 4.0

[1] J. Delsing, "Local Cloud Internet of Things Automation: Technology and Business Model Features of Distributed Internet of Things Automation Solutions," IEEE Industrial Electronics Magazine, vol. 11, no. 4, pp. 8–21, Dec. 2017, doi: 10.1109/MIE.2017.275934.

This new paradigm generates value through repurposing and extracting useful information from raw manufacturing data enabling the possibility of data's flow between different elements of a more complicated system that may connect directly with one another and with their local environment on many levels. This new representation (Figure 2) emphasizes, from a network perspective, how close each layer is to the field, enabling data collection straight from it. In order to monitor and control the production system and eventually support a decision-making process through the computation of Key Performance Indicators, the layers 1 to 4 require these data (KPIs)[1].

RAMI 4.0 utilizes a three-dimensional coordinate system to encompass all crucial elements of Industry 4.0. The three axes allow for the mapping of all critical aspects of Industry 4.0, creating a 3D representation of Industry 4.0 solutions. It serves as a guide for aligning the needs of various fields with various standards, to implement Industry 4.0 methods, and enables the recognition of overlapping requirements, gaps and their resolution in accordance with related industrial standards[2].

Layers in the vertical axis (Figure 3) indicate the appearance from various angles (a look from the market aspect, a look from a perspective of functions, information, communication, a look from an integration ability of the components)[3].

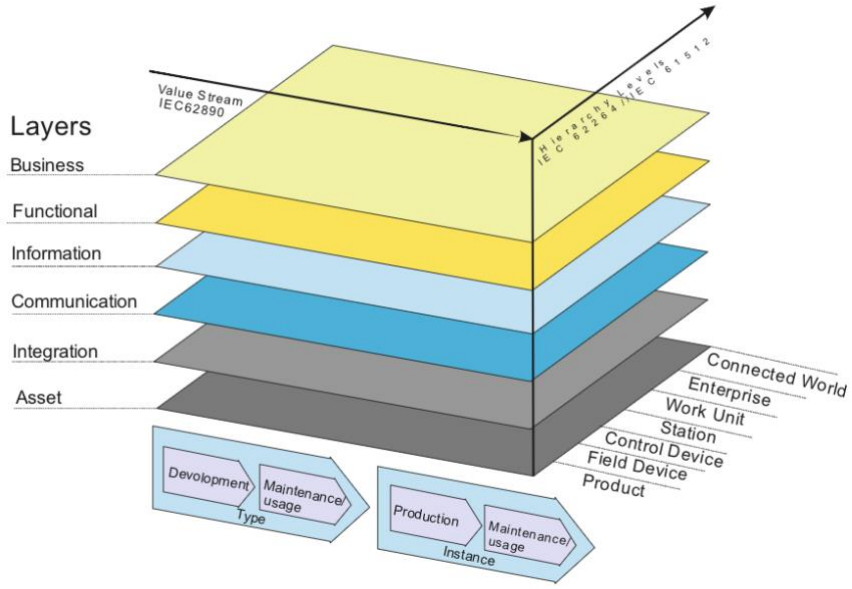

Figure 3 - RAMI 4.0

Moreover, the product life cycle and the value stream it comprises are crucial criteria in contemporary engineering and this characteristic can be seen along the left-hand horizontal axis. Examples given include continuous data collection throughout the life cycle. Even the complete digitalization of the entire

[1] W. Quadrini, E. Negri, and L. Fumagalli, "Open interfaces for connecting automated guided vehicles to a fleet management system," in Procedia Manufacturing, Elsevier B.V., 2020, pp. 406–413. doi: 10.1016/j.promfg.2020.02.055.

[2] S. M. e. a. Kannan, "Towards industry 4.0: Gap analysis between current automotive MES and industry standards using model-based requirement engineering," in *Proceedings - 2017 IEEE International Conference on Software Architecture Workshops, ICSAW 2017: Side Track Proceeding Institute of Electrical and Electronics Engineers Inc., Jun. 2017, pp. 29–35. doi: 10.1109/ICSAW.2017.53*.

[3] "C. , S. L. , & H. R. Manzei, Industrie 4.0 im internationalen Kontext. 2016".

development-market chain offers enormous possibilities for improving goods, machinery, and other Industry 4.0 architecture layers over their entire lifecycle[1].

The following model axis (right on the horizontal level) describes the functional positioning of the Industry 4.0 components. The functionality of the components is specified in this axis; there is no implementation specification, only the function assignment.

The following layers exist:

1. The asset layer, which is the bottom layer of RAMI 4.0, is in charge of fostering a shared knowledge of the physical assets present in an Industry 4.0 system. These resources could be anything, from infrastructure and buildings to tools and machinery. The asset layer is used to collect and store information about these assets' state, performance, and maintenance requirements in addition to providing a standard method of defining and identifying them.
2. Integration layer: In an Industry 4.0 environment, this layer is in charge of linking and coordinating the various systems and devices. It ensures that the various systems and devices can cooperate and allows data and information to flow between the various architectural layers. The integration layer also makes it possible to integrate older Industry 4.0 technologies with legacy systems and equipment.
3. Communication layer: This layer offers the infrastructure and protocols required for inter-system and inter-device communication. It facilitates the safe and dependable transfer of data and information between the various architectural layers and supports wired and wireless networks, among other communication technologies.
4. The information layer is in charge of managing and processing the data and information that the other levels acquire and produce. In addition to enabling the production of a digital twin of the physical assets and processes, it also comprises tools and technology for data storage, analysis, and visualization.
5. Functional layer: This layer offers the capabilities and functionality needed to regulate and improve the Industry 4.0 environment. It facilitates the development of intelligent, self-optimizing systems and encompasses technologies such as automation, robotics, and artificial intelligence.
6. Business layer: This layer is in charge of giving the Industry 4.0 environment its business context and objectives. It covers the operational procedures and procedures that the other levels make possible, such as supply chain management, customer relationship management, and production planning. The managerial and decision-making abilities necessary for the business to function are also included in the business layer.

3.3. Asset Administration Shell

The Asset Administration Shell (AAS) is defined as the standardized digital representation of an asset, which might be considered as any physical or logical object that has an actual or perceived value for an organization (IEC 62832-1:2020 [16]). Together, an asset and its corresponding AAS form the building blocks of the I4.0-component, which can represent any component within the RAMI 4.0 hierarchy levels, and it is the core element of Industry 4.0.

The AAS provides access to the entirety of information of its asset and ensures the interoperability between I4.0 components in modern Industrie 4.0 systems by managing the external communication of the

[1] "F. Zezulka, P. Marcon, I. Vesely, and O. Sajdl, “Industry 4.0 – An Introduction in the phenomenon,” in IFAC-PapersOnLine, Elsevier B.V., 2016, pp. 8–12. doi: 10.1016/j.ifacol.2016.12.002".

asset with the connected, digital world in a standardized format. Ultimately, the AAS ensures the unique, unambiguous identification of its asset via unique identifiers to be addressable within the network, acting as a standardized and secure communication interface of the asset in the digital world as shown in Figure 4.

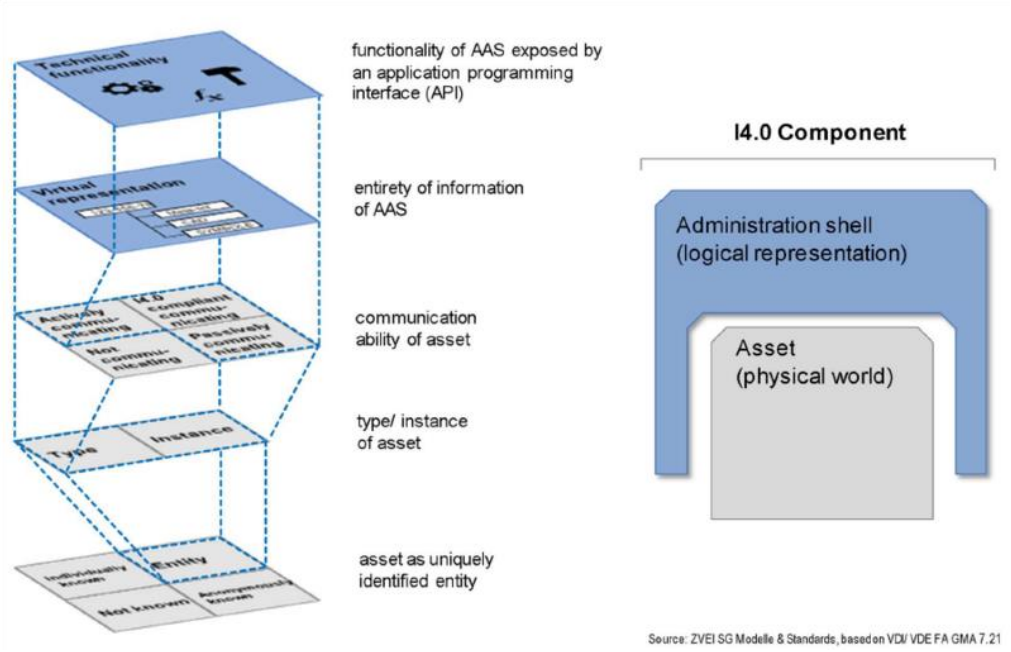

Figure 4 - AAS common layers and logical view

The AAS metamodel is a manufacturer-independent standardization that defines structural principles of the AAS in a formal manner (UML) with the aim of enabling information exchange via Industrie 4.0 communication protocols. The structure of the AAS usually consists of a number of domain-specific submodels that encapsulate all the information and functionalities of the asset, including its properties, capabilities, and references and relationships to other components or objects, among others. As shown in Figure 5, the information within the AAS follows a tree-based structure of properties in which submodels support the integration of different data sources of the asset, enabling the integration of external sources with proprietary interfaces by acting as reflexive interfaces or common points of interaction.

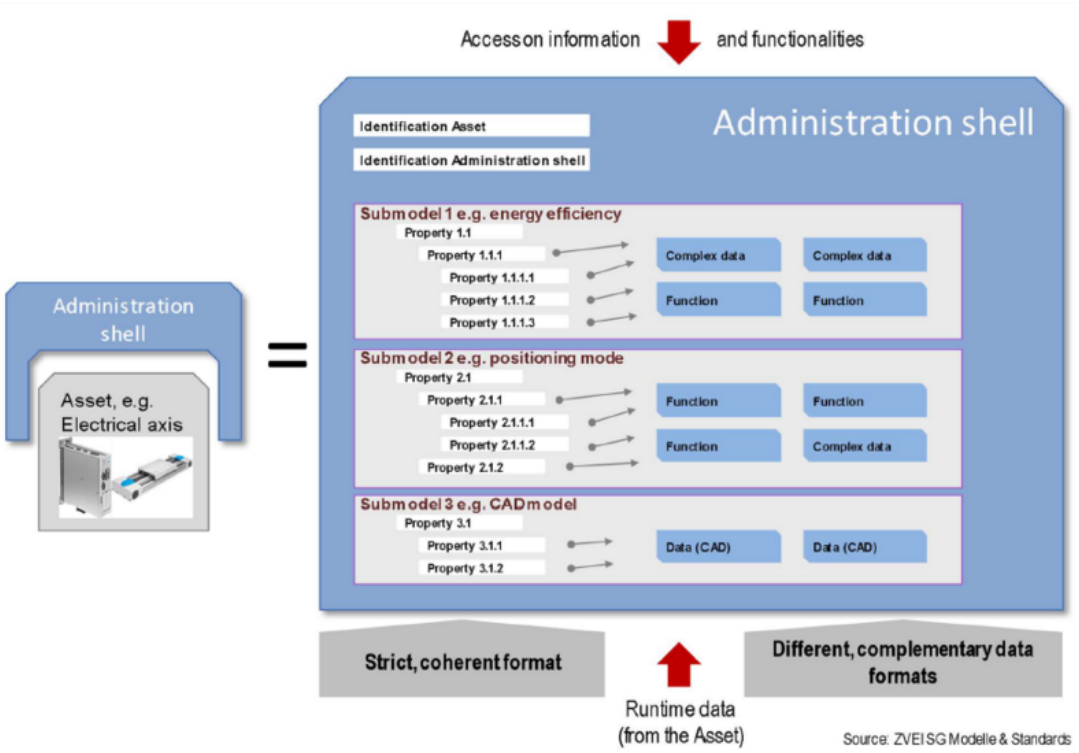

Figure 5 - AAS exemplary information structure

The AAS is supported by multiple serializations mappings that include XML, JSON, or OPC-UA. According to its behavior, three types of AASs can be distinguished: type I, passive AASs represented as serialized files containing static information; type II, reactive AASs existing as runtime instances; and type III, proactive AASs that show active behavior i.e., communication via standardized Industrie 4.0. language. On this line, the information model of the AAS support the integration of concept descriptions and semantic identifiers in the different elements of the AAS e.g., via ECLASS semantic identifiers, enabling the transition from implicit to explicit semantics in the communication between AASs in the connected, digital world.

3.4. BDVA Big Data Value Association

The European BDVA (Big Data Value Association) has designed a BDV Reference Model that serves as a common reference framework to locate Big Data technologies on the overall IT stack.

The Strategic Research and Innovation Agenda (SRIA) define the overall goals, main technical and non-technical priorities, and a research and innovation roadmap for the European contractual Public Private Partnership (cPPP) on Big Data Value.

The BDV Reference Model has been developed considering input from technical experts and stakeholders along the whole Big Data Value chain, as well as interactions with other related PPPs, in order to create a common understanding between all members interested in Big Data and Artificial Intelligence applications applied to the Smart Manufacturing Industry domain. It is structured into horizontal and vertical concerns which distinguishes between elements that are at the core of the BDVA and the features that are developed in strong collaboration with related European activities.

The Big Data Value Reference Model is shown in Figure 6.

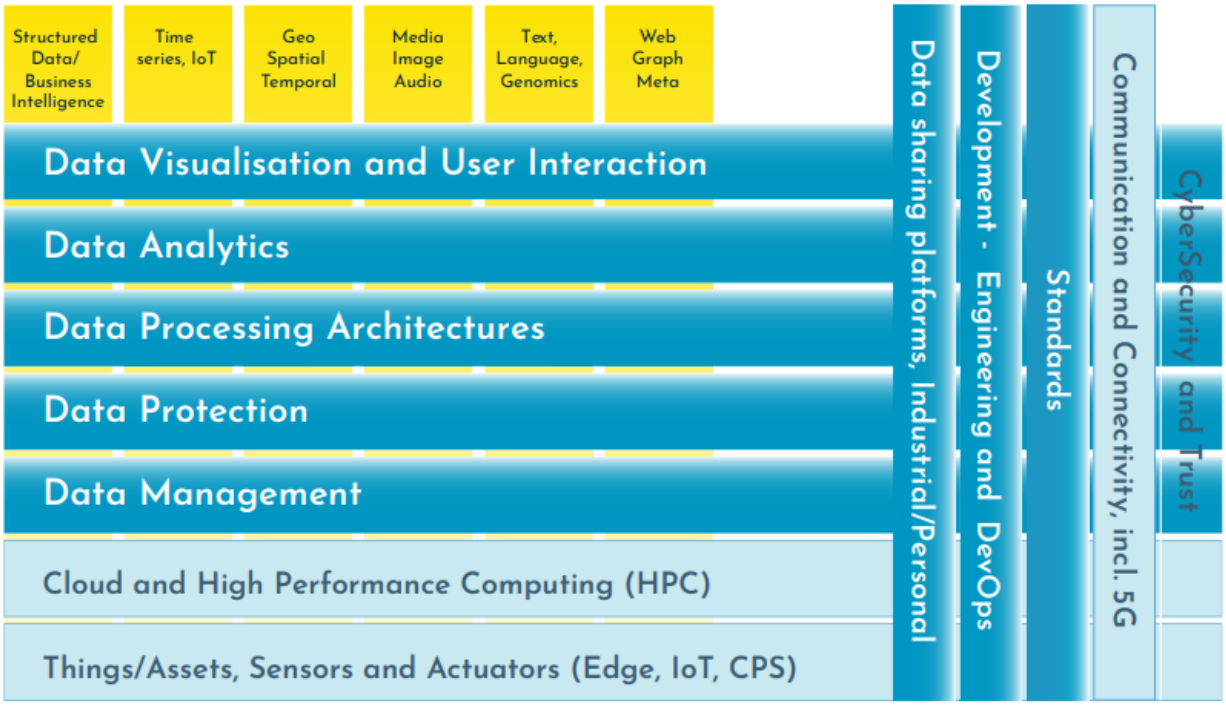

Figure 6 - Big Data Value Reference Model

The horizontal concerns cover specific aspects along the data processing chain, starting with data collection and ingestion, and extending to data visualization. It should be noted that the horizontal concerns do not imply a layered architecture. As an example, data visualization may be applied directly to collected data (the data management aspect) without the need for data processing and analytics.

