CoRoSect

D2.3. INITIAL SYSTEM ARCHITECTURE

corosect.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101016953

Author(s)/Organisation(s)	ATOS
Contributor(s)	UM, CERTH, OAMK, TECNOVA, AGVR, ROB, HSEL
Work Package	WP2.Use-cases, user requirements and system architectures
Delivery Date (DoA)	31.12.2021
Actual Delivery Date	26.12.2021
Abstract:	This deliverable contains the first version of the architecture proposed for the CoRoSect integrated System. The architecture is primarily presented in terms of a logical view, which includes its main components and the interactions between them. Other views (implementation, process, deployment) are also discussed. Early examples about integration and process orchestration are given. Based on similar cutting edge industrial and IoT initiatives, some technologies are also proposed for implementation.

Document Revision History			
Date	**Version**	**Author/Contributor/ Reviewer**	**Summary of main changes**
29/10/2021	0.1	ATOS	Table of contents
10/12/2021	0.2	ATOS	Reviewed ToC plus contributions
15/12/2021	0.3	HSEL, UM, CERTH	Contributions sections 2,3, 5
20/12/2021	0.4	ATOS, HSEL, UM, CERTH, OAMK	Contributions to Sections 3, 4, 5
22/12/2021	0.5	ATOS	SotA contributions
23/12/2021	1.0	ATOS	Version for Internal Review
23/12/2021	2.0	ATOS, HSEL, CERTH	Final version

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the EC Services)	
RE	Restricted to a group specified by the consortium (including the EC Services)	
CO	Confidential, only for members of the consortium (including the EC)	

Funding Scheme: Innovation Action (IA) ● Topic: H2020-ICT-46-2020

Start date of project: 01 January, 2021 ● Duration: 36 months

CoRoSect Consortium			
Participant Number	**Participant organisation name**	**Short name**	**Country**
1	UNIVERSITEIT MAASTRICHT https://www.maastrichtuniversity.nl/	UM	NL
2	ETHNIKO KENTRO EREVNAS KAI TECHNOLOGIKIS ANAPTYXIS https://www.certh.gr/	CERTH	GR
3	HOCHSCHULE EMDEN/LEER https://www.hs-emden-leer.de/en/	HSEL	GER
4	LUONNONVARAKESKUS https://www.luke.fi/	LUKE	FIN
5	OULUN AMMATTIKORKEAKOULU OY - OULU UNIVERSITY OF APPLIED SCIENCES https://www.oamk.fi/fi/	OAMK	FIN
6	FUNDACION PARA LAS TECNOLOGIAS AUXILIARES DE LA AGRICULTURA http://www.fundaciontecnova.com/	TECNOVA	ES
7	KATHOLIEKE UNIVERSITEIT LEUVEN https://www.kuleuven.be/kuleuven/	KU LEUVEN	BEL
8	ATOS IT SOLUTIONS AND SERVICES IBERIA SL https://atos.net/en/	ATOS	ES
9	ROBOTNIK AUTOMATION SLL http://www.robotnik.es/	ROB	ES
10	AGVR BV www.agvegroup.com	AGVR	NL
11	NASEKOMO AD https://nasekomo.life/	NASEKOMO	BG
12	ENTOMOTECH SL http://entomotech.es/	ENTOMOTECH	ES
13	ENTOCYCLE LTD https://www.entocycle.com/	ENTOCYCLE	GB
14	SOCIETA AGRICOLA ITALIAN CRICKET FARM SRL https://www.italiancricketfarm.com/	ICF	IT
15	INVERTAPRO AS https://www.invertapro.com/	INVERTAPRO	NOR
16	FIELD LAB ROBOTICS BV https://www.fieldlabrobotics.com/	FLR	NL
17	FoodScale Hub https://foodscalehub.com/	FSH	RS
18	AgriFood Lithuania DIH https://www.agrifood.lt/	AFL	LT
19	CENTRO INTERNAZIONALE DI ALTISTUDI AGRONOMICI MEDITERRANEI http://www.iamb.it/	CIHEAM	IT

Table of Contents

Executive Summary

This deliverable describes the initial version of the CoRoSect's System Architecture, addressing the project's main goal of *introducing a novel digitalized integrated robotic solution based on the Reference Architecture Model Industry 4.0 (RAMI4.0) implemented as an Industrial Cyber-Physical System (ICPS) to be able to support all phases of the insects' lifecycle inside insect farms*.

The architecture is presented as a set of modules grouped in layers that fulfil related functionalities. It also includes the structuring principles that drive their integration within an Industry 4.0 + IoT interoperable system that emphasizes the safe human-robot collaboration whilst enhances the production process efficiency. To define these modules and their structuring principles, the project considers a set of reference architectures for IoT systems and Industrial infrastructures (like ISA-95, RAMI4.0 and the Reference Architecture of the Industrial Internet Consortium), as well as some novel frameworks that interoperate with both, IoT and Industry 4.0 (such as the FIWARE for Industry or COPRA-AP).

This is a Service Oriented Architecture, which, on this first version, identifies the interfaces required to integrate and communicate the layers and components within, as well as the main dataflows that define the templates for the processes orchestrations which will later conform the CoRoSect pilots. These are provided as different views of the proposed CoRoSect System Architecture, that present the logic within layers and building blocks and their corresponding relationships; the common process orchestration; and the development of interconnections (interfaces) between them. All of these are derived from the set of functional and non-functional requirements set by the project's DoA, the RAMI4.0 compliance and the users scenarios provided by the WP2.

The standards, technologies and digitization methodologies that also includes this text will support the development of the CoRoSect components and the whole integration process that the project's work packages will lead. The feedback from these will, in turn, support the evolution of the CoRoSect initial architecture towards its Advanced version, to be presented in D2.4.

List of figures

List of Abbreviations and Acronyms	
AAS	Asset Administration Shell
AGV	Autonomous Guided Vehicle
AR	Augmented Reality
ARM	Architectural Reference Model
B2B	Business to Business
CD	Continuous Development
CI	Continuous Integration
CIM	Computer Integrated Manufacturing
COPRA-AP	Common Open Platform Reference Architecture for Agile Production
DB	Data Base
DBMS	Data Base Management System
DIH^2	Digital Innovation Hubs
DMS	Data Management (System)
DoA	Document of Action
DoW	Document of Work
D-Robot	(Stacking)-Destaking Robot
DS	Decision-Making System
DSS	Decision Support System
DT	Digital Twin
EC	European Commission
ERP	Enterprise Resource Planning
F4I	FIWARE for Industry
GUI	Graphical User Interface
H2020	Horizon 2020
HMI	Human-Machine Interface
HW	Hardware
I/O	Input/Output
I4.0	Industry 4.0
ICF	Italian Cricket Farm
ICPS	Industrial Cyber-Physical System
I-Crate	Intelligent Crate
IDS	International Data Space
IDS-RAM	IDS-Reference Architecture Model
IEC	International Electrotechnical Commission
IIoT	Industrial IoT
IIRA	Industrial Internet Consortium Reference Architecture
IMS	Information Management System
IMS	Information Management System
IoT	Internet of Things
IoT-A	IoT-Architecture
ISA	International Society of Automation
IT	Information Technologies

JSON	JavaScript Object Notation
LiDAR	Light Detection and Ranging
M12	Month 12
MES	Manufacturing Execution System
ML/DL	Machine Learning/Deep Learning
MOM	Manufacturing Operations Management
MQTT	Message Queue Telemetry Transport
M-Robot	Manipulation-Robot
NGSI	Next Generation Systems Interfaces
O1	Objective 1
OAuth	Open Authorization
OPC-UA	OLE (Object Linking and Embedding) for Process Control-Unified Architecture
OT	Operational Technology levels
PA	Process A
PLC	Programmable Logic Controller
PMS	Process Management System
Pub/Sub	Publish/Subscribe
R&D	Research & Development
RA	Reference Architecture
RAMI4.0	Reference Architecture Model for I4.0
RC	Robot Controller
ROS	Robot Operating System
SCADA	Supervisory Control and Data Acquisition
SFM	Shop Floor Manager
SLAM	Simultaneous Localization And Mapping
SoA	Service oriented Architecture
SQL	Structured Query Language
SW	Software
TM.PA	Task M from Process A
UK	United Kindom
VR	Virtual Reality
WP2	Work Package 2
YAML	Yet Another Markup Language

1 Introduction

1.1 Scope and objectives of the Deliverable

According to CoRoSect's DoA, the project "*will bring new insight to automated insect farming by introducing a novel digitalized integrated robotic solution based on the Reference Architecture Model Industry 4.0 (RAMI4.0) implemented as an Industrial Cyber-Physical System (ICPS) to be able to support all phases of the insects' lifecycle inside insect farms. The fundamental aim of the system (and the great innovation it provides) will be to provide repetitive but also cognitively and physically demanding tasks, like transferring and handling of crates (de-stacking and stacking), monitoring of environmental conditions, larvae separation/detection, insect feeding, which require increased manual effort or continuous human supervision, with correspondingly automatic robotic-based procedures, as service in an I40-compliant Information-Communication Infrastructure*".