The vertical concerns address cross-cutting issues, which may affect all the horizontal concerns. In addition, vertical concerns may also involve non-technical aspects.

The following technical elements are expressed in the BDV Reference Model:

- **Data Visualization and User Interaction**: Advanced visualization approaches for improved user experience. This technical priority is addressing the need or advanced means for visualization and user interaction capable to handle the continuously increasing complexity and size of data to support the user in exploring and understanding effectively Big Data.
- **Data Analytics**: Data analytics to improve data understanding, deep learning, and meaningfulness of data. The Data Analytics technical priority aims to progress data analytics technologies for Big Data in order to develop capabilities to turn Big Data into value, but also to make those approaches accessible to the wider public.
- **Data Processing Architectures**: Optimized and scalable architectures for analytics of both data-at-rest and data-in-motion with low latency delivering real-time analytics. This technical priority is motivated by fast development and adoption of Internet of Things (IoT) technologies that is one of the key drivers of the Big Data phenomenon with the need for processing immense amounts of sensor data streams.
- **Data Protection**: Privacy and anonymization mechanisms to facilitate data protection. This is related to data management and processing as it is a strong link here, but it can also be associated with the area of Cybersecurity.
- **Data Management:** Principles and techniques for data management. This technical priority is motivated by the data explosion that is mainly triggered by the increasing amount of data sources (e.g., sensors and social data) and their complexity in structure.

- **The Cloud and High Performance Computing (HPC)**: Effective Big Data processing and data management might imply the effective usage of Cloud and High Performance Computing infrastructures. This area is separately elaborated further in collaboration with the Cloud and High Performance Computing (ETP4HPC) communities.
- **IoT, CPS, Edge and Fog Computing**: A main source of Big Data is sensor data from an IoT context and actuator interaction in Cyber Physical Systems. In order to meet real-time needs, it will often be necessary to handle Big Data aspects at the edge of the system. This area is separately elaborated further in collaboration with the IoT (Alliance for Internet of Things Innovation (AIOTI)) and CPS communities.

The top elements, marked in yellow, are Big Data Types and Semantic identified, which often lead to the use of different techniques and mechanisms in the horizontal concerns and it is necessary to considered it for data analytics and data storage.

In addition, the Referent Model identifies some transversal elements on which Big Data RAs must be support:

- *Cybersecurity:* Big Data often need support to maintain security and trust beyond privacy and anonymization. The aspect of trust frequently has links to trust mechanisms such as blockchain technologies, smart contracts and various forms of encryption.
- *Communication and Connectivity:* Effective communication and connectivity mechanisms are necessary in providing support for Big Data. This area is separately further elaborated, along with various communication communities, such as the 5G community.
- *Standards:* Standardization of Big Data technology areas to facilitate data integration, sharing and interoperability.
- *Engineering and DevOps for building Big Data Value systems:* This topic will be elaborated in greater detail along with the NESSI Software and Service community.
- *Marketplaces, Industrial Data Platforms and Personal Data Platforms (IDPs/PDPs), Ecosystems for Data Sharing and Innovation Support:* Data platforms for data sharing include, in particular, IDPs and PDPs, but also other data sharing platforms like Research Data Platforms (RDPs) and Urban/City Data Platforms (UDPs). These platforms facilitate the efficient usage of a number of the horizontal and vertical Big Data areas, most notably data management, data processing, data protection and cybersecurity.

To this end, Big Data Value Association (BDVA), euRobotics, CLAIRE, ELLIS and EurAI have launched the creation of the AI, Data and Robotics Association (ADRA) [17]. ADRA represents the private stakeholder community of the AI, Data and Robotics co-programmed Partnership in Horizon Europe. The wider purpose of objectives and activities are Big Data Value, AI, Data and Robotics update in objectives and activities.

The association objective is to boost European AI (Artificial Intelligence), Data and Robotics research, development and innovation, and to foster value creation for business, citizens and the environment.

Moreover, the aim of this association is:

- Boosting European competitiveness, societal wellbeing and environmental aspects
- Promoting the widest and best uptake of AI, Data and Robotics technologies and services for public, professional, and personal use;
- Establishing the excellence in science and business in AI, Data and Robotics.

3.5. Industrial Internet Reference Architecture (IIRA)

The Industrial Internet Reference Architecture (IIRA) [18] is a standards-based open architecture for Industrial Internet of Things (IIoT) systems. The IIRA pretends to exploit its value thanks to its broad industry applicability in enhancing common understanding, drive interoperability, mapping technologies and standardizing development. IIRA description and its representation are generic and at a high level of abstraction, to better support the requisite of broad applicability to different sectors (e.g., energy, healthcare, manufacturing, and transportation). The IIRA ”distills and abstracts” common characteristics, features and patterns from use cases defined in the Industrial Internet Consortium (IIC, the promoter committee of IIRA) and elsewhere. It will be refined continually as feedback is gathered from its application in the testbeds developed in the IIC and real applications. The IIRA is also intended to be independent from nowadays technologies to be a conceptual reference point for future deployments, independently from the specific technology and not to tie the IIoT community to existing ones, removing possible barriers for future applied research.

IIRA has two declared purposes:

1. It is the foundational framework for technical documents promoted by the IIC.
2. For the broader IoT community (academics, practitioners and users), it provides guidance and assistance in the development, documentation, communication and deployment of IIoT systems.

The IIRA deploys an architectural description of IIoT systems using ISO/IEC/IEEE 42010:2022 *Software, systems and enterprise — Architecture description* [19], presenting its concept according to different perspectives (intuitively defined as ”Business view”, ”Usage view”, ”Functional view”, and ”Implementation view”). A short summary of the ”Functional View” is here presented.

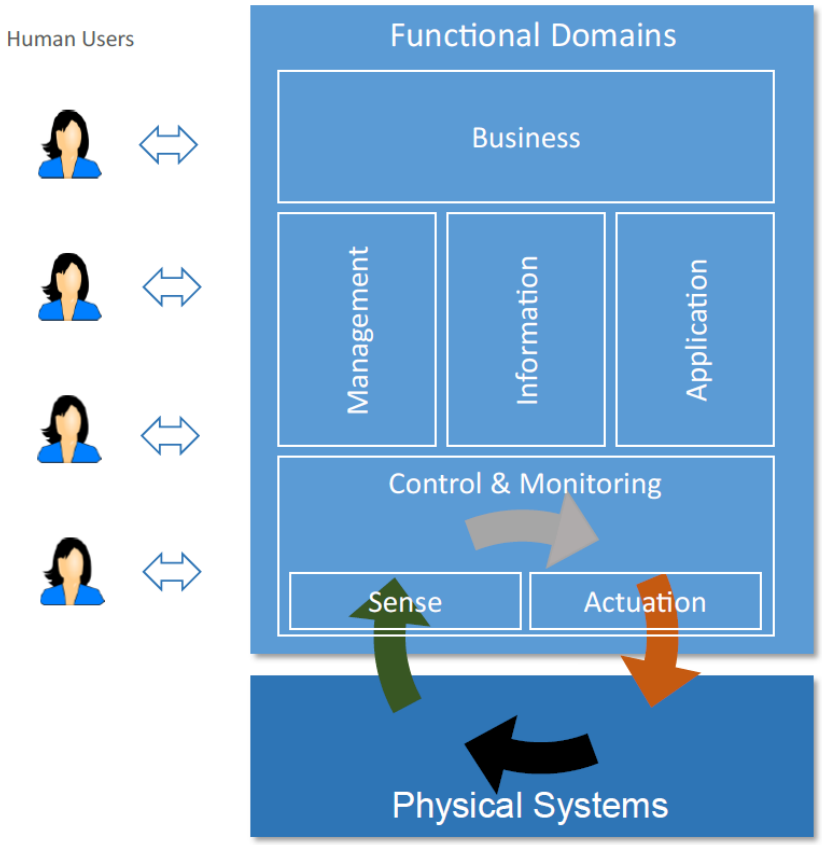

Figure 7 - Functional view of IIRA

The functional viewpoint of IIRA depicted in Figure 7 provides a five-layer description of the functions in an industrial system, their interrelation, structure, and interactions:

- Control And Monitoring domain: functions executed by industrial controllers and similar simple devices (e.g., sensing and actuation in closed-loop systems, monitoring functions and IoT communication features) as per Figure 8, where the "IoT Gateway" block refers to those cases where a gateway is needed to connect the assets to an informative system (e.g., to connect a Profibus-communicating machine).

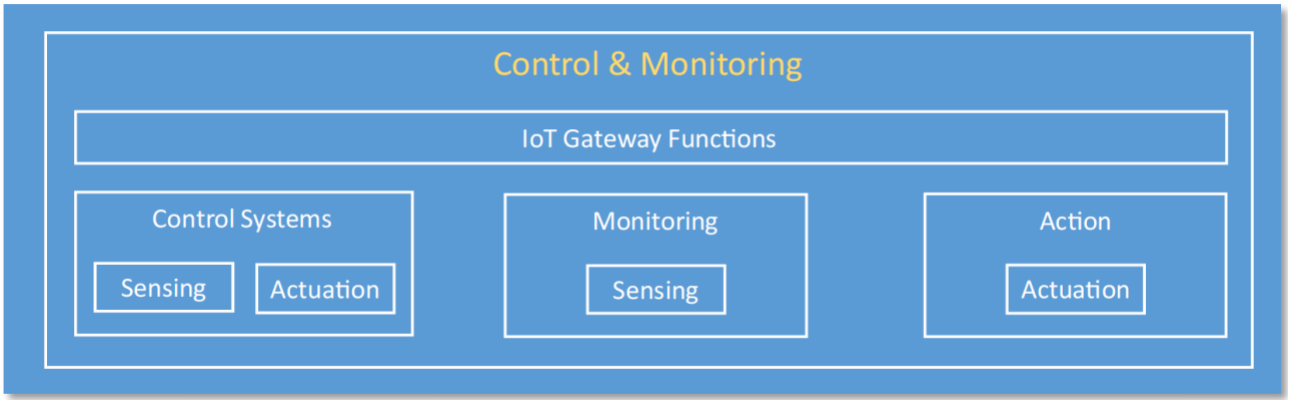

Figure 8 - Functional decomposition of Control and Monitoring domain

- System Management domain: functions that operate components throughout their life cycle (e.g., and deployment, monitoring, diagnostics of IIoT systems, as well as orchestration of different IIoT subcomponents, as depicted in Figure 9).

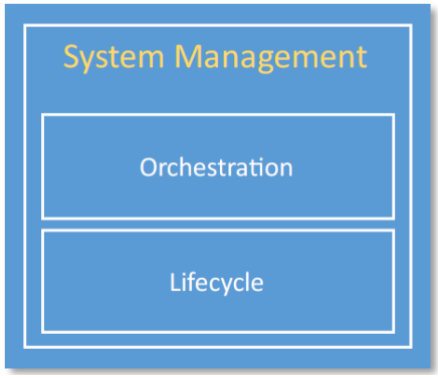

Figure 9 - Functional decomposition of System Management domain

- Information domain: functions for gathering data from the other domains (e.g., data collection and management and analytics).
- Application domain: functions implementing application logic to realize particular business functionalities (e.g., activity flows, rules and models, anyway exposed throughout User Interfaces, UIs, and Application to Program Interfaces, API).
- Business domain: functions business logics interacting with backend systems (examples of systems framed into this domain include Enterprise Resource Planning, ERP, Customer Relationship Management, CRM, Product Lifecycle Management, PLM, some functionalities of Manufacturing Execution Systems, MESs, and Human Resource Management, HRM). Figure 10 depicts the core modules and relations of the Information, Application and Business Domain.

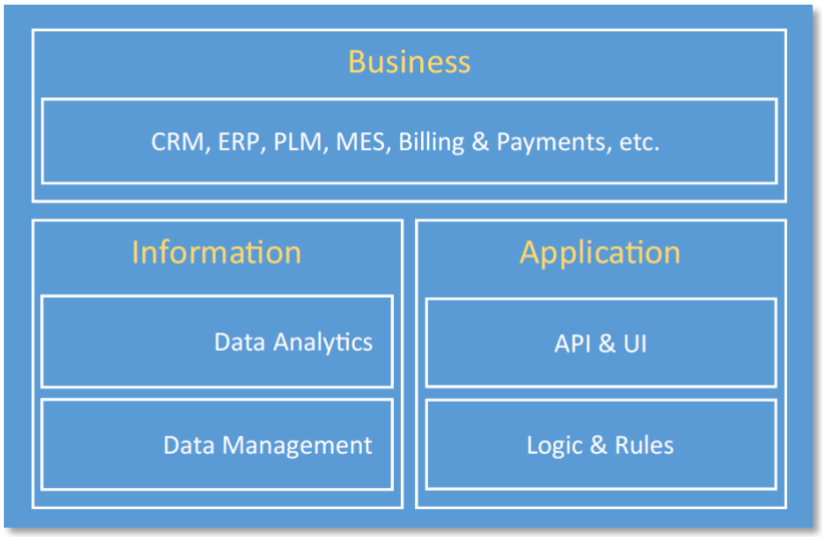

Figure 10 - Functional decomposition of Information, Application and Business domains

Figure 11 shows the alignment between IIRA and the manufacturing-focused RAMI4.0 in terms of their functional viewpoints and layers. Both models represent a taxonomy of functions associated with different layers of an enterprise, in a manner similar to the functional hierarchy model of ISA-95.

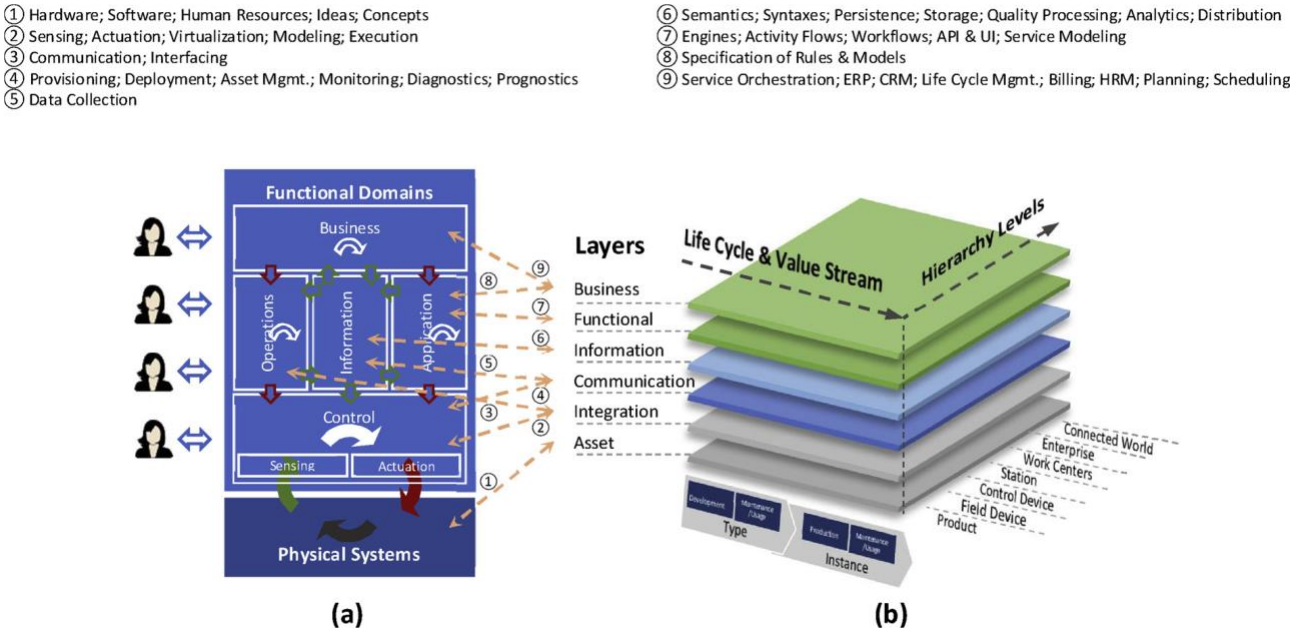

Figure 11 - Interrelations between RAMI4.0 and IIRA

Both the functional layers of RAMI4.0 and the functional viewpoint of IIRA begin from the physical assets and data acquisition and communication from the digital world to higher-level functions, data management and analytics, computational capabilities, services and business operations. CPS technologies are transforming this hierarchical representation of functions into a networked structure where the different functions can

take place in a decentralized fashion and through a variety of micro-services which connects them directly to the physical assets, thanks to the IIoT communication capabilities[1] .

3.6. OPEN DEI

The digital transformation strategy of the European Union identifies as a priority the creation of common data platforms based on a unified architecture and an established standard. The OPEN DEI "Aligning Reference Architectures, Open Platforms and Large-Scale Pilots in Digitising European Industry" project focuses on "Platforms and Pilots" to support the implementation of next generation digital platforms. The OPEN DEI main objective was to become an essential pillar of the development of Digitising European Industry policies, implementing four action lines:

- **Platform building:** Comparing reference architectures and open-source reference implementations, enabling a unified industrial data platform.
- **Large scale piloting:** Contributing to a digital maturity model, creating a set of assessment methods and a migration journey benchmarking tool.
- **Data ecosystem building:** Enabling an innovation and collaboration platform, forging a European network of DIHs, contributing to industrial skills catalogue and observatory.
- **Standardisation:** Conducting cross-domain surveys, performing promotion and implementation, building alliances with existing EU and standard developing organisations.

OPEN DEI[2] has the objective to harmonize and coordinate different Digital Transformation (DT) approaches under a common Reference Architecture Framework (RAF), which combines knowledge and tools, capable to foster knowledge transfer and best practices exchanges on how systems supporting DT can be architected, crossing the boundaries of specific applicative sectors. OPEN DEI proposes a conceptual model for integrated data-driven services for Digital Transformation pathways, to guide their planning, development, operation and maintenance by adopting organizations. The model is modular and comprises loosely coupled service components interconnected through a shared common data infrastructure.

The OPEN DEI project has defined the approach for designing a common Reference Architecture Framework able to describe the Cross Domain Digital Transformation. Data-driven pipelines and workflows management is nowadays crucial for data gathering, processing, and decision support. OPEN DEI has adopted the following 6C architecture to cope with this complexity, based on the following pillars (using a bottom-up reading):

- **Connection**, making data available from/to different networks, connecting systems and digital platforms, starting from the capability to make data available from/to different physical and digital assets.
- **Cyber**, modeling and in-memory based solutions to convert data into information, leveraging several information conversion mechanisms.
- **Computing**, storing and using data on the edge or on cloud.
- **Content/Context**, correlating collected data for extracting information, creating a digital space for data-information continuum, not something to push out to one side of the adopted information architecture.
- **Community**, sharing data between people and connecting stakeholders for solving collaboration needs.

[1] Mohsen Moghaddam, Marissa N. Cadavid, C. Robert Kenley, Abhijit V. Deshmukh, Reference architectures for smart manufacturing: A critical review, Journal of Manufacturing Systems, Volume 49, 2018, Pages 215-225, ISSN 0278-6125, doi: 10.1016/j.jmsy.2018.10.

[2] "OPEN DEI project website," [Online]. Available: https://www.opendei.eu/resources/. [Accessed 28 March 2023].

- **Customization** to add value to data following each own user perspective and to match their expectations.

The above-mentioned 6C Architecture principles drive the design of the OPEN DEI RAF, developed around the main concept of Data Spaces in which data is shared (published and accessed), identifying three main different layers described in the following using a bottom-up approach:

- **Field Level Data Spaces**, it includes the Smart World Services able to collect data and support the interaction with the IoT Systems, Automation and Smart Assets (robots, machinery, and related operations) and Human Systems (manual operations, supervision, and control, etc.).
- **Edge Level Data Spaces**, it defines the typical edge operations from the data acquisition (from the logical perspective) to the data processing through the data brokering. The edge services will play a key role also for data analytics (i.e., validating and improving models for data analysis).
- **Cloud Level Data Spaces**, it includes data storage, data integration and data intelligence operations on the cloud. The cloud services will process big data, deploy algorithms, integrate different source platforms and services, provide advanced services such as AI prediction and reasoning.

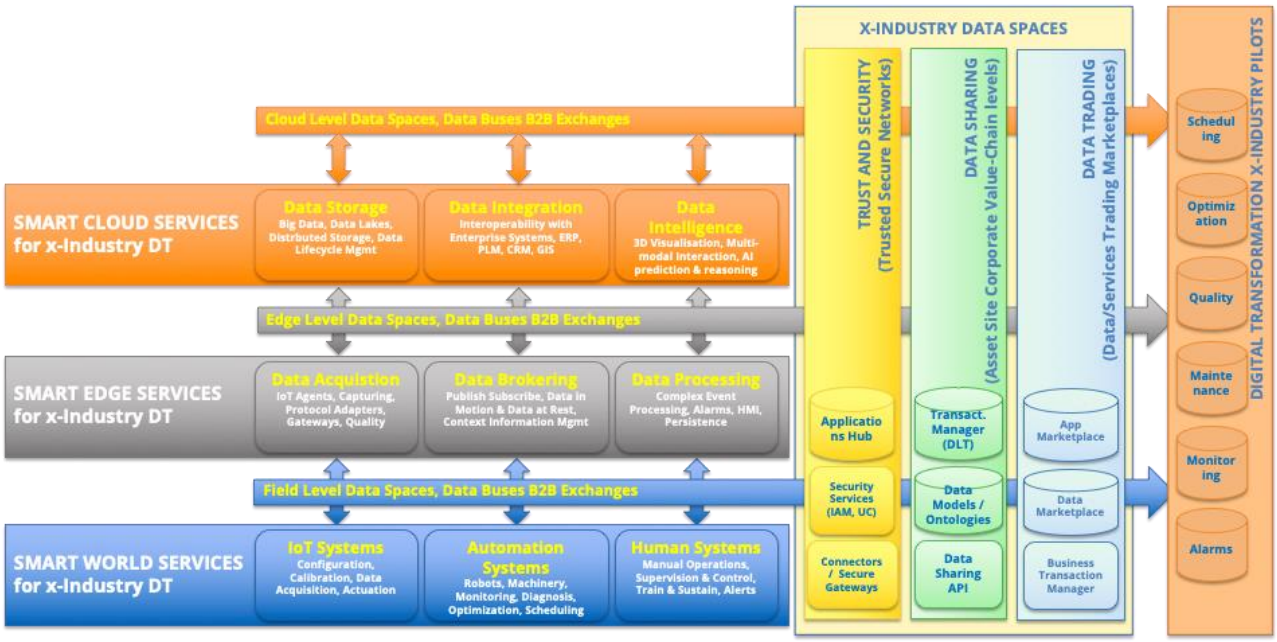

Figure 12 - OPEN DEI Reference Architecture Framework

Furthermore, all these horizontal Data Spaces tiers will feed an orthogonal dimension, named X-Industry Data Spaces, structured in:

- Trust and Security, including technical frameworks and infrastructures that complements the previous ones, to support trusted and secure exchange, which embraces:
    - Applications Hub, an infrastructure collecting the recipes required to provision applications (e.g. deployment, configuration and activation) in a manner that related data access/usage control policies can be enforced.
    - Security Services, a technical framework to support Identity Access Management, Usage Control and other security services.
    - Connectors and Secure Gateways, a technical framework for trusted connection among involved parties.

- Data Sharing, incorporating technical frameworks and infrastructures for an effective and auditable data sharing, which more specifically embraces:
    - Transaction Manager, a distributed ledger/blockchain infrastructure for logging selected data sharing transactions.
    - Data Models and Ontologies to leverage common standard and information representations.
    - Data Sharing API, a technical framework for effective data sharing.
- Data Trading, including technical frameworks and infrastructures for the trading (offering, monetization) of data, which embraces:
    - App Marketplace, offering applications and building blocks which can be integrated in plug&play mode to enrich existing data spaces.
    - Data Marketplace, offerings data resources with associated terms and conditions, including data usage/access control policies as well as pricing schemas.
    - Business Support Functions, enabling data/applications usage accounting as well as implementing Clearing House, Payment and Billing functions.

Finally, all the mentioned layers serve the realization of Digital Transformation X-Industry Pilots, to enable applications for supporting business scenarios from experiments. The OPEN DEI initiative and its Reference Architecture Framework fully support the Data Space conceptualization and implementation.

3.7. Arrowhead Framework

3.7.1. The local automation cloud approach

Arrowhead (https://arrowhead.eu/) follows the idea to provide local computational heads which can be used as cloud computing resources by devices connected to the head. For example, a telecom base station can be such a computational head.

To address the strong real-time, specific engineering, and security requirements of automation systems, Arrowhead introduced the concept of local clouds. This enables implementations to meet strong requirements on protecting the automation functionalities from, e.g., network traffic that changes communication latency and network security attacks. It is further foreseen that integrated automation systems will comprise much larger number of actors, devices and systems than today. Thus scalability becomes important. Together this puts a strong push on engineering costs.

The local cloud concept takes the view that some specific automation actions should be supported. These actions have strong requirements on real time, ease of engineering, operation and maintenance plus security. The local cloud idea is to let the local cloud be self contained and only involve the devices and systems required to perform the desired automation tasks. Thus viewing the cloud border as a protective fence towards any other internet communication.

Within each of these local clouds the necessary infrastructure for building IoT based SoS automation is provided. This at the minimum includes:

- Service registry services
- Orchestration services
- Authorisation services

- Loose coupling
- Late binding
- Manageability
- Authentication
- Authorization
- Low latency
- Data ownership
- Privacy
- …

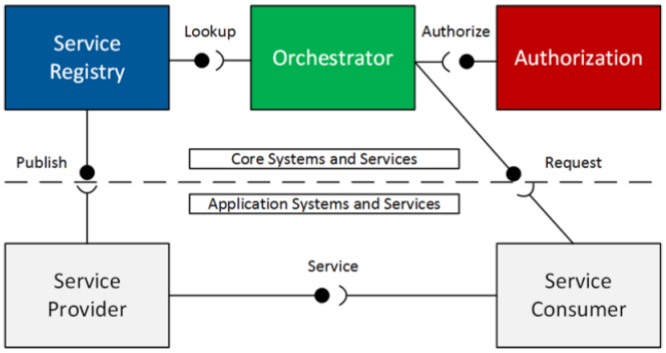

Figure 13 – Local Cloud structure

The local cloud objective is to:

• provide a way of making service discovery external to the local cloud where the discovery process is initiated

• enable orchestration push and pull distribution of service exchange end points between systems residing in different local clouds

• enable system authentication and service consumption authorisation between systems residing in different local clouds

The local cloud concept is visualised in the following figure as a number of local clouds with largely independent automation tasks. Higher-level cloud coordination is supported by inter-cloud communication.

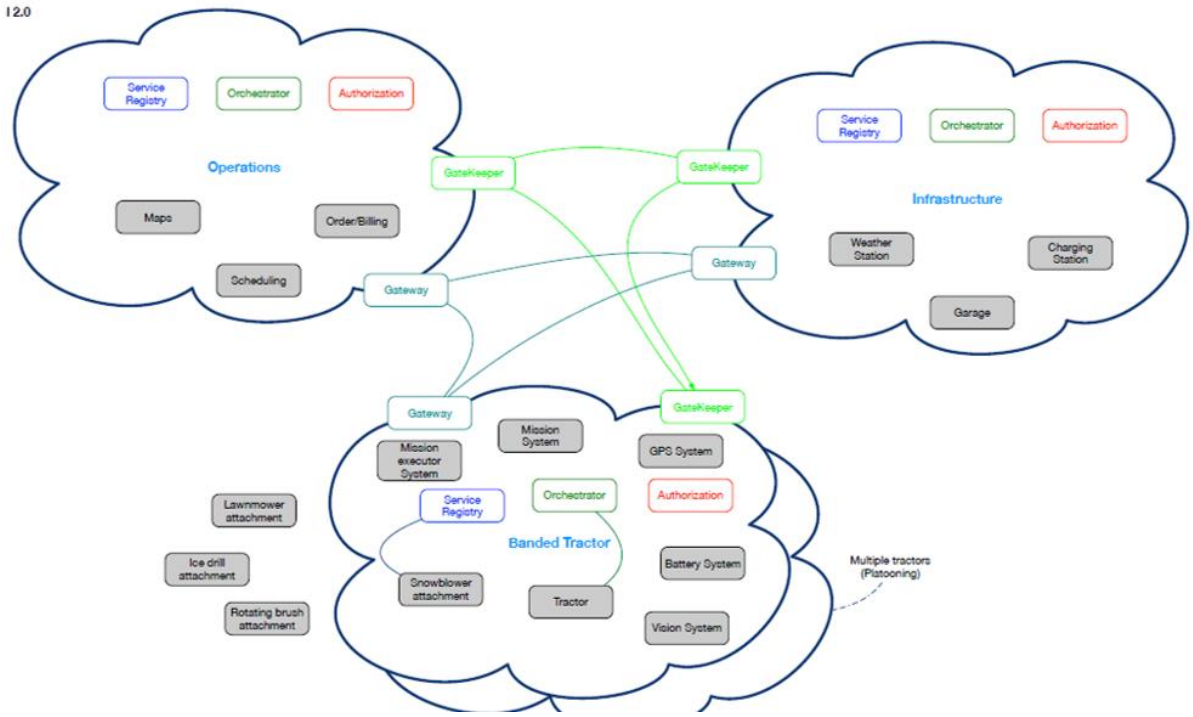

Figure 14 – System of Systems

Each cloud can be complemented by a number of automation supportive services. Together with a number of application systems and their services the desired automation application functionality can be engineered, deployed, operated, and maintained in run-time.

It is obvious that many applications may require multiple local clouds to build a total automation system. Thus inter-cloud communication has to be ssupported while maintaining local cloud internal properties.

3.7.2. Architecture fundamentals

The objective of the Arrowhead Framework architecture is to facilitate the creation of local automation clouds. Thus enabling local real time performance and security, paired with simple and cheap engineering, while simultaneously enabling scalability through multi-cloud interaction.

The architecture addresses the move from large monolithic organisations towards multi-stakeholder cooperation where cooperation is fostered by market requirements. This is to support the high-level topics in today's society such as sustainability, flexibility, efficiency, and competitiveness in production.

Devices in such local clouds are considered to be IoT devices speaking at least one SOA (Service-Oriented Architecture) protocol.

Thus the following properties are important

- Loose coupling
  - Autonomy - a service exchange is not supervised
  - Distributed - services are distributed over several devices
  - A system is responsible, owns the information, and can decide whom to share with
- Late binding
  - Possible to use information any time by connecting to the correct resource at a given time
- Lookup
  - Publish and register services to notify others about endpoints (how to reach me)
  - Discover others that I comply with (expected/wanted ServiceType)

The design of the Arrowhead Framework is further based on the following fundamentals:

- A system producing a service has the initial authority of its own service offering
- Information assurance shall be at the service exchange level
- Information centric networking

The Arrowhead Framework allows for:

- A Publish-Subscribe approach
- Both the Push and Pull approaches
- Dynamic creation of new services and their subsequent usage

3.7.3. Security in local clouds

One of the arguments for local clouds is the possibility to provide a local security "fencing" around the local cloud and appropriate authenticating, authorisation, and encryption of services within the local cloud. Such "fencing" does need ways of keeping non-internal network activity outside the local cloud. Thus a secure local cloud have should have firewalls at the interface to other clouds which support blocking of "external" traffic to come into the local cloud.

Such security fencing is dependent on control and minimising of communication in and out of the local cloud. This will reduce the number of holes made through the firewall in the local cloud security fence. The provision of all the administrative functionalities for a SOA-based system of systems within the local cloud is thus an important design principle. Thus there is no need for external resources in establishing the local cloud and its SOA-based infrastructure. This requirement is supported by the idea of having the mandatory core services deployed within a local cloud

3.7.4. Automation support core systems

To facilitate automation application design, engineering, and operation, the Arrowhead Framework further contains a number of automation support services provided by the related automation support core systems. The objective of the automation support core systems is to support:

- The implementation of "plant" automation. Here plant could be the infrastructure of, e.g., a car manufacturing plant, a mine, an infrastructure
- Housekeeping within the local cloud
- Security and bootstrapping of a local cloud
- Inter-cloud service exchange
- System and service interoperability

3.7.5. Data encryption

Data encryption can be provided either by enforcing IP communication encryption or relying on SOA protocol provided payload encryption. Local clouds should allow for the use of a mixture of the known encryption approaches. However, if an application has need for services using different encryption technologies, a translation between different encryptions may be needed. This might open up man-in-the-middle security holes.