The objective of this deliverable is to present **the first version of the Service Oriented CoRoSect System Architecture** that supports this main objective. This is based on the requirement and first functionalities provided by end-users and selected use cases compiled by WP2, and sets the rules to implement a "*collaboration environment, where humans and robots will harmoniously share and undertake at the same time different processing and manipulation tasks, targeting the application case of insect farming*".

In this context, CoRoSect System is proposing a service-oriented architecture aligned with cutting edge Industrial systems that manages production process. Current trends in industrial environments include devices and infrastructures that exploit IoT systems (e.g. sensing devices) and IoT technologies (e.g. communication protocols or data models). In the case of CoRoSect, this proposed system will be designed to support and integrate IoT architectures with Industrial mechatronic systems, so this deliverable analyses first most used reference architectures, from both the Industrial and the IoT sectors, with particular emphasis on those that combines (or allows) devices from smart factories and smart cities. RAMI4.0 will guide the final implementation of its architecture, so all the subsystems and integration interfaces derived from this analysis will be compliant with this.

This first version of the architecture is introduced based on several interrelated modules and the interfaces between them. The deliverable provides initial specifications for each module that comprises the architecture, along with its role in the implementation of the project's use cases. It also includes the interactions between these modules that illustrates the template for CoRoSect's processes orchestration.

Based on current implementations and EC initiatives that covers similar integration targets, this document also proposes several technologies, from IoT and Industrial environments, that without being mandatory, can help on further implementations of the corresponding interfaces.

1.2 Relation to other CoRoSect's work packages

The architecture produced by this deliverable is a key component for the project's workplan, as it is being used to drive the digitization and integration processes addressed in other work packages. In particular:

- Within WP4 (Farm-level modelling and orchestration) the CoRoSect's process view helps to identify the relationships between its MES layer, and so the main functionalities to be addressed to the Shop Floor Manager, the Decision Support System and the Information Management System.

- WP6 (Robotic actions planning and control) and WP7 (Cognitive robots and smart mechatronics) will use the components in the logical view to identify the interfaces to be developed by their corresponding controllers. These will expose their devices/systems and collects required information from the MES.
- In a similar way, WP8 (Human-robot collaboration schemes) will have, from this architecture, the required interfaces to interact with all the Shop Floor components and to collect information form the MES.
- The system architecture will have a key role in WP9 (Secure platform integration) as this will be the map to plan, run and test the full system integration. The set of interfaces and relationships between components described here will be performed mainly within this work package and so, feedback from the tasks within will guide the second version of the CoRoSect System architecture.

1.3 Methodology and structure of the Deliverable

The methodology to specify the initial version of the CoRoSect System Architecture follows these steps, derived form a common approach on architecture's definition:

- a first review and Analysis of the state of the art, including relevant reference architectures, standards and initiatives related to the Industry 4.0 and IoT frameworks to get insights in terms of interoperability and integration
- synthesis and Specification of the CoRoSect's reference architecture, as a first diagram that addresses and links the components from the initial document of action with the functional and non-functional requirements identified within the interactions with the end users. This provides a first set of layers to propose a detailed CoRoSect system.
- definition of the CoRoSect System Architecture, based on the combination of concepts from the analysis of the previous steps, but also based on the specification of solutions that successfully address the main functional and technical requirements for CoRoSect. This specification is provided in terms of a description of its main modules and of the interfaces between them. It also includes both logical and development views of the architecture, plus the dynamic behaviour of the common process orchestration.

The validation of the CoRoSect architecture against the project's scenarios and use cases is left for the second version of this architecture (D2.4 Advanced System Architecture), when the final requirements (both, functional and technical/non-functional) are better identified and the end users' scenarios for validation are defined by WP9.

According to the presented methodology, the structure of this deliverable is as follows:

- Section 2 follows the introduction of this deliverable and presents a summary with the most relevant (to CoRoSect) aspects of each reference architecture analysed, with special emphasis on the RAMI4.0 standard, as referred in the DoA. It also describes two initiatives that exploits these reference architectures to propose Industry 4.0 + IoT integrated frameworks.
- Section 3 extracts requirements from the common CoRoSect's usage scenario to propose the first approach to CoRoSect System, the CoRoSect RA. This identifies and group main components and systems in functional levels interconnected.
- Section 4 introduces the CoRoSect System Architecture, following the main principles of the proposed reference architecture. It provides a detailed description of the functionalities of each main modules of the architecture, along with a high-level description of the interfaces

and interactions between them. It also provides guidelines for its implementation following RAMI4.0 standard.

- Section 5 proposes a set of candidate technologies to implement integration interfaces layers functionalities, based on previously analysed (Section 2) frameworks.
- Section 6 is the final section with the conclusions of the deliverable. It draws the main conclusions from the CoRoSect architecture specification process and provides a first overview of the next version/release of this deliverable (D2.4 Advanced system architecture)

2 Internet of Things and Industrial 4.0 references. State of the Art

As its main target, CoRoSect is proposing a novel digitalized and integrated robotic solution which is based on the Reference Architecture Model Industry 4.0 (RAMI4.0). This solution is intended to support the defined production processes that compose the CoRoSect pilots. It also integrates all the CoroSect Shop floor components. These components come mostly from the industry sector (Robots, AGVs, etc.) but also includes sensing devices (such as the I-Crates) more related to the IoT sector. This section provides a soft overview of the current relevant standards from IoT and Industry so the proposed CoRoSect System can integrate not only components but also both worlds' infrastructures.

2.1 Standard Reference Architectures and Reference Architecture Models

2.1.1 CIM Pyramid (ISA-95)

In an insect production system or any other production system, the physical components are necessary for conducting the production procedure. Similarly, for performing activities like order management, resource, and asset management, work plan definition, various IT related components are present in an insect production system.

In an insect production system or any other production system, the operations are managed and controlled in the fashion of the 5-layered architecture known as the CIM Pyramid or the ISA-95 functional hierarchy. This ISA-95 [1] standard defines the various levels in which the operations facility is divided from the management, control, and automation point of view.

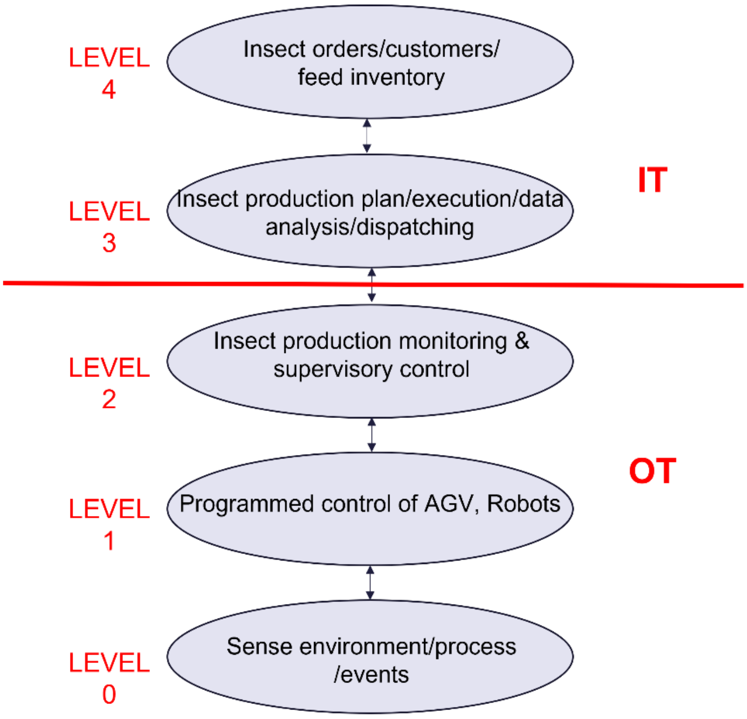


Figure 1. Functional Hierarchy of Insect Production System based on ISA-95

The operations of insect farms are divided between the levels of OT and IT. Levels 4 and 3 are recognized as Information Technology (IT) levels and Levels 2, 1 and 0 as Operational Technology levels (OT). This is depicted in Figure 1.

1. Level 4

In Level 4, are located all the functions associated with Customer Relationship Management (CRM) and Enterprise Resource Planning (ERP), including e-Procurement. This is where the data regarding the insect orders, customers and feed inventory is positioned.