The above properties, fundamentals, and functionalities are provided by the Arrowhead Framework through

- A minimal set of mandatory services to create a System of Systems
- A set of automation support services - facilitating design of application System of Systems

3.7.6. An example: SMEI5.0 SHOP FLOOR USE CASE

In this part, we present a collaborative SMEI5.0 use case that is compliant with I5.0 core values: a) Resilience, b) Sustainability, c) Human centricity.

The proposed SMEI5.0 consists of fully integrated and collaborative Smart Building Automation and Energy Systems, Smart Logistic Robots, Smart Production Line Systems, Smart Human Workers / Operator Aide Systems, and Smart Workflow Management Systems on the shop floor as shown in following figure. Integrating such a diverse set of industrial systems in SMEI5.0 requires extensive AI models and algorithms. Arrowhead proposes grouping industrial IoT and systems that perform similar automation tasks on the shop floor into LC-based edge systems of systems. Every edge system of systems has its own edge AI that is most optimal for its dataset. Then the Smart Manufacturing Workflow Management Systems dynamically generate the production workflow recipes by collaboration among all the various edge AIs of each respective edge system of systems.

The distributed local cloud-based architecture of the framework ensures total data privacy, as all data is stored in the local cloud at the edge. The use of the Arrowhead framework to realize the SMEI5.0 shop floor use case makes the ecosystem resilient by design, as the framework already provides service-level authorization, composability, dynamic orchestration, and access control, which are required for a resilient system operation.

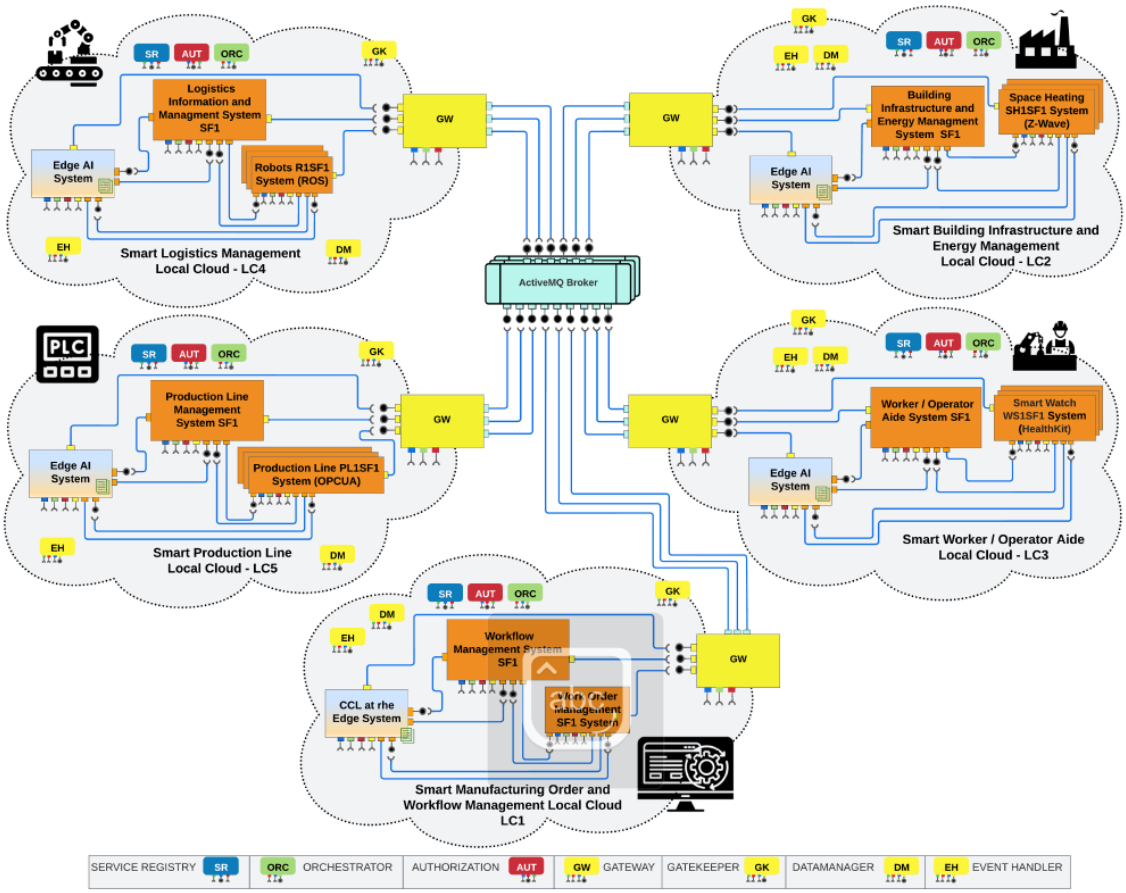

Figure 15 – System of Systems

In this way the local cloud concept with inter-cloud communication and interaction seams to have a number of interesting properties with regards to scaling.

The architecture and a number of the core services implementing the architecture are open source.

The Arrowhead Framework community development site is located at http://forge.soa4d.org/plugins/mediawiki/wiki/arrowhead-f/index.php/Main_Page also reachable from http://www.arrowhead.eu.

3.8. Edge Computing

Many major cloud service providers now offer edge computing services. For example, AWS has rolled out a comprehensive set of services that facilitate edge computing for a variety of use cases. They essentially extend their cloud infrastructure to the edge and allow data centers to be set up locally in specific cities, on premises and/or within 5G networks.

Services like these, from AWS and other cloud service providers, bring more options, flexibility and simplicity for edge computing initiatives. In turn, these services allow your organization to start quickly by leveraging on-demand infrastructure and to evolve efficiently by maintaining a standardized, repeatable environment.

In a distributed architecture that spans from the cloud to the edge, it is required to facilitate data processing throughout every layer of the ecosystem. All layers need to share a real-time understanding of the data, and any layer should be able to run in isolation in the event of loss of connectivity.

This means, there is a need for a database that natively distributes its storage and workload across the various tiers of an edge architecture. The database must also have the ability to instantly replicate and synchronize data across database instances, whether they're in the cloud or in an edge data center.

In addition, the database needs to be embeddable. Data storage should be integrated directly to the edge device in order to facilitate data processing when completely offline. As such, the embedded database must be able to operate without any central cloud control point, and it must automatically synchronize with the rest of your data ecosystem when connectivity returns.

Furthermore, synchronization must be bi-directional and controllable in order to provide a secure and optimal flow of data throughout your edge architecture. For example, in a smart factory scenario, high velocity data captured from an assembly line can be processed and analyzed at the edge, but – for network bandwidth efficiency – only aggregated data is synchronized to the cloud for ultimate storage.

3.8.1. IBM Edge computing reference architecture

The edge computing architecture has four regions: the device region, the edge server/gateway region, the edge network or micro data center, and the enterprise hybrid multicloud region. The edge server not only connects all edge devices in a secure manner but also allows for management of all those devices.

Edge computing regions

The primary regions for edge computing are as follows:

- Devices. The edge and IoT devices are equipped to run analytics, apply AI rules, and even store some data locally to support operations at the edge. The devices could handle analysis and real-time inferencing without involvement of the edge server or the enterprise region.
- Edge server or gateway. Edge servers are used to deploy apps to the devices. They are in constant communication with the devices by using agents that are installed on each of the devices. Thousands of edge servers maintain a pulse on the millions of devices. If something more than inferencing is needed, data from the devices is sent to the edge server for further analysis.
- Edge network or micro data center. New networking technologies have resulted in the edge network or micro data center, which can be viewed as a local cloud for devices to communicate with. The edge network reduces the distance that data from the devices must travel and thus decreases latency and bandwith issues especially with the advent of 5G. This region also offers more analytical capabilities and more storage for models.
- Enterprise hybrid multicloud. This region offers the classic enterprise-level model storage and management, device management, and especially enterprise-level analytics and dashboards. This region can be hosted in the cloud or in an on-premises data center.

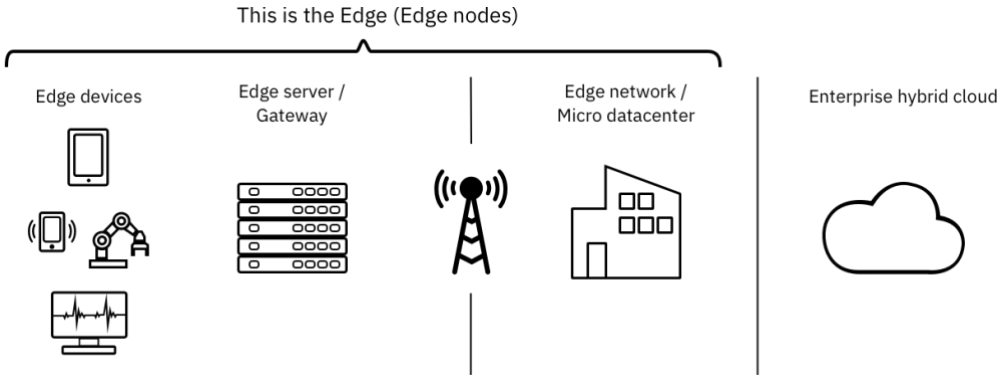

Figure 16 – Edge computing layers

Edge computing pushes the intelligence, processing power, and communication capabilities of an edge server or gateway directly to IoT and other edge devices. Edge extends IoT to a much larger scale.

3.8.2. IBM edge architecture principles

The IBM Edge computing architecture builds on open source technologies and security. It is rooted in four principles: the need to secure your data, drive innovative solutions, develop portable solutions based on open-source technologies, and manage at scale.

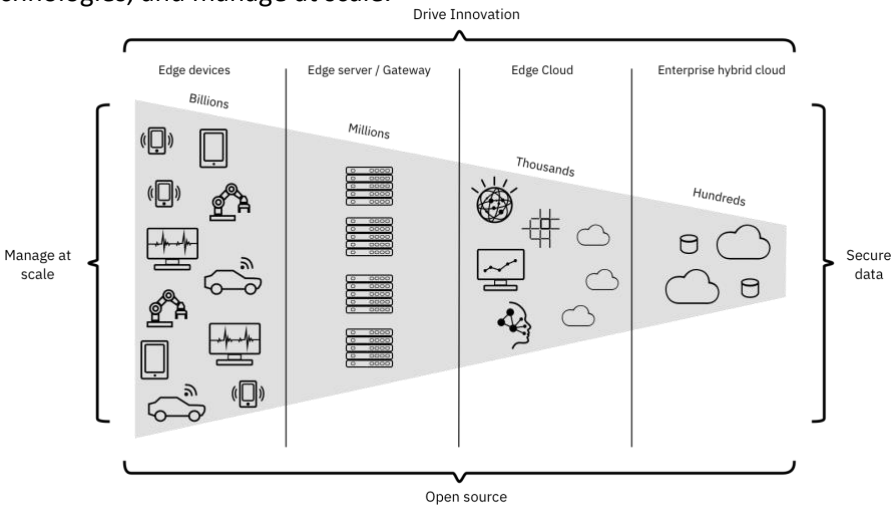

Figure 17 – Edge computing layers: data view

Manage at scale

With so many devices out there, managing edge workloads at massive scale and providing visibility, control, and automation is a challenge. IBM Edge Application Manager products offer the common management and orchestration across clusters and edge, policy-based role and compliance management, and event and service management.

Create a portable solution with open-source technology

The portability of the solution is key across private, public, and edge cloud. This portability is possible by using open technologies like Docker, Kubernetes, and the Open Horizon project. IBM Edge Application Manager offers open container technology for app delivery; open source middleware for data, messaging, and caching services; OpenPOWER, x86, and Advanced RISC Machines (ARM) hardware; and multitier and multicloud support.

Secure the data

As so much data is generated and transmitted, data protection takes on a whole new meaning in the context of security and compliance. Because data doesn't need to travel long distances, edge inherently mitigates the risk factors. IBM Edge Application Manager offers a focus on policy and privacy for regulated industries, signed workloads with runtime isolation and data encryption, and compliance supported with reporting and auditing.

Drive innovation

The need for machine learning in edge cloud is pushing the use of services such as IBM Watson® AI, IBM Intelligent Video Analytics (IVA), IoT services, and other specialized platform services that promise to drive innovation. IBM Edge Application Manager products offer a rich catalog of services, such as Watson AI, IVA, and IoT; function-specific infrastructure access, such as graphics processing unit (GPU) and field-programmable gate array (FPGA); and machine learning and deep learning in the cloud and at the edge.

3.8.3. IBM Edge computing reference architecture

The Edge computing reference architecture requires the ability to deploy scalable apps at the edge. It uses IBM Cloud® Internet of Things (IoT), data, and AI services to analyze and visualize the insights that are gained from analyzing vast amounts of data.

By using the Model Management Service, you can have independent lifecycles for your code and for your data. The Model Management Service manages the lifecycle of data files on edge nodes or devices, remotely and independently from code updates. In general, the service provides facilities to securely send any data files to and from the edge nodes.

Two topologies are at work, as depicted by the two reference architectures. The first is where devices with agents can communicate with other components to securely orchestrate software lifecycle management. Another common scenario is where data is sent to an edge server, where advanced visualization and AI analytics are done.

IBM Edge Application Manager for Devices is installed on a Kubernetes cluster. Devices are registered and authenticated directly within Horizon exchange. Agents are installed on all edge devices that support ARM and other architectures that run Docker. These agents communicate with other components to securely orchestrate software lifecycle management on their machines. In the reference architecture, the focus is on the outermost edge, the edge devices region.

Edge workloads are microservices that are packaged as containers, either individually or as a collection. Workloads are placed on individual nodes as the result of an agreement that is negotiated between the device agent and an agreement-bot (AgBot).

Negotiations are predicated on a set of expressed conditions, which are either a pattern or a policy. The agent and the AgBot actively monitor that the negotiated agreements remain valid in the presence of changing conditions.

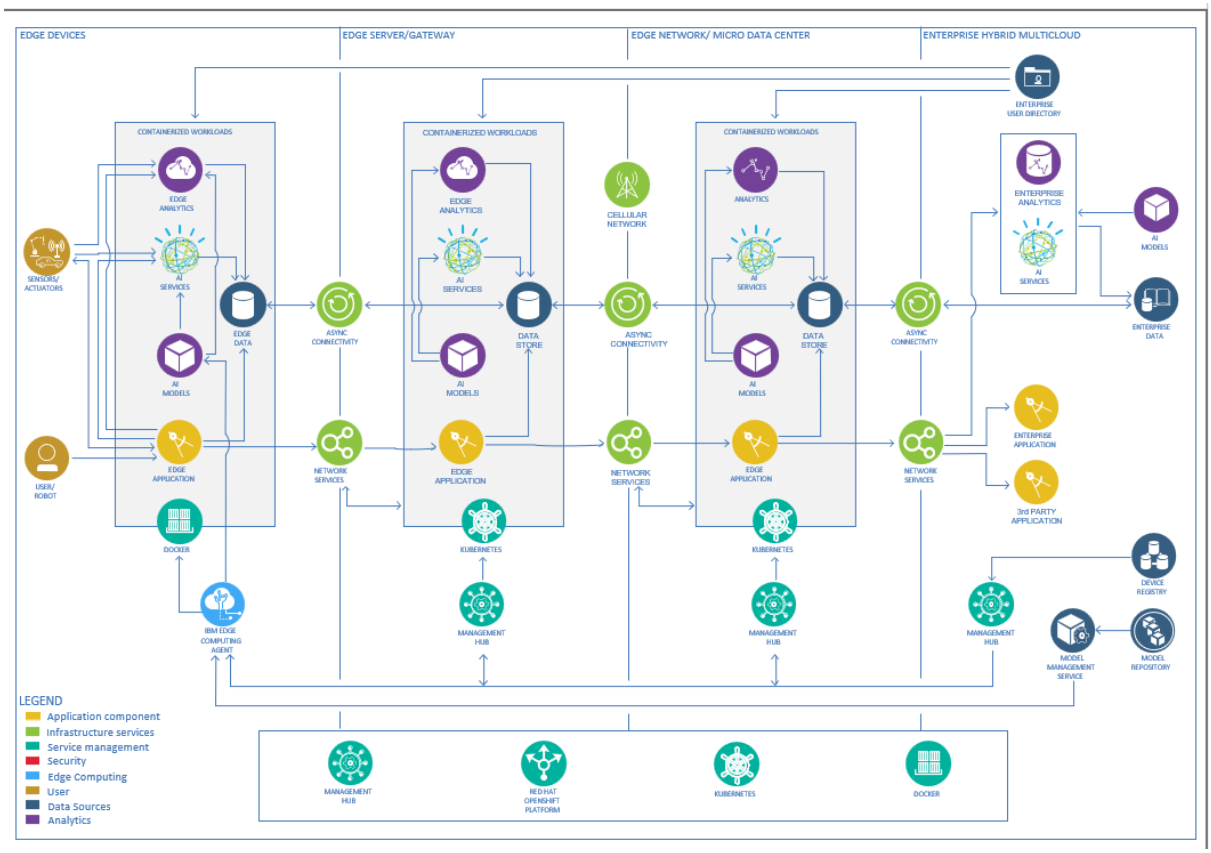

Figure 18 – Edge computing infrastructure

Nonfunctional requirements

Edge computing architectures must satisfy several nonfunctional requirements.