The range of activities included in this level are a) getting customer orders for insects, b) planning for inventory for the feed for insects, c) planning for logistics of harvested insects, d) ordering from suppliers, e) planning for production and operation, f) planning for maintenance of components of the insect production system.

2. Level 3

It is Level 3 which functionally covers all the activities of manufacturing operational management, i.e., management, control, and operations. This is the level that will be responsible for transforming the business objectives defined at level 4, into operational objectives to be achieved in Levels 1 and 2.

It is here where the customer orders of insects and their associated work plans are transformed into production execution recipes. This is where the data analysis of the insect production process takes place.

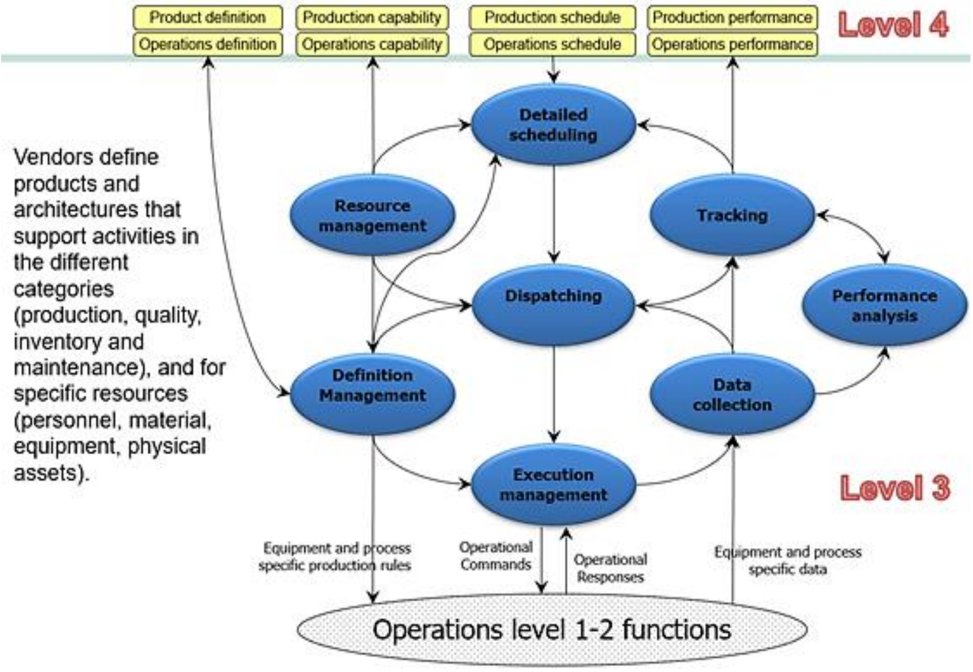


Figure 2. The MOM Generic Activity model (level 3 related to level 4 (IT) and levels 1-2 (OT) [2])

As shown in Figure 2 an essential set of 8 basic functions, performed by associated SW modules, will have to be provided by this level. This level is typically implemented by a general Manufacturing Execution System also known as MOM [2].

Functionally this can be explained by a scenario from the Insect Production Farm. Once the business processes are defined like in this case for Insect Production Farm there might of a customer order of 2000 Black Soldier Fly. This order will be taken by ERP, transformed into an operation request.

This operations request shall be taken care of by an MES by preparing a detailed schedule of the production, then dispatching/reserving the needed assets (e.g., AGV, Robots and Intelligent Crates) and then the Execution Management Activity shall take care of making sure that the assets work at the given time and following the invariants and preconditions set.

The Data collection activity shall be responsible for collecting the production-specific data from the assets, storing it, and then exposing it for various activities like tracking, performance analysis and dispatching.

3. Level 2 And Beyond

The OT level has the operational technology components like the robots with robot control, AGV with their PLC for operating in the production units in an insect farm. These components are in real-time communicating with their associated sensors and actuators.

The SCADA system is responsible for monitoring the data from the PLC'S, RCs etc. This is where the shopfloor managers or HMI Operators visualize the status of insect production.

The implementation of levels of ISA-95 in an insect production system by respective software and hardware can be seen in Figure 3.

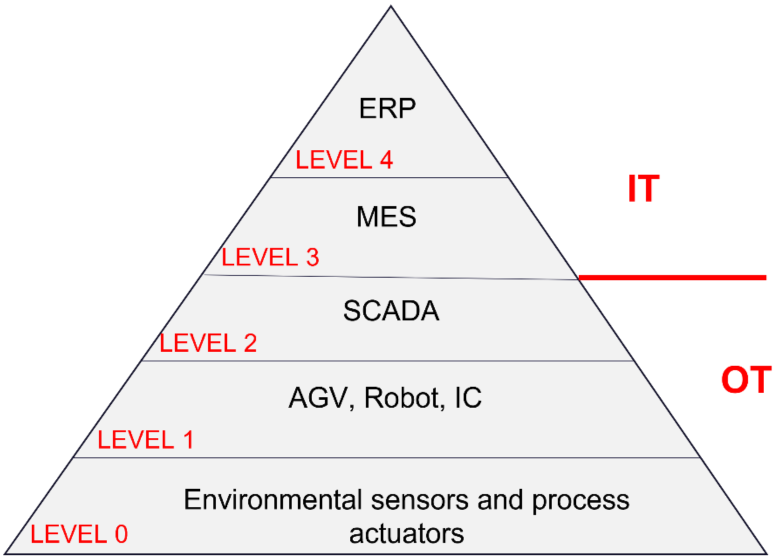


Figure 3. Typical Implementations of the ISA-95 Levels in Insect Production System

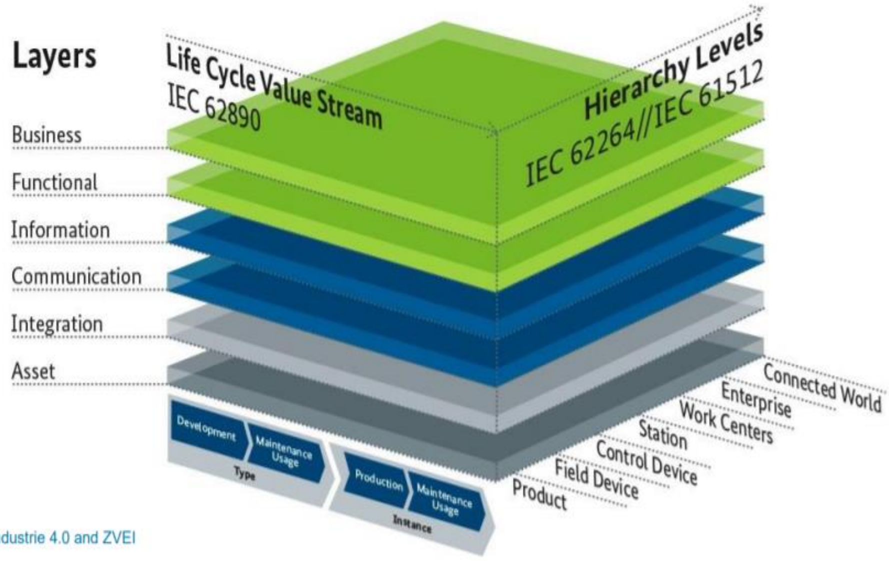
Level 4 in an insect production system is implemented using ERP software. Level 3 is implemented using MES software. This is called the IT Layer. In the OT layer, Level 2 is implemented using SCADA software. The Level 1 of the insect production system will have AGV, Robots and Intelligent Crates. At Level 0, there would be environmental sensors, process actuators that are close to the production process.

It is important to note that Level 4 and Level 3 and Level 2 work in different time scales. Level 4 works in the time level of days and months, whereas Level 3 usually works around the time level of hours. Level 2 works in minutes or near real-time mode.

2.1.2 DIN SPEC 91345 RAMI 4.0

DIN SPEC 91345 RAMI 4.0 stands for Reference Architectural Model Industry 4.0 and comes from the standardization effort of DIN SPEC 91345. In RAMI 4.0 a mutual understanding is maintained among all the stakeholders from the perspective of business, functional, information and communication.

DIN SPEC 91345 RAMI 4.0 helps in changing assets into modules that are self-contained and inherit the concepts of security and data privacy effectively creating a cyber representation of the physical assets be it hardware or software. RAMI4.0 uses the concepts of SoA where these modules will be communicating among themselves using services in the network of services.



Graphics © Plattform Industrie 4.0 and ZVEI

Figure 4. RAMI 4.0 Graphical Representation [3]

On its graphical representation (Figure 4), DIN SPEC 91345 RAMI 4.0 has three 3 dimensions to it. The description of these three dimensions can be seen below:

a) **Hierarchy Levels axis**: The hierarchical levels on the right of the map are adapted from the world-known standards like IEC 62264 and IEC 61512. These also represent ISA-95 [1] and ISA-88. It is important to note that these hierarchical levels define the various functional levels into which the assets or components of a production system or factory can be classified. From the original levels of ISA-95, Connected World and Product are added here to also include them while discussing the application of Industry 4.0.

b) **Life Cycle Value Stream axis**: The left axis represents the life cycle of each of the assets that are mapped in the hierarchical level axis. It is based on standard IEC 62890 [4]. The lifecycle is divided into multiple phases. These phases are differentiated based on Types and Instances. The Types denote when the asset is in the building or prototyping stage. The instance phase starts when this actual asset is manufactured and gets engaged in a working plant. It is important to add the lifecycle dimension to the DIN SPEC 91345 RAMI 4.0 model because the main objective in Industry 4.0 is to create new business models based on the data present in the various stages of the assets. It is also important that the data from all the stages are connected because otherwise, data like customer orders, production history, value chain and maintenance remain disconnected.

c) **Vertical Layers**: The vertical axis composes of different layers belonging to business, functional, information, communication, integration, and asset. These layers form the basis of digitalization or create the cyber representation of the asset that belongs to one of the levels in the hierarchy. It is also important to consider the lifecycle phase when digitalizing. Thus, the vertical layers form the very basis of specifying and implementing the digitalization of each of the assets present in a factory or production system [5].

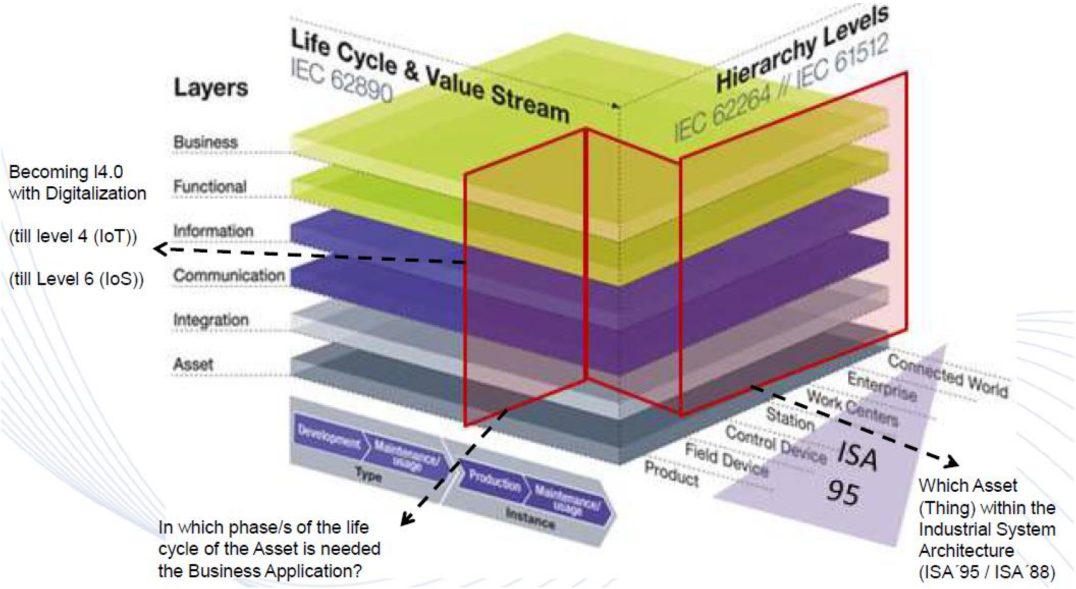


Figure 5. Navigating the RAMI 4.0 [6]

With the combination of all these axes, all the aspects of Industry 4.0 can be successfully mapped. The model has an extensive reach from helping in mapping the business requirements with the associated functions, while also discussing what communication medium could be used for communicating the services. The layers would always help in answering questions like What is Digitalizing? Why Digitalizing and How Digitalizing (Figure 5). It can be noticed that DIN SPEC 91345 RAMI 4.0 is domain neutral. It can be used in any industry that wants to explore the application of the Industry 4.0 concepts and implement them in their facility. RAMI 4.0 is needed to understand manufacturing and its constituent supply chain in an industrial context.

2.1.3 Industrial Internet Consortium Reference Architecture (IIRA)

The IIRA [7], on its current version 1.9, defines a common architecture framework for developing interoperable IoT systems within different vertical industries. This is an open, standards-based architecture that fosters interoperability between of IoT technologies, but mainly from an abstract and high-level perspective. Because of this, it is useful to drive the structuring principles of an IoT based infrastructure, supporting the integration and coesixtence of different technologies an existing IoT systems. This also facilitates developers and stakeholders’ collaboration.

In this sense, IIRA represents general IoT infrastructures using four different viewpoints (Figure 6), coming from the analysis of various Industrial IoT use cases developed by the IIC, the identification of relevant stakeholders and the determintation of the proper framing of concerns. These four viewpoints are named i) Business (stakeholders and business visions); ii) Usage (expected system usage); iii) Functional (components and functionalities); and iv) Implementation (technologies for functions’ implementation).

Within these four viewpoints, the functional one focuses on the functionalities of an IIoT system: it specifies distinct functionalities in the form of the so-called “functional domains”. Functional domains can be used to decompose an IoT system into a set of important building blocks, which are applicable across different vertical domains and applications. As such functional domains are used to conceptualize concrete functional architectures. The IIRA decomposes a typical IoT/IIoT system into five functional domains, namely a control domain, an operations domain, an information domain, an application domain and a business domain as outlined in Figure 7.

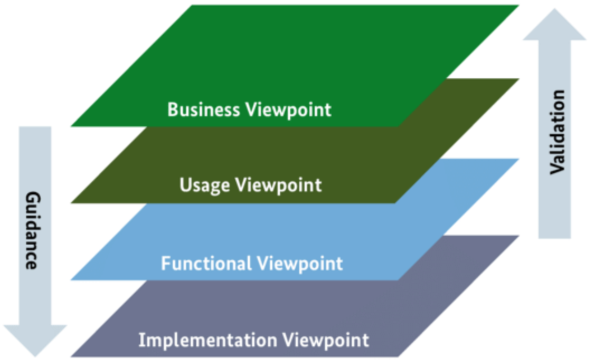


Figure 6. Functional Domains in IIRA

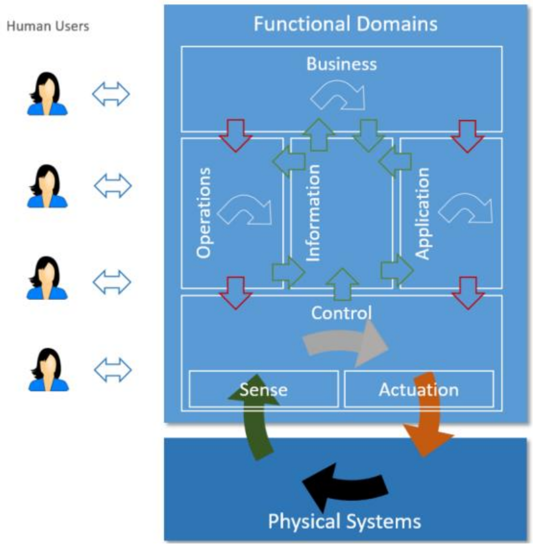


Figure 7. Functional Domains in IIRA Functional Domains in IIRA

The implementation viewpoint of the IIRA is based on a three-tier architecture, which follows the edge/cloud computing paradigm, as shown in Figure 8. It includes an edge, a platform and an enterprise tier.