Security

All components in an edge solution must meet regulatory, compliance, and local security standards. All data that is in transit must be sent over secure protocols. When devices are onboarded, use keys, certificates, or both.

Service management and operations

- Availability. Availability is key in an edge computing architecture. The edge devices must remain functioning and the solution must continue to operate when and where there is no connectivity. It is important to eliminate all single points of failure (SPOFs) in the solution.
- Latency. Because edge computing is so tightly tied to the network, you commonly hear about latency and bandwidth. Getting the data from the devices and acting upon that data expeditiously is critical. Reduced latency is one of the primary requirements in any edge solution.
- Resiliency. The ability of a system to manage itself is important. This requirement refers to internal failures as opposed to external ones. When conditions change, are the agreements immediately terminated or refreshed? Make sure that agents and AgBots are always in a well-defined state.
- Scalability. Scalability must be inherently built into any edge solution. The abilities to register and deploy any number of devices and to scale edge servers are the most basic of requirements. Deployment and management at scale is a given.
- Maintainability. Given thousands of devices, often in remote locations, maintaining them on a regular basis becomes an important requirement. Having a planned maintenance schedule with specific maintenance scenarios addresses this requirement. But what about unplanned maintenance? One might argue that unplanned maintenance is exactly what edge solutions are designed to solve and handle.

DevOps

While maintainability is an operations function, manageability is more of an administration function. Decisions about when devices need to be upgraded, when policies must be updated, or when new product releases are rolled out all affect manageability.

3.9. Real-Time Location Systems (RTLS) in Manufacturing

3.9.1. Introduction

Most of existing reference architectures is related to the static data sources, i.e. the data is collected from the sensors which are fixed on the machines, or in the shop-floor, measuring only static values of parameters, but not their spatial dynamics kinematics. However, the expansion of the sensing technologies which are measuring movements (various types of tracking) is changing the requirements/needs towards more dynamic processing of sensor data.

Real-Time Location Systems (RTLS) are one of such examples. They have been around for two decades now, but recently they have begun to proliferate throughout the manufacturing industry. A key part of the Industry 4.0 movement, RTLS offers manufacturers a chance to pursue optimum efficiency across their industrial sites and yards.

RTLS has been used across manufacturing, logistics, and aerospace industries for many years, but is beginning to move into a wide variety of industries including healthcare, defence, and even office spaces. Its main purpose is to track the location of assets (anything from machines, vehicles, stock, tools and people) in

real-time, creating a digital overview of the plant/factory/yard/hanger. This new “digital twin” can allow for simpler management of the site by predicting errors, unnecessary tasks, resource shortfalls, employee malpractices and more. Boundaries can also be defined within the workplace, helping with safety practices, and assets and personnel can be quickly located and redirected. By employing an RTLS system, almost any manufacturing site can root out inefficiencies and begin to automate many time-consuming processes.

3.9.2. RTLS structure

There are three key components to an RTLS system: transmitters, sensors, and software. By setting up sensors around a site and tagging various assets, one can transmit location data to a central data hub for processing, either on-site or in the cloud. Specialized software can then build-up an image of the space and deliver insights into activity on the site.

There are a wide variety of sensor and tag types which vary depending on the environment they are deployed in, and the type of tracking required. RFID tags, for example, are a type of tag that is ubiquitous in manufacturing and logistics, and are very cheap and easy to us, but they offer limited functionality. UWB tags, as another example, are extremely useful in factory settings as they are “active” tags, sending out signals over a long distance, even offering 3D positioning to centimetre-level accuracy in some cases. They also suffer from far little interference from machinery and Wi-Fi signals.

Other examples of RTLS technology include BLE (Bluetooth Low Energy), Wi-Fi and GPS. Each with their own benefits and disbenefits.

Many of these sensors and tags will contain software elements, but the most important software handles the interpretation and visualization of the gathered data. Some key questions to ask when looking into any RTLS software package are:

- Floorplans: Can the software incorporate the site’s current floorplans?
- Solid reporting: Does the software have good analytics and reporting features “out-of-the-box”?
- Is the software hosted on site or in the cloud?
- Will the software scale with the business?
- Is support and training offered?

3.9.3. RTLS for Manufacturing

As RTLS systems are being used differently in a wide variety of industries, some of the most common use cases across different sectors are provided in the following text. Examples are simply the most obvious ways in which RTLS is being used after initial setup,

Aerospace Industry

RTLS is currently used to track the aircraft assembly process, including the location of tooling kits and various parts and personnel. Managing limited numbers of expensive tools, worker safety, and production logistics helps to reduce the overall spend on these complex production processes. Many of the checks performed on aeroplanes can also be changed to a “when needed” model as opposed to regular scheduled checks. The ability for RTLS systems to observe parts and machinery and offer reports based on a variety of data (including visual, infra-red, and vibration data) can save time and money on each production cycle.

Automotive Industry

RTLS is used to speed up production cycles in vehicle manufacturing by tracking the location of cars and wireless tools. Tool operations can be automated, and tools can be calibrated based on their proximity to certain vehicles. As more and more customizable options are being offered by car manufacturers, each car in a production line often needs new tool settings. By replacing traditional barcode scanners with UWB tags, manufacturers are now able to speed up their production lines, meeting targets quicker.

The tagged vehicles are then tracked as they move into storage and can even retain tags to help with maintenance and repair tasks in future once the vehicle has been sold. Tesla currently keeps a digital twin of all their vehicles to ensure the vehicle history remains clear across its lifetime.

Yard Management

Manufacturing and logistics usually interface in a yard environment and ensuring that a facility's products are delivered in a coordinated and timely fashion is very much in the wheelhouse of RTLS systems. By ensuring transport vehicles are at the right loading bays at the right time, and the correct inventory, pallets, and containers are loaded or stored, many logistical problems can be diminished. Since the global pandemic, more manufacturing sites are building up reserves of stock to help better weather any future supply chain disruptions. This has led to a greater burden on storage and logistics management, which RTLS has helped alleviate.

3.9.4. A possible scenario

To illustrate, consider a forklift in a factory hall divided into several sections. In some sections, we only need to know the forklift's rough location (BLE). In others, where certain zones are restricted to forklifts, UWB shall be activated to send an alert if the forklift approaches such a zone. In another zone with strict speed limits, we shall activate UWB and MEMS to monitor the forklift's speed. This seamless handling of complex localization/tracking scenarios is made possible by our single type of tag, making our system optimal for various industrial applications.

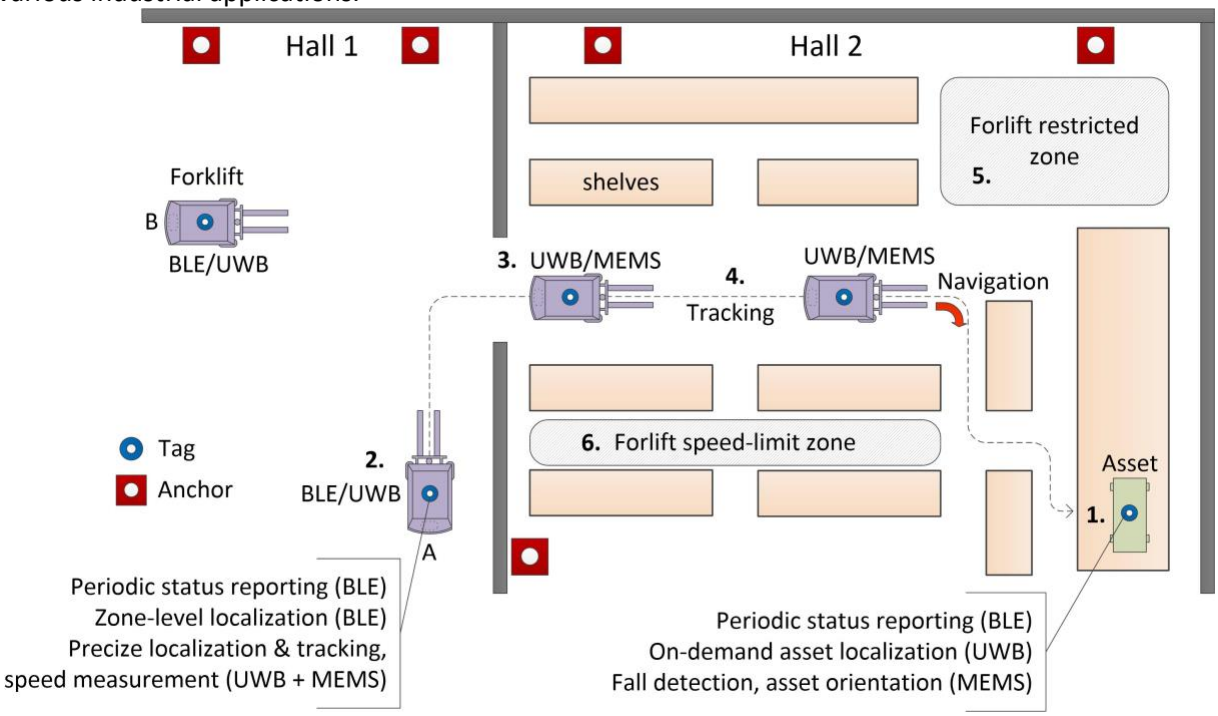

Figure 19 – Edge computing infrastructure

Picture description: In the provided image, we visualize a complex, real-world use-case showcasing the versatility of our tags and system. The following key points correspond to the numbered locations on the image:

1. The Asset: Equipped with our versatile tag, the asset is located in Hall 2. Our system can identify its approximate location within the hall using BLE technology, which consumes minimal power.

2. Forklift Selection (2. on the picture): When the asset needs to be relocated, the system seeks out the closest forklift to Hall 2. The forklift, equipped with the same type of tag, is identified (Forklift A), and selected for this task.
3. Forklift A Enters Hall 2 (3. on the picture): Once the selected forklift A enters Hall 2, the system switches to UWB technology to determine the precise location of the asset. This enables the tracking and navigation functions (4. on the picture) for the forklift to guide it to the exact location of the asset within the hall.
4. Restricted Zones (5. on the picture): With the precise real-time location of the forklift known, the system can implement restricted zone functions. If the forklift approaches a designated no-entry area, an alert will be triggered, preventing potential mishaps, and enhancing safety.
5. Speed Monitoring: In areas with strict speed limits (6. on the picture), the system activates UWB and MEMS technologies. These technologies measure the forklift's speed, ensuring adherence to the set speed regulations and further improving safety within the facility.

Optional - Control Room: Overlooking the entire operation is the Control Room, housing the server for system monitoring and management. This is where all data from the tags converge, allowing for real-time tracking, navigation, and enforcement of safety measures. It is here that operators can monitor the entire operation, implement changes, and manage potential issues promptly.
This comprehensive scenario encapsulates the utility and adaptability of our unified tag, underscoring its potential in managing complex localization and tracking situations within various industrial applications. In conclusion, our hybrid system is not just a standard RTLS with fixed functionality but a flexible platform that allows users to adapt it to their specific needs

3.10. Modern Data Platforms

In the past, data platforms were viewed as a means to an end versus the final product built by your data team. Now, companies are taking a page from software engineering teams and beginning to treat data platforms like production-grade software (data-as-a-service), dedicating valuable team resources to maintaining and optimizing them.

A data platform is a central repository and processing house for all of an organization's data. A data platform handles the collection, cleansing, transformation, and application of data to generate business insights.

Data-first companies have embraced data platforms as an effective way to aggregate, operationalize, and democratize data at scale across the organization. As more organizations begin implementing data mesh architecture[1], and principles like data products, data teams have also to consider the data platform as the machine that develops, manages, surfaces, and governs data products.

[1] https://www.montecarlodata.com/blog-what-is-a-data-mesh-and-how-not-to-mesh-it-up/

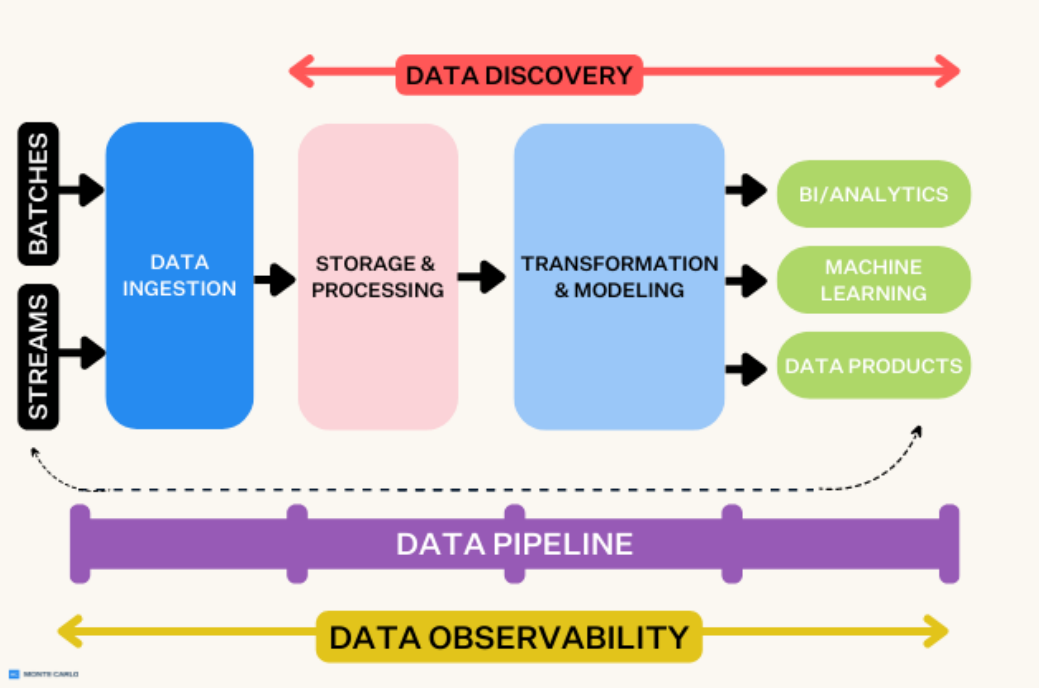

Figure 20 – The position of Data Observation

A data platform, often referred to as a 'modern data stack,' is the central processing hub for an organization's data ecosystem. The data platform manages the collection, normalization, transformation, and application of data for a given data product—from business insights and dashboards to ML and AI engineering.

By integrating tools from a variety of vendors, a data platform enables a data engineering team to not only *manage* an organization's data but also *activate* it for a domain's use cases.

A comprehensive data platform solution powers data acquisition, storage, preparation, delivery, governance, and even the robust security needs of users and applications.

In today's data-driven landscape, building a data platform is no longer a nice-to-have, but a necessity for most organizations. In fact, many companies will even *differentiate themselves from competitors* based on their ability to glean actionable insights from their data—whether that's to improve their customer experience, increase revenue, define their brand, or even create their own large language models for advanced AI applications.

And just like the companies they power, no two data platforms are ever exactly alike. From data sources and use cases to industry and company size, how a company structures its data platform will depend on a variety of critical factors. Even with a revolving door of new data platform tooling—in an increasingly business critical field no less—there are still a few must-haves to get you pointed in the right direction.