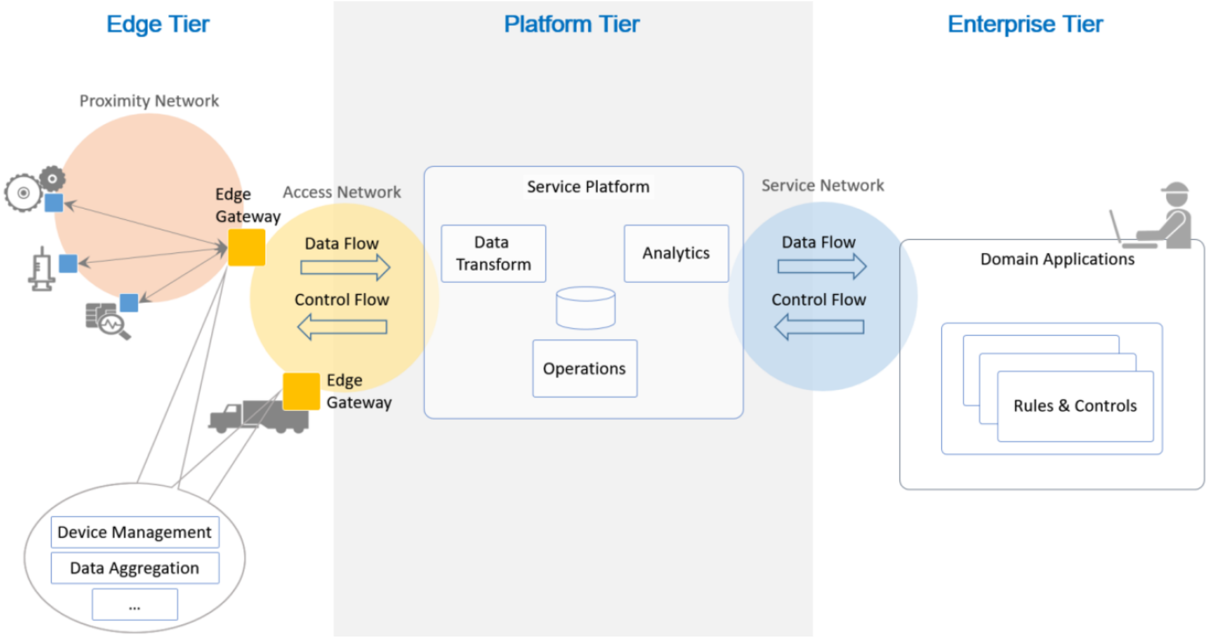


Figure 8. Three Tier Architecture for IIoT Systems – Implementation Viewpoint for IIRA

CoRoSect develops a general-purpose system architecture able to combine IoT-based system with mechatronics developed for industrial environments, which emphasizes on integration. This IIRA functional viewpoint and its five domains division provides an interesting common model to analyse each system, initially from the IoT sector but comparably to robotic systems for dissecting their functionalities and designing an integration path for each of them. CoRoSect considers and combines this approach with others' Reference Architecture's integration mechanisms to propose a common RAMI 4.0 integration model.

2.1.4 IoT-A

The European Lighthouse Integrated Project has addressed for three years the Internet-of-Things Architecture and created the proposed Architectural Reference Model (ARM) [8] together with the definition of an initial set of key building blocks. The primary goal of the ARM concept is to solve the interoperability problem that affects IoT based systems: *"Many IoT-enabled solutions exist with recognised benefits in terms of business and social impact, however they form what we could call a set of* ***Intra****nets of things, not an* ***Inter****net of things*".

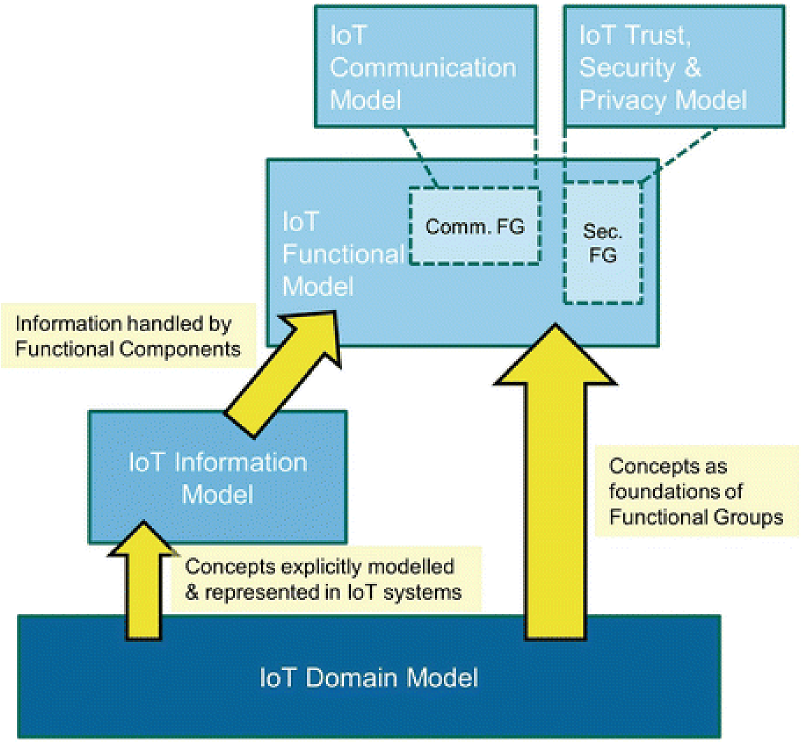


Figure 9. IoT-A Submodels

The ARM consists of several sub-models that set the scope for the IoT design space (Figure 9) and address different architectural views. The primary and thus the key model is the IoT Domain Model, which describes all the concepts that are relevant in the Internet of Things, so the other models and the IoT Reference Architecture itself are based on the concepts introduced here. While certain models, such as the IoT Communication Model and the IoT Trust, Security, and Privacy Model might be less critical in certain application scenarios, the IoT Domain Model is mandatory for all usages of the IoT ARM.

Based on this IoT Domain Model, IoT ARM develops the IoT Information Model, which defines the structure (e.g. relations, attributes) of IoT related information in an IoT system on a conceptual level. This is used to represent IoT Devices, IoT Services and Virtual entities. On the other hand, the IoT

Functional Model identifies groups of functionalities, of which most are grounded in key concepts of the IoT Domain Model. A number of these Functionality Groups (FG) build on each other, following the relations identified in the IoT Domain Model. The Functionality Groups provide the functionalities for interacting with the instances of these concepts or managing the information related to the concepts, e.g. information about Virtual Entities or descriptions of IoT Services. The functionalities of the FGs that manage information use the IoT Information Model as the basis for structuring their information.

These concepts of IoT-A Information and Functional models links with the Asset Administration Shell (AAS) concept of the RAMI4.0 that digitalises mechatronics (section 4.4.1) to be applied in CoRoSect.

2.1.5 International Data Space – IDS

As part of the currently targeted Industry 4.0, digitization is fundamentally changing companies, as it enables new business models and changes the self-image of entire industries. Digitization means that data makes a significant contribution to corporate success. The International Data Spaces initiative (former Industrial Data Space), aims at creating a secure data space that supports enterprises of different industries and different sizes in the autonomous management of data. It is promoting its Reference Architecture Model (IDS-RAM) [9] to establish an international standard for building data-driven ecosystems, products and services. To achieve this goal, the Association pools the requirements from various industries and provides use cases to test the results gained from the model's implementation.

The International Data Spaces are made up of the entirety of all endpoints (connectors) as well as various components (software) such as brokers, clearing houses, identity providers and the app store. The standard is intended to materialize in the IDS-RAM itself, but also defined methods for secure data exchange and data sharing facilitated by the IDS Connector, the central technical component of the International Data Spaces.

IDS-RAM is described using multiple layers: business, functional, process, information and system. Transversal functionalities to all of them are security, certification and governance. In contrast to IoT-A ARM, IDS focuses its specification of roles within the business layer in the data flows between different domains or data spaces, defining key participants like Data Owner, Data Provider, Data Consumer, Data User or Broker Service provider.

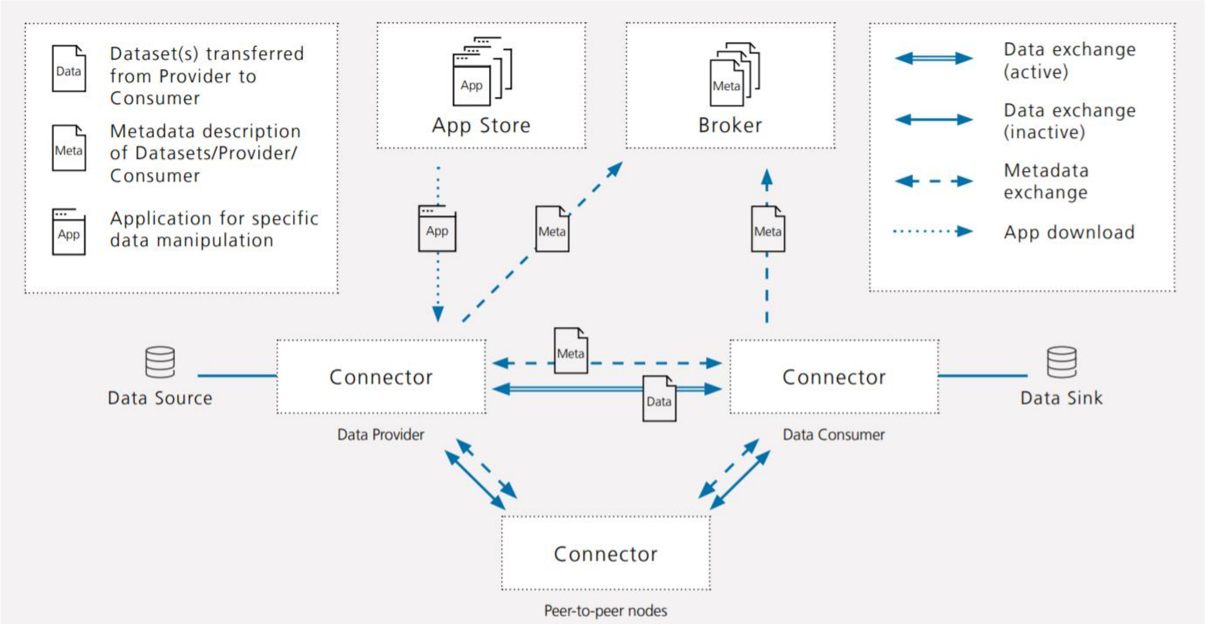


Figure 10. IDS-RAM System Layer [9]

In the System Layer, these roles are mapped onto a concrete data and service architecture in order to meet the requirements, resulting in what is the technical core of the Industrial Data Space. From the requirements identified, three major technical components can be derived: i) the Connector; ii) the Broker; and iii) the App Store. The interaction of these components on the System Layer is depicted in Figure 10.