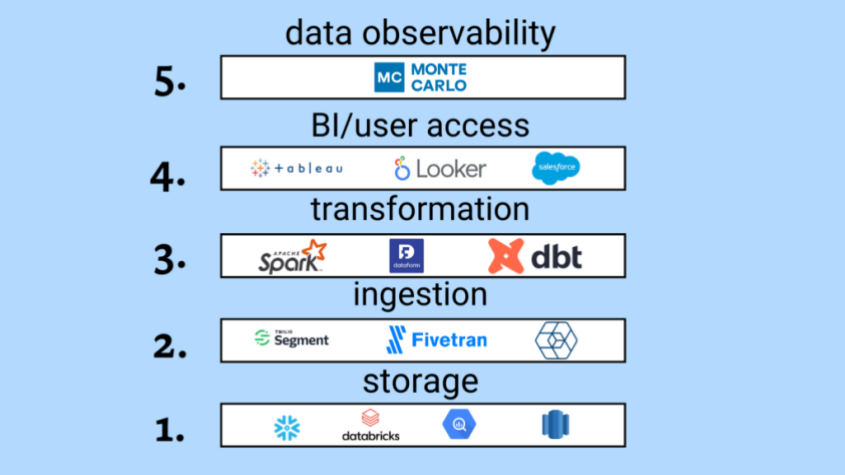

Figure 21 – The position of Data Observation

3.10.1. Data Storage and Processing

The role of Data storage and processing layer is to enable to store the data and process it before it is later transformed and sent off for analysis. It becomes especially important to have a data storage and processing layer when starting to deal with large amounts of data and holding that data for a long period of time and need it to be readily available for analysis.

With companies moving their data platforms to the cloud, the emergence of cloud-native solutions (data warehouse vs data lake or even a data lakehouse[1]) have taken over the market, offering more accessible and affordable options for storing data relative to many on-premises solutions.

Whether you choose to go with a data warehouse, data lake or some combination of both is entirely up to the needs of your business.

Recently, there's been a lot of discussion around whether to go with open source or closed source solutions (the dialogue between Snowflake[2] and Databricks[3]' marketing teams really brings this to light) when it comes to building the data platform.

Below, we highlight some leading options in today's cloud warehouse, lake, landscape:

Snowflake – The original cloud data warehouse, Snowflake provides a flexible payment structure for data teams, as users pay separate fees for computing and storing data.

- Google BigQuery[4] – Google's cloud warehouse, BigQuery, provides a serverless architecture that allows for quick querying due to parallel processing, as well as separate storage and compare for scalable processing and memory.
- Amazon Redshift[5] – Amazon Redshift, one of the most widely used options, sits on top of Amazon Web Services (AWS) and easily integrates with other data tools in the space.
- Firebolt[6] – A SQL-based cloud data warehouse that claims its performance is up to 182 times faster than other options, as the warehouse handles data in a lighter way thanks to new techniques for compression and data parsing.

[1] https://databricks.com/blog/2020/01/30/what-is-a-data-lakehouse.html
[2] https://www.snowflake.com/blog/choosing-open-wisely/
[3] https://www.youtube.com/watch?v=dNjy_B7A414
[4] https://cloud.google.com/bigquery
[5] https://aws.amazon.com/redshift/?whats-new-cards.sort-by=item.additionalFields.postDateTime&whats-new-cards.sort-order=desc
[6] https://www.firebolt.io/home-v2

- Microsoft Azure[1] – Microsoft's cloud computing entrant in this list common among teams that leverage heavy Windows integrations.
- Amazon S3[2] – An object storage service for structured and unstructured data, S3 gives you the compute resources to build a data lake from scratch.
- Databricks[3] – Databricks, the Apache Spark-as-a-service platform, has pioneered the data lakehouse, giving users the options to leverage both structured and unstructured data and offers the low-cost storage features of a data lake.
- Dremio[4] – Dremio's data lake engine provides analysts, data scientists, and data engineers with an integrated, self-service interface for data lakes.

3.10.2. Data Ingestion

As is the case for nearly any modern data platform, there will be a need to ingest data from one system to another.

As data infrastructures become increasingly complex, data teams are left with the challenging task of ingesting structured and unstructured data from a wide variety of sources. This is often referred to as the extraction and loading stage of Extract Transform Load (ETL) and Extract Load Transform (ELT).

While you may have heard the term "zero-ETL"[5] in reference to integrating data from one location to another without the need for data pipelines, the widespread adoption of this practice is still quite a ways into the future.

Below, we outline some popular tools in the space broken into two groups. **Batch**, which ingests data from a source at a pre-defined time or in pre-defined groupings, is the most popular way for data engineers to gather data. **Data streaming** is becoming more popular, especially for operational use cases where the value of (near) real-time data is worth the added complexity.

Batch data ingestion solutions include:

- Fivetran[6] – A leading enterprise ETL solution that manages data delivery from the data source to the destination.
- Singer[7] – An open source tool for moving data from a source to a destination.
- Stitch – A cloud-based open source platform that allows you to rapidly move data from a source to a destination.
- Airbyte[8] – An open source platform that easily allows you to sync data from applications.

Data streaming ingestion solutions include:

[1] https://www.google.com/search?q=microsoft+azure&rlz=1C5CHFA_enUS905US905&oq=microsoft+azure&aqs=chrome..69i57j0i433j0j0i433j0j0i433j0l2j0i433i457j0.4251j1j4&sourceid=chrome&ie=UTF-8#:~:text=Microsoft%20Azure%3A%20Cloud,us)%3A%20389/55.49K

[2] https://aws.amazon.com/s3/

[3] https://databricks.com/

[4] https://www.dremio.com/

[5] https://aws.amazon.com/about-aws/whats-new/2022/11/amazon-aurora-zero-etl-integration-redshift/

[6] https://www.fivetran.com/

[7] https://www.singer.io/

[8] https://airbyte.com/

- Apache Kafka[1] – Confluent[2] is the vendor that supports Kafka, the open source event streaming platform to handle streaming analytics and data ingestion. They also recently acquired Apache Flink[3], another streaming solution.
- Amazon Kinesis[4]– A streaming solution from AWS that may be ideal for those utilizing Redshift as their data warehouse.
- Google Pub/Sub[5]– A GCP service that enables you to ingest streaming data into BigQuery, data lakes, or operational data bases. Google made waves last year when they announced the ability to stream Pub/Sub directly into BigQuery without pipelines.
- Apache Spark[6]– Labeled as a unified analytics engine for large scale data processing, many leverage this open source solution for streaming use cases, often in conjunction with Databricks[7].

Even with the prevalence of ingestion tools available on today's market, some data teams choose to build custom code to ingest data from internal and external sources, and many organizations even build their own custom frameworks to handle this task.

Orchestration and workflow automation, featuring such tools as Apache Airflow, Prefect, and Dagster, often folds into the ingestion layer, too. Orchestration takes ingestion a step further by taking siloed data, combining it with other sources, and making it available for analysis.

I would argue, though, orchestration can be (and should be) weaved into the data platform after you handle the storage, processing, and business intelligence layers. You can't orchestrate without an orchestra of queryable data, after all!

3.10.3. Data Transformation and Modeling

Data transformation and modeling are often used interchangeably, but they are two very different processes.

When you transform your data, you are taking raw data and cleaning it up with business logic to get the data ready for analysis and reporting. When you model data, you are creating a visual representation of data for storage in a data warehouse.

Below, we share a list of common data transformation and modeling tools that data engineers rely on:

- dbt[8] – Short for data build tool, is the open source leader for transforming data once it's loaded into your warehouse.
- Dataform[9] – It allows you to transform raw data from your warehouse into something usable by BI and analytics tools.
- Sequel Server Integration Services (SSIS) – Hosted by Microsoft, SSIS allows your business to extract and then transform that data from a wide variety of sources which you can then later use to load into your destination of choice.
- Custom Python code and Apache Airflow – Before the rise of tools like dbt and Dataform, data engineers commonly wrote their transformations in pure Python. While it might be tempting to

[1] https://kafka.apache.org/
[2] https://www.confluent.io/
[3] https://flink.apache.org/
[4] https://aws.amazon.com/kinesis/
[5] https://cloud.google.com/blog/products/data-analytics/pub-sub-launches-direct-path-to-bigquery-for-streaming-analytics
[6] https://spark.apache.org/
[7] https://www.databricks.com/
[8] https://www.getdbt.com/
[9] https://dataform.co/

continue using custom code to transform your data, it does increase the chances of errors being made as the code is not easily replicable and must be rewritten every time a process takes place.

The data transformation and modeling layer turns data into something a little more useful, readying it for the next stage in its journey: analytics.

3.10.4. Business Intelligence (BI) and Analytics

The data you have collected, transformed, and stored within your data platform serves your business no good if your employees can't use it.

If the data platform was a book, the BI and analytics layer would be the cover, replete with an engaging title, visuals, and summary of what the data is actually trying to tell you. In fact, this layer is often what end-users think of when they picture a data platform, and for good reason: it makes data actionable and intelligent, and without it, your data lacks meaning.

Below, we outline some popular BI solutions among top data teams:

Looker[1] – A BI platform that is optimized for big data and allows members of your team to easily collaborate on building reports and dashboards.

- Sigma Computing – A BI platform that delivers cloud-scale analytics with the simplicity of a spreadsheet and familiar data visualizations.
- Tableau[2] – Often referred to as a leader in the BI industry, it has an easy-to-use interface.
- Mode[3] – A collaborative data science platform that incorporates SQL, R, Python, and visual analytics in one single UI.
- Power BI[4] – A Microsoft-based tool that easily integrates with Excel and provides self-service analytics for everyone on your team.

[1] https://dataform.co/

[2] https://www.google.com/aclk?sa=l&ai=DChcSEwjVh760gqzxAhUGrsgKHfO8DYYYABAAGgJxdQ&sig=AOD64_2nK_YvIHT962la-NW-lgEySYsO2g&q&nis=1&adurl&ved=2ahUKEwjW6bS0gqzxAhXumuAKHdxZBI4Q0Qx6BAgCEAE

[3] https://mode.com/

[4] https://powerbi.microsoft.com/en-us/

4. AI REDGIO 5.0 Reference Architecture

4.1. Architecture Requirements and Specifications

In this section requirements and specifications coming from the AI REDGIO 5.0 experiments, Artificial Intelligence challenges and AI REGIO Techno Assets will be analyzed. In particular, inputs from experiments will be listed, in synergy with the results of WP2 activities, describing the meaning and the context.

4.1.1. Inputs from Experiments

User requirement identifiers for Industry 5.0 Data4AI Platform & Data Spaces

This subsection focuses on the user requirements of the 7 SME driven experiments and 14 Didactic Factories from the Industry 5.0 Data4AI Platform and Data Spaces (WP4). The users were prompted for input what requirements were needed for their experiment to progress. The requirements were collected and are presented here in a generalised form to allow open solutions to be developed.

In the following table, we present the requirements from 7 SME driven experiments.

Pilot	Organization name	I5.0 Data4AI & Data Spaces (WP4)		
		Secure Edge-to-Cloud continuum	Data4AI Platform & Data Quality	Data Spaces & Interoperability
I	SCAMM	Confidentiality of the IIOT data produced between edge and cloud	Missing values in input data can't be filled by synthetic data, but have to be addressed somehow	Sharing data with supplier means data interoperability (different formats standards)
II	PERNOUD	No Current plans to send data out of Enterprise in Phase 1	Data preprocessing and cleaning activities are expected	N/A
III	GPALMEC	At this stage of the project we do not plan to send data outside the machine. We might be interested in case of a further develop of the project.		
IV	POLYCOM	Edge to local cloud connectivity is needed that provides secure and reliable communication channel. The data security is not critical but data privacy should be provide especially if local cloud is not in the same network.	To improve the robustness of the data processing pipelines data validity needs to be verified before analytics in order to verify whether the ML algorithms are valid for the considered data set (detection of model extrapolation, detection of missing/defect data)	N/A

Pilot	Organization name	I5.0 Data4AI & Data Spaces (WP4)		
		Secure Edge-to-Cloud continuum	Data4AI Platform & Data Quality	Data Spaces & Interoperability
V	QUESCREM	No current plans to transfer data outside the enterprise's facilites, at the moment	Data preprocessing and cleaning activities are expected	N/A
VI	CAP	No Current plans to send data out of Enterprise in Phase 1	Industreweb.cloud platform will host Data Management Tools	Open to discuss opportunities
VII	KATTY FASHION	Confidentiality of the 3D models and patterns used as input for the QA Analysis	QA analysis results needs to be complex enough to be validated by a junior QA team member	The 3D models and patterns used as input for the QA Analysis are confidential to each customer/supplier and cannot be shared between them. But each customer/supplier data is processed by product development team to obtain a standardized model of data needed for production, which is close to industry standards.

In the following table, we present the requirements from 14 Didactic Factories.

DF	Organization name	I5.0 Data4AI & Data Spaces (WP4)		
		Secure Edge-to-Cloud continuum	Data4AI Platform & Data Quality	Data Spaces & Interoperability
I	POLIMI - I4.0Lab	…	…	…
II	UNIBO - E2MECH	Privacy preservation (data should not be public)	N/A	N/A
III	JSI - IJS Systems&Control Lab	As the edge and local cloud may not be on the same local network, secure communication channel should be used	Basic detection of data validity (basic detection of missing or faulty data)	N/A
IV	Brainpoint Industries-Flexible Manufacturing	…	…	…
V	UniTwente - AMC	Data processed within AMC or SME	N/A	N/A
VI	FBK - 4.0iLab	Different edge processing for evaluating the IIoT application	Data collected need to be process (verify, correct, clean) at the Edge	N/A
VII	MAKE -PM50	N/A - open to discussion	N/A - open to discussion	N/A - open to discussion
VIII	DMIW -Digital Manufacturing Innovation Hub	All data will be processed at the edge using the Industreweb platform provided by C2K	The AI interface will be built within the Industreweb platform using open source models	N/A
IX	MADE - BEhAI	MAYBE (Biometrics data needs to be protected)	Redundant data/potentially not useful data	N/A
X	TUIASI - I4.0	Most of the processes will be done locally on a SBC (acquire image, acces the database, detect if there is a defect or not, send an alert)	In the end will be a percentage if there is a defect or not, according to a model (reference)	it will be use a AI at edge platform with more or less conections to a specific data platform
XI	CTU - RICAIP	Fast and simple anomaly detection at the edge and more time consuming validation or retraining of the detectors in the cloud	Potential concept drift must be checked, recognized and learned to prevent false positive detections.	N/A at this moment, in case of future deployment in multiple sites and multiple processes, different methods can work differently on different problem instances and

DF	Organization name	I5.0 Data4AI & Data Spaces (WP4)		
		Secure Edge-to-Cloud continuum	**Data4AI Platform & Data Quality**	**Data Spaces & Interoperability**
				evaluation could need to share and transfer data or their properties
XII	AAU - Smart Lab	The solutions work locally only.	The data will be collected, cleaned and visulized for later processing.	The collected data will be saved withing the SME factory on the local server/platform.
XIII	PBN - amLab	NA	The data will be collected, cleaned for later processing.	The data will be stored on local server
XIV	GRADIANT - Galicia Industrial Logistics Lab	No current plans to transfer data outside the DF's facilites, at the moment	Data preprocessing and cleaning activities are expected	N/A

The identifier contains information about the requirement. It is based on the following pattern:
R is requirement
Data = Industry 5.0 Data4AI Platform & Data Spaces
No = Number of Requirement

Table 1: User requirements for Industry 5.0 Data4AI Platform & Data Spaces

Identifier	User requirement	Description
R.Data.01	Data security	The platform must provide a secure and reliable communication channel between edge to local or external cloud.
R.Data.02	Data storage	The platform must provide secure data storage.
R.Data.03	Data Validity	Support in identification of valid data. Data must be relevant and representative for intended use.
R.Data.04	Data privacy	Data must be only accessible by intended user.
R.Data.05	Data interoperability	Platform must be able to ensure data interoperability for data exchange.
R.Data.06	Data interpretation	Provision of data interpretation support to collect information and draw conclusion using different analytical approaches.
R.Data.07	Data Cleaning & Accuracy	Provision of data editing services to monitor data inconsistencies, identify incorrect data and fill in data gaps.

R.Data.08	Data Integrity	Provision of data accuracy testing to avoid mismatching among others.
R.Data.09	Data Processing	Provision of data processing at the Edge.
R.Data.10	Drift detection	Concept drift invalidates the data model. The platform must provide measures to detect concept drift.
R.Data.11	Detection of model extrapolation	The platform must provide measure to detect unwanted model extrapolation to ensure data accuracy.
R.Data.12	Quality Assurance process	Quality Assurance results must be complex enough to be checked by a junior Quality Assurance specialist.
R.Data.13	Platform Compatibility	Platform must be able to communicate with external data platforms (e.g. Industreweb).

User requirements for Industry 5.0 EDGE AI Toolkit & AI-On-demand Platform

This subsection focuses on the user requirements of the 7 SME driven experiments and 14 Didactic Factories from the Industry 5.0 EDGE AI Toolkit & AI-On-demand Platform (WP5). The users were prompted for input what requirements were needed for their experiment to progress. The requirements were collected and are presented here in a generalized form to allow open solutions to be developed

In the following table, we present the requirements from 7 SME driven experiments.