The IDS Connector represents a standardized interface to the International Data Spaces for the participating companies. This access point enables, on the one hand, targeted and controlled provision of their data, and, on the other hand, authorized access to data from other participants. The International Data Spaces are therefore not a central data store, but rather follow a federal architectural concept. This concept and its IDS-RAM approach will be also useful to define the CoRoSect's components integration and data sharing processes, while maintaining its Industry 4.0 compliance.

2.2 Review of Architectures from Related R&D Projects and Initiatives

Besides the Industry and IoT standards and reference architectures, this analysis also included some architectural implementations currently being developed within other EC initiatives that address similar targets to CoRoSect. Approaches, enablers' catalogues, and expertise are valuable when defining the CoRoSect System Implementation.

2.2.1 FIWARE and FIWARE for Industry

Trustworthy data exchange across systems and organizations, together with the definition of open standard APIs and information models enabling portability and interoperability of applications, are becoming key drivers of Industry 4.0 and its evolution. FIWARE [10] is working on reference architectures and information models to help manufacturing companies in their digital transformation.

FIWARE [11] is an open-source initiative that, since 2011 works to create a platform to leverage on the combination of disruptive technologies like the Internet of Things (IoT), Big Data or Cloud architectures. Rather than proposing a tightly and closed solution that targets the specific requirements of a single domain, FIWARE eases the creation of new applications in multiple verticals since its catalogue contains a rich set of components that can be connected, combine and deployed in a flexible way.

In 2016 rose the FIWARE Foundation, which drives the definition – and the Open Source implementation – of key open standards that enable the development of portable and interoperable smart solutions in a faster, easier and affordable way, avoiding vendor lock-in scenarios, whilst also nurturing FIWARE as a sustainable and innovation-driven business ecosystem. FIWARE foundation leads the adoption of FIWARE standards in different domains such as agriculture (Smart Agrifood), Smart Cities, Energy (Smart Energy) and more recently, Industry (Smart Industry). Within this last domain, it releases the FIWARE for Industry (F4I) [12] which instantiates and extends FIWARE Technologies towards the implementation of Smart-Digital-Virtual Factories of the Future through an Academy of Industry 4.0 methods and tools, a Lab of Open Source components, a Hub of reference architectures and digital platforms, as well as a Showcase of Industry 4.0 success stories and best practices. F4I joins its knowledge and expertise on IoT infrastructures with the current Industry standards to provide a reference architecture (Figure 11) compliant with existing industry architectures such as the Reference Architecture Model Industrie 4.0, the Industrial Data Space Reference Architecture or the Industrial Internet Consortium Reference Architecture.

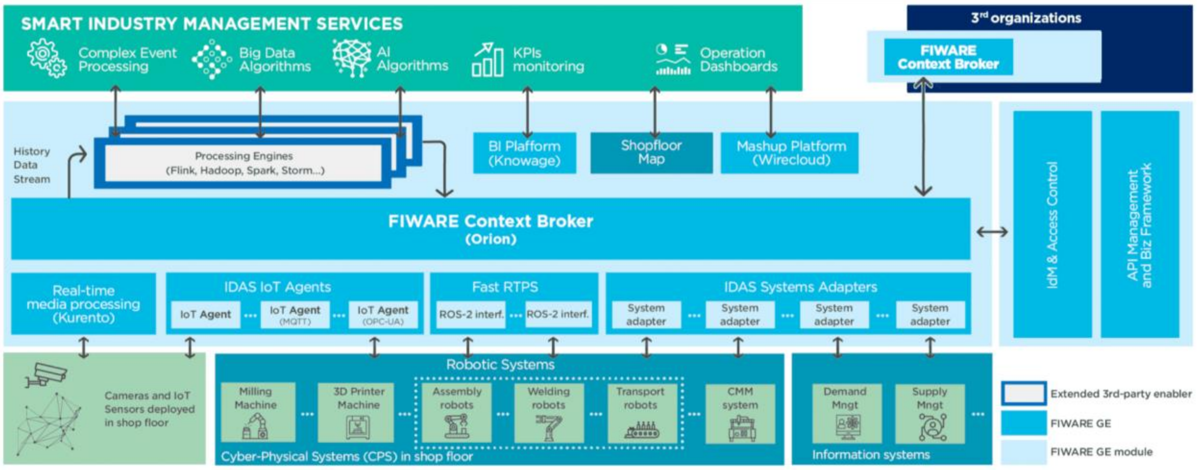


Figure 11. FIWARE for Industry (F4I) architecture

2.2.2 Common Open Platform Reference Architecture for Agile Production (COPRA-AP)

COPRA-AP is an Open-source digital platform that connects the real world of a physical production system with the advanced Agile Production applications to be developed in the context of DIH² [13]. Among the main features of this platform are its openness (in order to connect any real world with any application), adherence to standards (in order to develop and maintain just a few standard gateways) and security-privacy preservation (in order to implement both personal and non-personal B2B data exchanges). It is aligned with the reference architectures for Digitising the European Industry and reuses the outcomes provided by other H2020 related Industry projects. COPRA-AP is intended as a platform of platforms that supports:

- The creation of harmonized virtual representations of the production environments, composed of standardized digital entities and APIs for context data management.
- The integration of the physical environment with the virtual data space through convenient adapters.
- The acceleration of the production environment through multiple services and capabilities (e.g., integrated planning and execution, prescriptive analytics, KPIs monitoring), which are derived from the ability to monitor, control, augment and visualize the actual world through the virtual data space.

The proposed reference structure (Figure 12) can be divided into three main levels: i) the Industry 4.0 compliant components, defining the production environment (also identified as the shop floor); ii) the virtual data space, where all the components of COPRA-AP will run to adapt information from the I4.0 bottom level, manage collected data serve information to both, upper and lower levels; and iii) the connected world, with platforms or external organizations implementing COPRA-AP compliant solutions which expose virtualizations and interact with the COPRA-APs virtual data space.

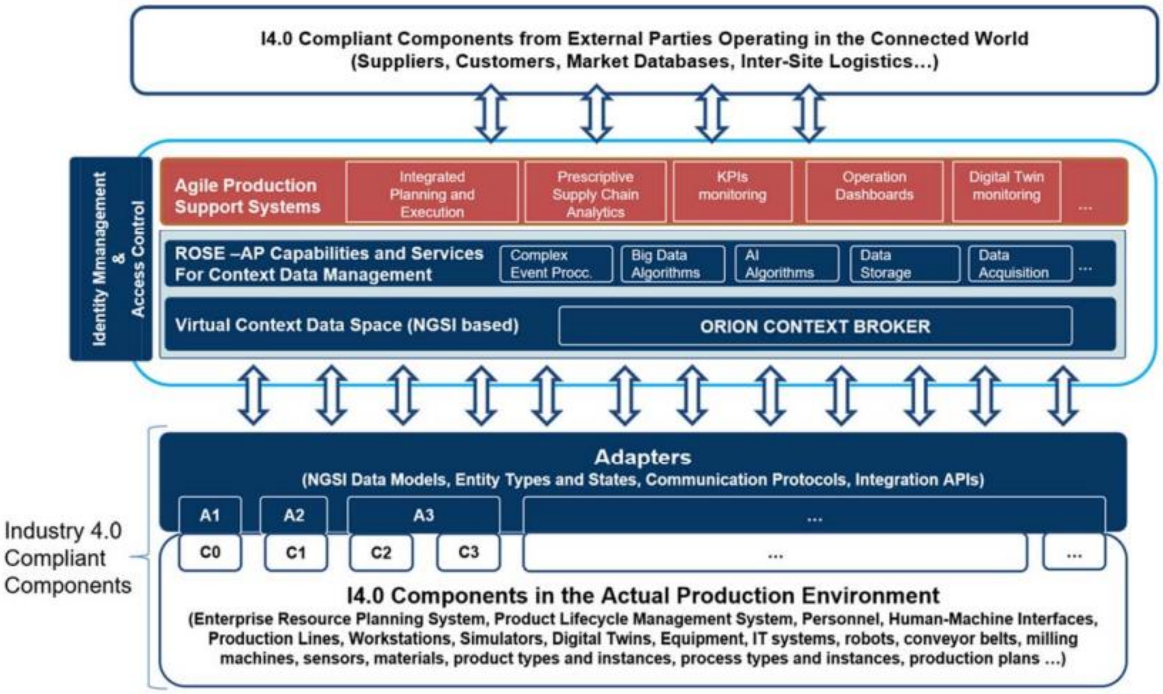


Figure 12. DIH^2 COPRA-AP framework [14]

Hence this is an integrated and harmonized environment not limited to intra-site communications, but also supporting inter-site communications between the production environment and remote I4.0 components from suppliers, customers, inter-site logistics, etc.