Pilot	Organization name	I5.0 EdgeAI Toolkit & AIoD Platforms (WP5)		
		Open hardware at the Edge	AI Pipelines & Orchestration	Human-AI Collaboration
I	SCAMM	Additional hw must be installed for collecting data from the production line: Process parameters from the PLC (they are not stored anywhere at the moment) Images of the defects have to be processed at the edge to extract synthetic indexes about product quality	The AI models should be orchestrated in a pipeline. Deep Learning model to detect and evaluate products defects NLP model to process operators input about detected issues and applied solutions (Company knoweledge management) Machine Learning model to correlate process parameters with quality indexes generated at the edge by DL model	Operators need to interface with the AI system whenever the AI system detects a defect, operators are informed through a user interface. Operators verify the issue and apply corrective actions to restart production. Description of the issue + applied solution has to be input in the knoweledge management system through the available user interface It is also possible to ask the operator to assess whether the AI system produced a false alarm through the available user interface

Pilot	Organization name	I5.0 EdgeAI Toolkit & AIoD Platforms (WP5)		
		Open hardware at the Edge	AI Pipelines & Orchestration	Human-AI Collaboration
II	PERNOUD	Hardware must be added to collect information directly from the machine PLC	TBD	Human-AI collaboration is the key collaboration for the experiment success. Human will receive information and prescription form AI and decide to use it as it is or to modify it according to expert vision
III	GPALMEC	Edge hardware is needed process IoT and sensor data for localization	TBD	Human-AI collaboration already guaranteed: driver receives the information from the AI logic and decides the next movement. Driver can always override the AI decision.
IV	POLYCOM	Edge hardware is needed to process the data locally, close to the data sources, the analytic's results are transfered to the central system (local cloud)	Central management of edge deployed algorithms that would provide a possibility to monitor the edge deployed algorihtms (status, performance) and possibility to deploy new pipelines from central system	Operators are monitoring the outputs of the AI solutions and annotate the special cases that were falsly detected as anomalies or not detected by the algorithms. This will represent an input for continous model imrpovement.
V	QUESCREM	AI models are expected to be run on a generic edge environment	The AI pipeline needed is very simple: it will involve the collection of real-time production data and the forecasting of new KPIs predictions (at the edge) and the prescription of the optimal production parameters (at the cloud).	Operators will label the results provided by the prediction module that will be developed. Also operators will monitor the predictions and prescriptions provided by the system.
VI	CAP	Industreweb (TM)	Industreweb Data Platform	Dashboards that are context sensitive
VII	KATTY FASHION	Edge hardware is needed to be able to shorten the QA check duration. AI models are expected to be run on a generic edge environment.	Each product will be analyzed by a pipeline of AI model based on provided model data. A result of this analysis is expected before moving to the next product.	Operators need to interface with the AI system to indicate the defect type and location. Would be usefull to mark the defect as valid, invalid, fixed, not fixable.

In the following table, we present the requirements from 14 Didactic Factories.

DF	Organization name	I5.0 EdgeAI Toolkit & AIoD Platforms (WP5)		
		Open hardware at the Edge	AI Pipelines & Orchestration	Human-AI Collaboration
I	POLIMI - I4.0Lab	…	…	…
II	UNIBO - E2MECH	…	…	…
III	JSI - IJS Systems&Control Lab	Time series will be processed close the data source (one process cell) and only aggregated results will be passed forward for monitoring and logging in local cloud	Deployment of new algorithms from central point could be useful to enable scalability of the system	Operators to support reasoning of detected segments and anomalies. Validation of the deployed algorithms and annotation of special cases.
IV	Brainpoint Industries-Flexible Manufacturing	…	…	..
V	UniTwente - AMC	Raspberry Pi is applied for proof of concept	N/A	N/A
VI	FBK - 4.0iLab	Different Hardware platform to test different Edge Scenarios	Basic ML Pipelines according to each IIoT application	N/A
VII	MAKE -PM50	N/A - open to discussion	N/A - open to discussion	N/A
VIII	DMIW -Digital Manufacturing Innovation Hub	All data will be processed at the edge using the Industreweb platform provided by C2K	The AI interface will be built within the Industreweb platform using open source models	Operators will confirm actions suggested by the AI to improve the manufacturing process, the AI will also be used to train human operators.
IX	MADE - BEhAI	N/A - Open to discussion	Central management of edge-deployed algorithms / three pipelines dedicated to the analysis of the three main streams of data - Open to discussion	Adaptation of AI to the Human - monodirection - Open to discussion
X	TUIASI - I4.0	SBC computer in order to run a few tasks (acquire image, process image, run AI fuzzy program, sens un alarm)	we are using fuzzy AI	QA personnel from KF will use a simple and compact HMI in order to send amd receive data
XI	CTU - RICAIP	…	…	Human confirms the AI decision and thus provides ground truth

DF	Organization name	I5.0 EdgeAI Toolkit & AIoD Platforms (WP5)		
		Open hardware at the Edge	**AI Pipelines & Orchestration**	**Human-AI Collaboration**
XII	AAU - Smart Lab	The solution currently works locally. Edge devices could be possible to be used for deploying the data analysis/ML model.	The AI pipeline used in our case include: collecting real-time data, visulizing the data, reformat the data, modeling building/testing, and deployment on shopfloor.	Human operators will be able to monitoring the production data through the user interface. This can be used for assisting the operator to make adjustments and future planning.
XIII	PBN - amLab	to implement the AI device later on	Our AI pipeline and orchestration encompass data collection, processing, model development, real-time decision-making, deployment, and ongoing maintenance. These steps are designed to optimize energy usage, enhance recycling operations, and promote sustainability in the context of our experiment.	N/A
XIV	GRADIANT - Galicia Industrial Logistics Lab	Edge devices might be needed to complement the DF industrial lines in order to run the AI models at the Edge	The AI pipeline needed is very simple: it will involve the collection of real-time production data and the forecasting of new KPIs predictions (at the edge) and the prescription of the optimal production parameters (at the cloud).	Operators will label the results provided by the prediction module that will be developed. Also operators will monitor the predictions and prescriptions provided by the system.

The identifier contains information about the requirement. It is based on the following pattern:

R is requirement
AI = Industry 5.0 EDGE AI Toolkit & AI-On-demand Platform
No = Number of Requirement

Table 2: User requirements for Industry 5.0 EDGE AI Toolkit & AI-On-demand Platform

Identifier	User requirement	Description
R.AI.01	Edge Hardware	Provision of hardware at the edge for local data processing.
R.AI.02	Edge Environment for AI models	Provision of generic edge environment to run AI models.
R.AI.03	Imagine Analysis Processing	Ability to execute imaging analytics at the edge.
R.AI.04	Edge Hardware Testbed	Provision of different hardware platforms to test different Edge scenarios.
R.AI.05	AI-User Interface	Creation of a user interface for interaction with AI tool.
R.AI.06	AI management	Management System support for AI algorithms e.g., central management of edge deployed algorithms, possibility to deploy new algorithms from central system.
R.AI.07	AI monitoring	Provision of a platform to monitor status and performance of edge deployed algorithms.
R.AI.08	Human-AI interaction	Provision of feedback loop for human to monitor (correct, monitor, modify) AI in different applications, using a context sensitive dashboard.
R.AI.09	Human-AI feedback	Provision of possibility to correct, modify or override AI input after human supervision.
R.AI.10	AI training	Provision of improved Ai algorithms by incorporation of human feedback loops for further AI modification and training.
R.AI.11	Knowledge management system	Preparation of a knowledge management system to access information about issues and applied solution.
R.AI.12	AI Quality Control Dash Board	Creation of a context sensitive dashboard for quick AI output assessment.
R.AI.13	AI Pipeline validation	Provision of interface for validation of the deployed algorithms and annotation if necessary. Human monitors the output of AI algorithm and annotates cases of e.g. falsely detected anomalies or missed anomalies.

4.2. Design Principles for AI REDGIO 5.0 RA

The AI REDGIO 5.0 Reference Architecture will follow, like happen for the most of IIoT system implementations, the well-established architectural patterns described in relation with the benefits that can bring to the AI REDGIO 5.0 project:

- **Three-tier architecture:** It is an easy representation of an Industrial Internet solution, but simultaneously describes the main areas to be considered when design an Industrial Internet System. Three-tier architecture is composed by:

- **Edge tier:** data from all the end nodes is collected, aggregated, and transmitted over the proximity network to a border gateway. Depending on the protocols and technologies used within the edge tier, some data translation and interface integration may be applied at hubs, remote I/O devices, or protocol convertors. The edge tier contains the functions for the control domain.

- **Platform tier:** receives data from the edge tier over the access network and is responsible for data transformation and processing. The platform tier is also responsible for forward control commands from the enterprise tier to the edge tier. It is within the platform tier where it will be located the majority of the functions related to the information and operations domains.

- **Enterprise tier:** implements domain-specific applications, applications and business logic for decision support systems and end-user interfaces to end-users including operation specialists. The enterprise tier receives data flows from the edge and platform tier. It also issues control commands to the platform tier and edge tier.

  They are connected by three networks (Proximity, Access and Service networks).

- **Layered Databus:** The Layered Databus architecture pattern is often applied to IIoT systems, as it provides low-latency what minimizes delays between the transmission and receipt of data, secure and peer-to-peer data communications across logical layers of the system. At the lowest level, the **Machine Databus** is used by smart objects such as: sensors, objects, devices, machines and products for local control, automation and real-time analytics. The Site and Unit layers could also be coincident and use another Databus for supervisory control and monitoring.

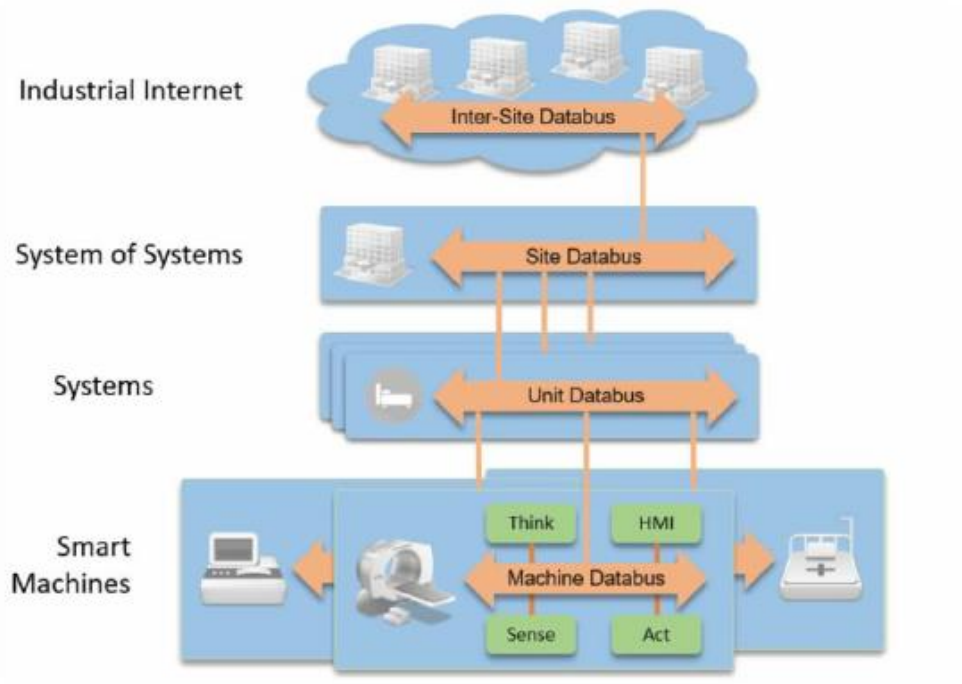

Figure 22 - Layered Databus Architecture

A special focus must be done on how manage the data-in-motion and the data-at-rest to have a data-driven architectural approach, where:

- **Data in Motion:** platforms are software subsystems that perform real-time calculations on event data in motion. They retain a relatively small working set of stream data in memory for a limited time window, typically seconds to hours.
- **Data at Rest:** data, structured or unstructured, that has been flushed out from the memory and stored in any digital form (e.g. cloud storage, file hosting services, databases, data warehouses, …).

4.3. AI REDGIO 5.0 Reference Architecture

Reference architecture is presented in the following figure.

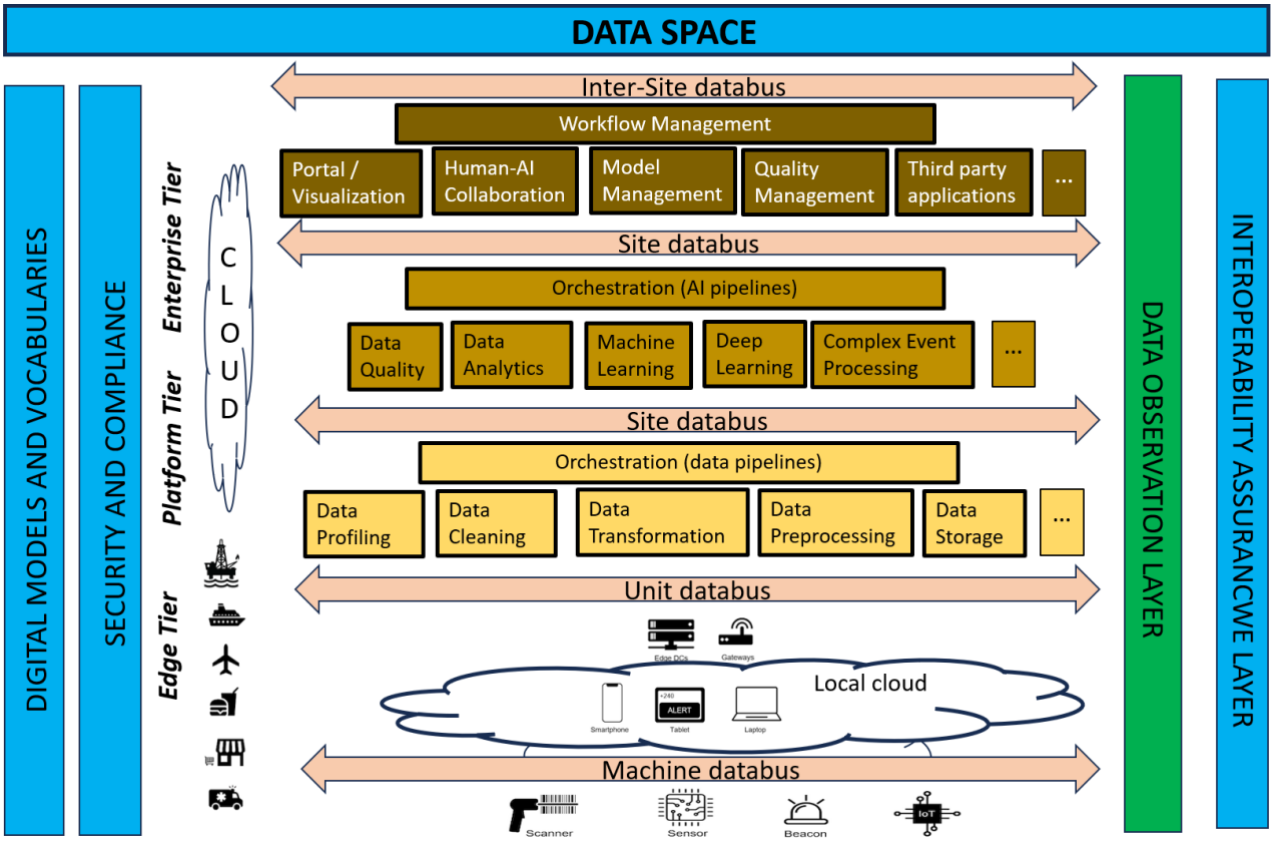

Figure 23 - AI REDGIO 5.0 Reference Architecture

- **Edge**: ecosystem of heterogeneous devices (IoT Sensors, Machine Tools, IT Systems, Robots, Cameras and so on) to gather machine data and make it available to the upper layers and IoT systems.
- **Edge Tier**: computes some data management and analysis functions, in small datasets, using data, applications, and services contained in the edge.