The virtual data space is generated and managed by the central gateway, the FIWARE Orion Context Broker. This virtual data space can be interpreted as a photograph of the current state of the real world or as a blackboard where each virtualization has its last state represented. COPRA-AP offers also a set of components for Robotics-based Open Standard Enablers for Agile Production (ROSE-AP Catalogue) which extends the capabilities and services offered by the core virtual space. In general, these capabilities and services provide generic abilities to store, process, augment and visualize the virtual space. Lastly, the purpose of the bounding box that encapsulates the COPRA-AP in Figure 12 illustrates that the virtual space is a secured data-space where different policies can be implemented to offer a convenient data bus for each type of data exchange.

3 CoRoSect's platform requirements

Starting from the DoW that introduces the main targets of the CoRoSect integrated system and supplied by the work done in Task 2.1 and Task 2.2 regarding user's requirements and technical specifications, this section identifies the minimum requirements from an architectural perspective in terms of components addressing specific functions. Considering also the previous analysis of various RAs for IoT and Industrial environments and their functional characteristics, it proposes herewith the CoRoSect Reference Architecture that will support all the following CoRoSet's system views and will guide the development, deployment and integration steps.

3.1 Usage scenarios

According to the project's Document of Work (DoW), CoRoSect is providing a novel service-oriented, full integrated and open human-robot working environment to support and enhance the production pipeline in Industry 4.0 insect farms. This environment will interconnect the different subsystems (robots, sensing devices, vehicles, information and management systems, controllers, etc.) that conform to the final automated production plant system, which interweaves communication, information, and security layers to finally compose the CoRoSect System (Figure 13). This integrated system will be intended to support the different factory tasks to create the CoRoSect scenarios and so, to bring the CoRoSect's Innovations to them.

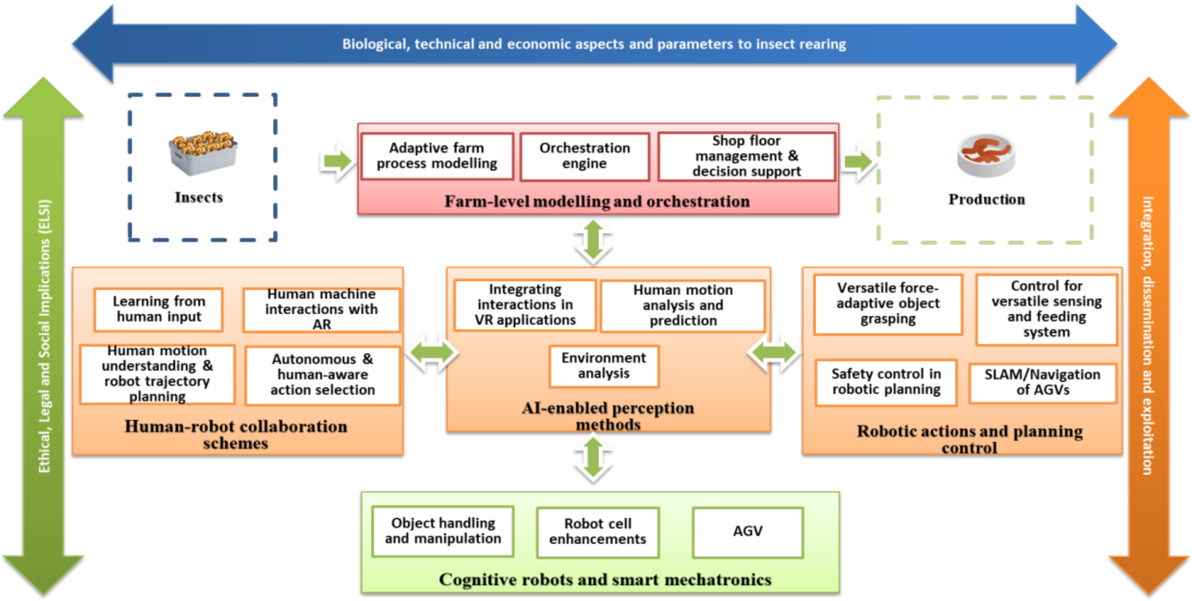


Figure 13. CoRoSect System from DoA

As presented in D2.1 (Use cases and user requirements), this system will be deployed in 5 different end-users' scenarios, corresponding with 5 insect farms' premises:

- **Italian Cricket Farm** (ICF) in Italy works with Acheta domesticus (crickets) for the pet food market, but with the vision of the future insect market for human food. With this near future target in mind, the company looks for an increase in automation processes to ensure an increase in their production.
- **Entocycle**, in the UK, is an early-stage biotech company specialising in black soldier fly's rearing. It already uses proprietary technology to manage and optimize some of its plant

processes. It is interested on the industry standards' adoption to provide a fully automated and standardised end to end facility for insects' rearing.
- **Nasekomo** is a small-scale factory in Bulgaria breading black soldier fly, which already has some automated processes running supported by proprietary technologies. The company pursues the expansion of the automation and robotization of its processes according to Industry 4.0 standards.
- **Invertapro** (Norway) is focused on the yellow mealworm (Tenebrio molitor) rearing. The company has some previous related expertise, but its objective is to validate processes and technology for the full automation of its plant.
- **Entomotech**, in Spain, is a R&D company that provides the industry with biotechnological tools, and is currently planning an expansion in the insect farming sector for producing insects for human consumption. Their premises and expertise in production systems will be used by CoRoSect as a testbed to validate the pilots, previously to their final deployment.

WP2 and WP10 have identified a set of processes for each of these scenarios, according to the end-users' expectations and the CoRoSect's technical partners capabilities, defining the CoRoSect pilots. These pilots will be deployed in the corresponding end-users' premises to validate and evaluate the CoRoSect System performance. In this line, D2.1 (Uses cases and user requirements) jointly with D10.1 (Pilot scenarios) construct these scenarios considering three levels of deployment: a) the **functional** level, that describes the different lay outs for each working area; b) the **hardware** level, that identifies the robotic systems and devices involved on each pilot to be implemented on each scenario; and c) the **software** level that lists all the actions and interactions required per each component to perform the tasks linked to each pilots' processes.

The objective of the proposed pilots is so to evaluate the CoRoSect's innovations by deploying and integrating all the CoRoSect technologies and components, bringing the CoRoSect's Dynamic Cell into reality. This D-Cell concept is based on the efficient orchestration of a set of elements (from D10.1):

- AGVs and SLAM/Navigation module, to operate automated vehicles on farm floors with human workers in a safe way. This also includes the AGV and its corresponding controller.
- Robot trajectory planning for safe Human-Robot collaboration service.
- Safety control in robotic planning module, to assist on a safe human/robot objects' manipulation.
- Versatile force-adaptive object grasping service through parameter adaptive manipulation controllers.
- Intelligent Crates (I-Crates) with printed sensors that monitor the critical variables and support the automation of the insect rearing process.
- Visual analysis module, with computer vision techniques and/or thermal cameras to monitor insect's behaviour, including the controllers to integrate these components.
- Robotic systems to execute all the rearing processes at the shop floor level, covering the hardware (robots and devices) and the software (controllers). These include the Stacking-Destaking robots (D-Robot) with its gripper and the Manipulation Robot (M-Robot) and the attached tools.
- Information Management System (IMS) for robotic appliances to homogenise, store and drive all the information captured by the CoRoSect whole system, synchronising and updating all components.
- Human-Robot Collaboration schemes to improve the productivity and quality of work of the designed system

- Learning from human input mechanisms to allow robots to interact with humans to execute actions or proposed solutions and update their knowledge databases.