- **Data profiling**: the service used for making the information available within its environment and adapt its execution accordingly.
- **Data cleaning**: the process able to guarantee the correctness of a huge amount of data, starting from data mining.
- **Data preprocessing**: in charge of data manipulation to prepare it in the format needed to be analyzed.
- **Data storage**: local storage, which enables advanced processing.
- **Orchestration**: implements the logic for defining complex data flow processes between devices, data services, applications and people to produce desired outcomes. It is the link to the upper layer, allowing the data pipeline to communicate with the Platform services
-

- **Platform Tier**: leverages the processing of data enables making smarter, and sometimes faster, business decisions.
  - **Data quality**: the process to discover and repair corrupted or inaccurate data, ensuring the quality of information in the dataset.
  - **Machine Learning**: the services used to create models from data.
  - **Data analytics**: set of data analytics techniques and services used to take advantage from the information extracted/provided by the subsystems.
  - **Deep Learning**: using ANN for learning from data
  - **Complex Event Processing:** Complex pattern matching on real-time streaming data
  - **Orchestration services**: services used to deploy the solution along the supply chain once it is designed thanks to the AI REDGIO 5.0 platform. It interacts with the lower layer to gather data and storage it in order to make it available for the services in the platform.

- **Enterprise Tier**: receives the data and integrates it with data from other systems, to perform analysis across business silos, carrying out industry domain-specific business applications, related decision support and business intelligence systems. This tier provides interfaces to human consumers.
  - **Portal/Visualization**: enables access to data in an interactive graphical way.
  - **Human-AI Collaborative Intelligence**: referred to Industry 5.0, is the service offered to improve the interaction human-machine with the help of AI.
  - **Quality Management**: services for ensuring the quality of the product and processes. An example is anomaly detection
  - **Model Management**: services for managing the drift of the models
  - **Orchestration**: is the service used to design the solution along the supply chain.
  - **Third Party Applications**: any other kind of application can integrate AI REDGIO 5.0 Platform.

- **Digital Models and Vocabularies**: the set of models and vocabularies used in the among all the layers allowing the interoperability.

- **Interoperability Assurance Level**: ensures interoperability among all the levels finding a link to connect them, allowing the exchange of information.

- **Security and compliance Level**: service offered to all the layer act to ensure a trusted environment at all the levels.

- **Data Observation:** set of services which process metadata related to the execution of any of the services in any of the layers

Data Observability

With data pipelines becoming increasingly complex and organizations relying on data to drive decision-making, the need for this data being ingested, stored, processed, analyzed, and transformed to be trustworthy and reliable has never been higher. Simply put, organizations can no longer afford data downtime i.e., partial, inaccurate, missing, or erroneous data. Data observability is an organization's ability to fully understand the health of the data in their data ecosystem. It eliminates data downtime by applying best practices learned from DevOps to data pipelines, ensuring that the data is usable and actionable.

Our data observability layer must be able to monitor and alert for the following pillars of observability:

- **Freshness**: is the data recent? When was the last time it was generated? What upstream data is included/omitted?
- **Distribution**: is the data within accepted ranges? Is it properly formatted? Is it complete?
- **Volume**: has all the data arrived?
- **Schema**: what is the schema, and how has it changed? Who has made these changes and for what reasons?
- **Lineage**: for a given data asset, what are the upstream sources and downstream assets which are impacted by it? Who are the people generating this data, and who is relying on it for decision-making?

An effective, proactive data observability solution will connect to your existing data platform quickly and seamlessly, providing end-to-end lineage that allows you to track downstream dependencies.

Additionally, it will automatically monitor your data-at-rest without requiring the extraction of data from your data store. This approach ensures you meet the highest levels of security and compliance requirements and scale to the most demanding data volumes.

4.4. Reference Implementation

Reference implemenation is presented in the following figure.

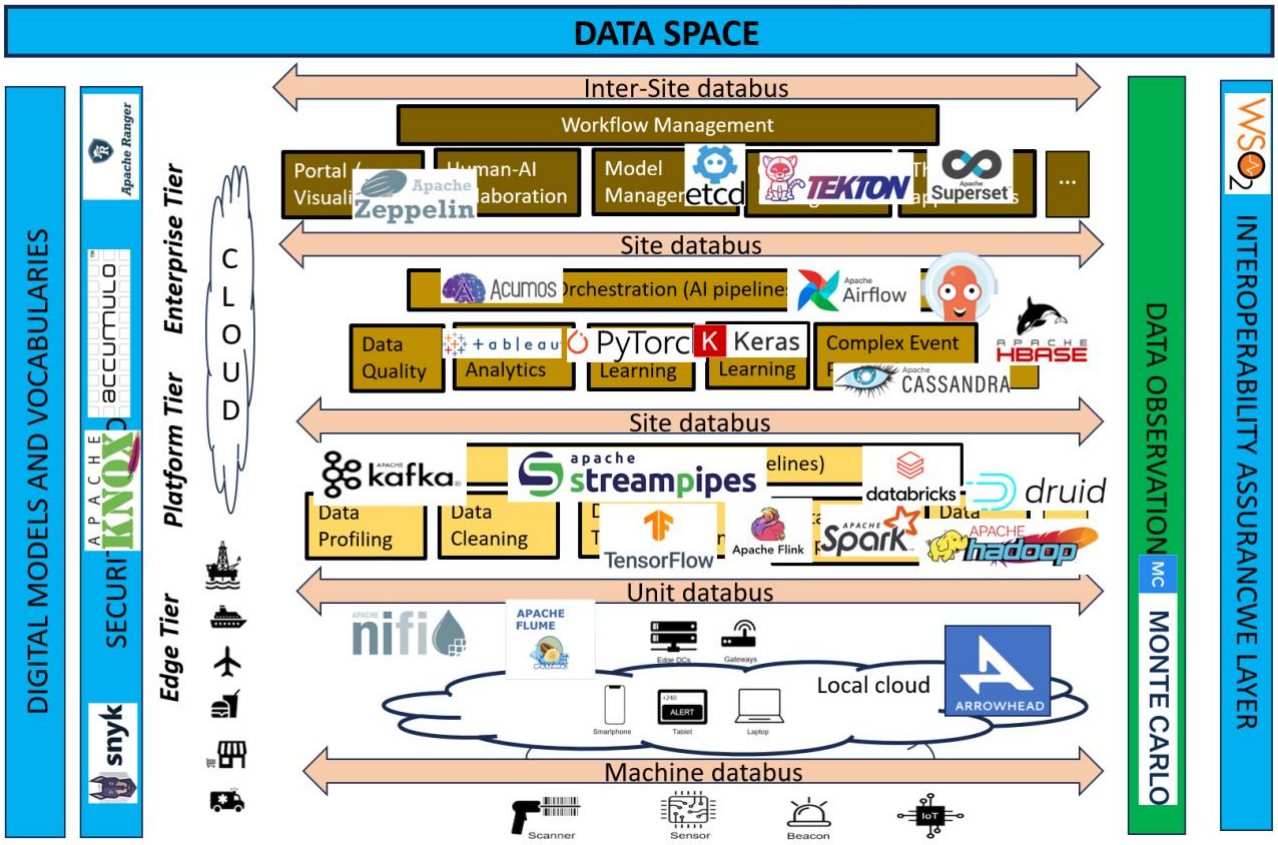

Figure 24 - AI REDGIO 5.0 Reference Implementation

- **Nifi:** Apache NIFI supports powerful and scalable directed graphs of data routing, transformation, and system mediation logic. Some of the high-level capabilities of Apache NiFi include Web-based user interface, Seamless experience between design, control, feedback, and monitoring, data Provenance, SSL, SSH, HTTPS, encrypted content, etc, pluggable role-based authentication/authorization.Apache nifi is highly configurable with loss tolerant vs guaranteed delivery, low latency vs high throughput, dynamic prioritization, flow can be modified at runtime, back pressure.

- **Flume:** A Java-based ingestion tool, Flume is used when input data streams-in faster than it can be consumed. Typically, Flume is used to ingest streaming data into HDFS or Kafka topics, where it can act as a Kafka producer. Multiple Flume agents can also be used collect data from multiple sources into a Flume collector.

- **Kafka:** Kafka is a distributed publish-subscribe messaging system which integrates applications/data streams. It is fast, scalable and reliable messaging system which is the key component in Hadoop technology stack for supporting real-time data analytics or monetization of Internet of Things (IoT) data. Kafka can handle many terabytes of data without incurring much at all. Apache Kafka is altogether different from the traditional messaging system. It is designed as a distributed system and which is very easy to scale out.Kafka is designed to deliver three main advantages over AMQP, JMS etc.

- **Druid:** Apache Druid is a database that is most often used for powering use cases where real-time ingest, fast query performance, and high uptime are important. As such, Druid is commonly used for

powering GUIs of analytical applications, or as a backend for highly concurrent APIs that need fast aggregations. Druid works best with event-oriented data.

- **Streampipes:** StreamPipes is a self-service (Industrial) IoT toolbox to enable non-technical users to connect, analyze and explore IoT data streams. StreamPipes has an exchangeable runtime execution layer and executes pipelines using one of the provided wrappers, e.g., standalone or distributed in Apache Flink. Pipeline elements in StreamPipes can be installed at runtime - the built-in SDK allows to easily implement new pipeline elements according to your needs. Pipeline elements are standalone microservices that can run anywhere - centrally on your server, in a large-scale cluster or close at the edge.

- **Flink:** Flink is a streaming data flow engine which aims to provide facilities for distributed computation over streams of data. Treating batch processes as a special case of data streaming, Flink is effective both as a batch and real-time processing framework, but it puts streaming first. Flink offers several APIs which includes static data API like DataStream API, DataSet API for Java, Scala and Python and SQL-like query API for embedding in Java, Scala static API code. Flink also has its own machine learning library called FlinkML, its own SQL Query called MRQL as well as graph processing libraries.

- **Spark:** Apache Spark is an open-source parallel processing framework for running large-scale data analytics applications across clustered computers. It can handle both batch and real-time analytics and data processing workloads. It's part of a greater set of tools, including Apache Hadoop and other open-source resources for today's analytics community. In this way, it can be considered as a data analytics cluster computing tool. It can be used with the Hadoop Distributed File System (HDFS), which is a particular Hadoop component that facilitates complicated file handling.

- **Hadoop:** The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodity hardware. It has many similarities with existing distributed file systems. However, the differences from other distributed file systems are significant. HDFS is highly fault-tolerant and is designed to be deployed on low-cost hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets.

- **Airflow:** an open-source workflow management platform. Creating Airflow allowed Airbnb to programmatically author and schedule their workflows and monitor them via the built-in Airflow user interface. It is written in Python, and workflows are created via Python scripts. Airflow is designed under the principle of "configuration as code". While other "configuration as code" workflow platforms exist using markup languages like XML, using Python allows developers to import libraries and classes to help them create their workflows.

- **Argo:** Argoproj is a collection of tools for getting work done with Kubernetes like: Argo Workflows (Container-native Workflow Engine), Argo CD (Declarative GitOps Continuous Delivery), Argo Events (Event-based Dependency Manager), Argo Rollouts (Progressive Delivery with support for Canary and Blue Green deployment strategies).

- **Pytorch:** is an open-source machine learning library based on the Torch library(an open-source machine learning library, a scientific computing framework, and a script language based on the Lua programming language. It provides a wide range of algorithms for deep learning, and uses the

scripting language LuaJIT, and an underlying C implementation), used for applications such as computer vision and natural language processing.

- **Keras**: is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System).

- **Tensorflow**: is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. Tensorflow is a symbolic math library based on dataflow and differentiable programming. It is used for both research and production at Google.

- **HBase:** Apache HBase is an open source distributed non-relational database written in Java and used with elements of the Apache software foundation's Hadoop suite of big data analysis tools. It represents one of several database tools for the input and output of large data sets that are crunched by Hadoop and its various utilities and resources. Some of the popular features of Apache HBase include some kinds of backup and failover support, as well as APIs for popular programming languages. Its compatibility with the greater Hadoop system makes it a candidate for many kinds of big data management problems in enterprise IT.

- **Cassandra:** Cassandra is a distributed NoSQL database designed to manage vast amounts of structured data across an array of commodity servers. Cassandra boasts a unique architecture that delivers high distribution, linear scale performance, and is capable of handling large amounts of data while providing continuous availability and uptime to thousands of concurrent users. Cassandra delivers very consistent performance in a fault tolerant environment. This makes Cassandra ideal for processing online workloads of a transactional nature, where Cassandra is handling large numbers of interactions and concurrent traffic with each interaction yielding small amounts of data.

- **Zeppelin:** Apache Zeppelin is an open-source, web-based “notebook” that enables interactive data analytics and collaborative documents. The notebook is integrated with distributed, general-purpose data processing systems such as Apache Spark (large-scale data processing), Apache Flink (stream processing framework), and many others. Apache Zeppelin allows you to make beautiful, data-driven, interactive documents with SQL, Scala, R, or Python right in your browser.

- **Acumos:** Acumos AI is a platform and open-source framework that makes it easy to build, share, and deploy AI apps. Acumos standardizes the infrastructure stack and components required to run an out-of-the-box general AI environment. This frees data scientists and model trainers to focus on their core competencies and accelerates innovation. Acumos is part of the LF AI Foundation, an umbrella organization within The Linux Foundation that supports and sustains open-source innovation in artificial intelligence, machine learning, and deep learning while striving to make these critical new technologies available to developers and data scientists everywhere.

- **TEKTON:** Tekton is a powerful and flexible open-source framework for creating CI/CD systems, allowing developers to build, test, and deploy across cloud providers and on-premise systems.

- **ETCD:** is a strongly consistent, distributed key-value store that provides a reliable way to store data that needs to be accessed by a distributed system or cluster of machines. It gracefully handles leader elections during network partitions and can tolerate machine failure, even in the leader node.

- **Superset:** Superset is a modern, enterprise-ready business intelligence web application, fast, lightweight, intuitive, and loaded with options that make it easy for users of all skill sets to explore and visualize their data, from simple line charts to highly detailed geospatial charts.

- **WSO2:** is an open-source technology provider founded in 2005. It offers an enterprise platform for integrating application programming interfaces (APIs), applications, and web services locally and across the Internet. WSO2 offers a platform of middleware products for agile integration, application programming interface (API) management, identity and access management, and smart analytics.

- **Ranger:** is a framework to enable, monitor and manage comprehensive data security across the Hadoop platform. The vision with Ranger is to provide comprehensive security across the Apache Hadoop ecosystem. With the advent of Apache YARN, the Hadoop platform can now support a true data lake architecture. Enterprises can potentially run multiple workloads, in a multi-tenant environment. Data security within Hadoop needs to evolve to support multiple use cases for data access, while also providing a framework for central administration of security policies and monitoring of user access.

- **Accumulo:** is a highly scalable sorted, distributed key-value store based on Google's Bigtable. It is a system built on top of Apache Hadoop, Apache ZooKeeper, and Apache Thrift. Written in Java, Accumulo has cell-level access labels and server-side programming mechanisms. According to DB-Engines ranking, Accumulo is the third most popular NoSQL wide column store behind Apache Cassandra and HBase.

- **KNOX**: is an Application Gateway for interacting with the REST APIs and UIs of Apache Hadoop deployments. The Knox Gateway provides a single access point for all REST and HTTP interactions with Apache Hadoop clusters.

- **Snyk:** is an open-source security platform designed to help software-driven businesses enhance developer security. Snyk's dependency scanner makes it the only solution that seamlessly and proactively finds, prioritizes and fixes vulnerabilities and license violations in open-source dependencies and container images.

- **Tableau** is software that helps people see, understand and act on data. It is a visual analytics platform transforming the way we use data to solve problems—empowering people and organizations to make the most of their data.

- **Monte Carlo** is data observability platform that automates monitoring and anomaly detection for data pipelines, protecting against issues related to data freshness, volume, distribution or schema. By coupling proactive alerting with end-to-end observability, data teams get alerted to data issues as soon as they happen. Monte Carlo enables teams to assess the downstream business impact and work quickly to resolve these issues.

- **Databricks** is an organization and big data processing platform founded by the creators of Apache Spark. DataBricks was founded to provide an alternative to the MapReduce system and provides a just-in-time cloud-based platform for big data processing clients.

5. Conclusions

This deliverable has provided two main results: the AI REDGIO 5.0 Reference Architecture and the Reference Implementation.

The AI REDGIO 5.0 Reference Architecture has been defined starting from the analysis of relevant Reference Architectures for Industrial Internet of Things and I4MS past initiatives. Most important sources are requirements from 7 SMEs experiments and 14 Didactic Factories around Europe

The AI REDGIO 5.0 Reference Implementation has been proposed, starting from the AI REGIO Reference implementation, taking into account the integrations due to the tools for AI supporting the services.

The project will defines an open Reference Architecture for AI scenarios totally implementable with open-source components.

The Reference Architecture and the proposed Reference Implementation will be validated in the context of WP6 during the implementation of the use cases.