According to the above mentioned, the Figure 14 presents an overall approach with all components to be deployed in all the five usage scenarios to support and run the defined pilots and to evaluate the CoRoSect performance.

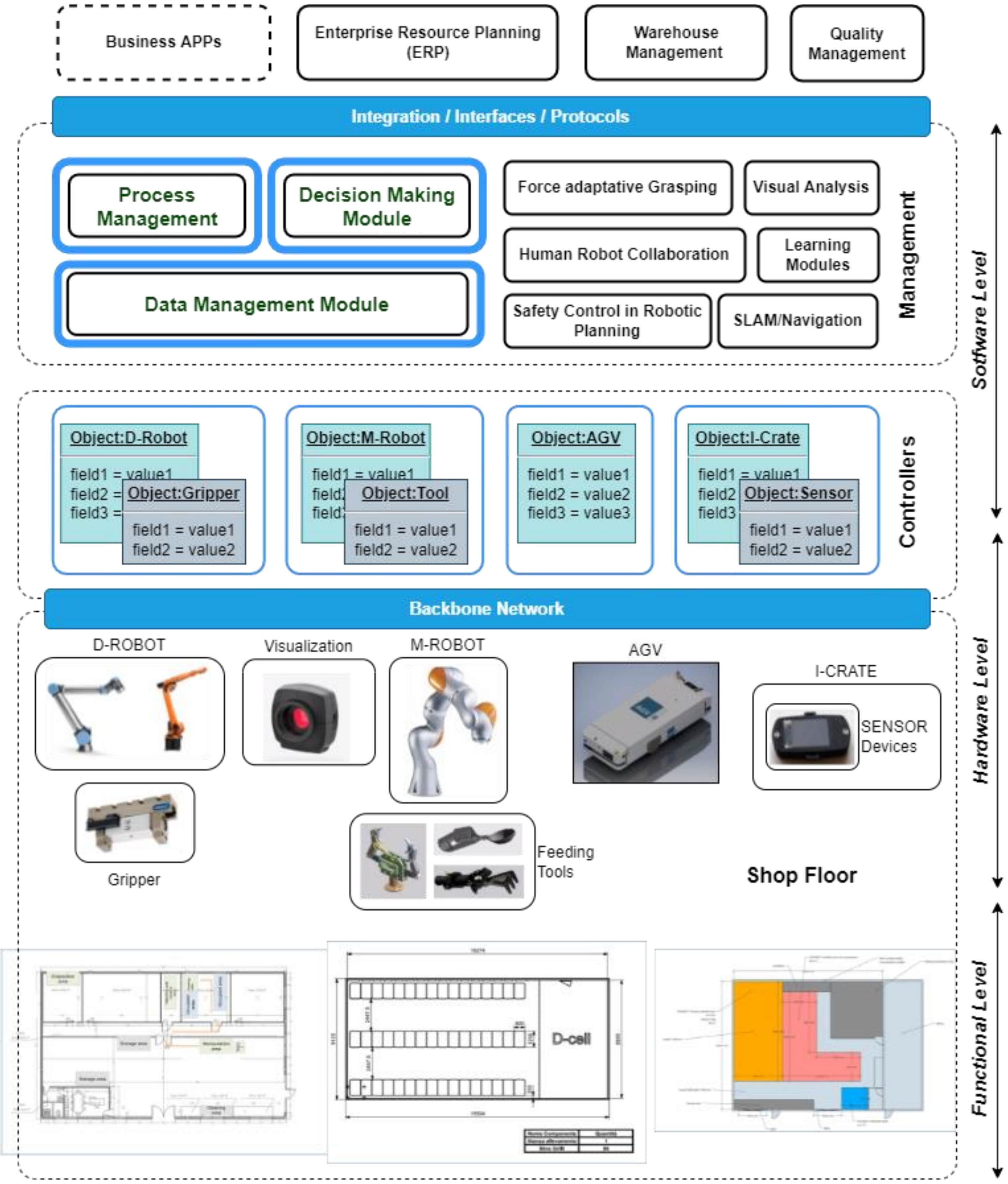


Figure 14. CoRoSect's overall usage scenario

3.2 Functional and Non-Functional Requirements

This chapter presents the main functional and non-functional requirements of the CoRoSect's Manufacturing Execution System (MES), which will support the factory **Process Management**, the **Decision-Making** actions and the whole **Data Management** activities. The list presented here is non-exhaustive because this is the first iteration of the deliverable and there are possibly going to be changes made to the overall system before finalizing these requirements. Moreover, the complete list is intended to be reported in an internal deliverable D2.5 which is at the disposal of the EU should it need to be further examined.

3.2.1 Functional requirements

The specifications of a product's functionalities are known as functional requirements (features). Simply, functional requirements specify what a piece of software must perform and how it must react to user input. The software's goals are defined by functional requirements, which means that if these criteria are not satisfied, the software will not work.

From the work done within Task 2.1 and Task 2.2 identifying and defining the use cases and processes to be addressed by the CoRoSect system, is derived an indicative list of the most important functional requirements for each MES component, which are classified as part of the Process Management, the Decision Making and the Data Management systems. These systems represent, from a holistic approach, the CoRoSect's architectural components that later in this document will support these functionalities:

Process Management System (PMS)

1. The PMS can receive trigger to start a certain process either from sensors or operator
2. The PMS receives a command from an operator to start executing a specific task
3. The PMS requests the respective list of tasks and its subtasks from the DMS
4. The PMS controls and executes the list of tasks
5. The PMS receives error messages from controllers
6. The PMS forwards error messages to DS in order to examine further actions/tasks needed to be executed.

Decision-Making System (DS)

1. DS has a Graphical User Interface (GUI) to show decisions made to the users.
2. DS provides the logic behind decisions made when comparing values, examining conditions within the system.
3. DS accepts values from other components and compares them to relevant values stored in the DMS.
4. DS instructs DMS to execute certain processes depending on the outcome of a condition, I.e., decision made.

Data Management (DMS)

1. The DMS stores data stemming from I-crates.
2. The DMS stores historical data produced by the cameras.
3. The DMS stores threshold values entered by an operator.
4. The DMS can provide the ID of the task and the status of the controller
5. The DMS stores the list of tasks and subtasks
6. DMS provides data to other components upon request
7. DMS provides data to other components in a Pub/Sub fashion

As mentioned before, the full list of functional requirements of the complete system will be presented in D2.5.

3.2.2 Non-functional requirements

Non-functional requirements specify how the system will carry out this function, whereas functional requirements establish the system's essential behaviour. Non-functional requirements, unlike functional requirements, do not comprise the system's backbone. That is, even if the non-functional requirements are not satisfied, the system will still function. Non-functional needs, on the other hand, should not be overlooked. Non-functional requirements are more user-oriented than functional requirements, which are largely focused on the client's demands. Some typical non-functional requirements are performance, scalability, capacity, availability, reliability, recoverability, maintainability, serviceability.

An indicative list of the most important non-functional requirements is provided below, mapped to individual MES components.

- **Auditability:** All CoRoSect MES components should record error logs in a well-formatted manner so that potential malfunctioning could be attributed to specific components and functions within them.
- **Deployability**: All MES components should be easily installed, deployed and configured.
- **Interoperability**: All MES components should be implemented providing interfaces to each other allowing them to communicate in order to complete a shared task.
- **Robustness:** All MES components should be implemented in a robust way allowing them to handle abnormal, erroneous inputs and operational conditions.
- **Reliability:** All MES components must be able to communicate and reliably process continuous data. Reliability is critical in most cases where Human-robot interaction and collaboration is taking place in terms of providing safety for both the human workers.
- **Performance:** All MES components should respond quickly to user actions and/or other components' triggers and output.
- **Replicability:** the whole CoRoSet system should be easily replicable according to the hardware and software conditions of the environment to be deployed. The hardware and software minimum requirements needed to run the system should be clearly specified and based on open standards.
- **Scalability:** the CoRoSect MES and backbone components (mainly software pieces designed to interconnect OT and IT levels) should be easily scalable to be adapted to wider infrastructures or to support an increase in demanding/demanded services.

The full and more concrete elaborations of non-functional requirements will be presented in D2.5.

3.3 Reference Architecture

Derived from the usage scenario (Figure 14) and the requirements identified in the previous section, Figure 15 proposes the CoRoSect Reference Architecture (RA) that aligns the building blocks of the CoRoSect System (introduced in the project's Document of Work) with the integration schema that will support the pilots, covering the CoRoSect's objectives. This diagram will set the basis for the CoRoSect system implementation, to be finally presented in D2.4 (Advanced System Architecture). Following subsections introduce each of its main components, linking them also with the corresponding CoRoSect Work Package where they are developed.

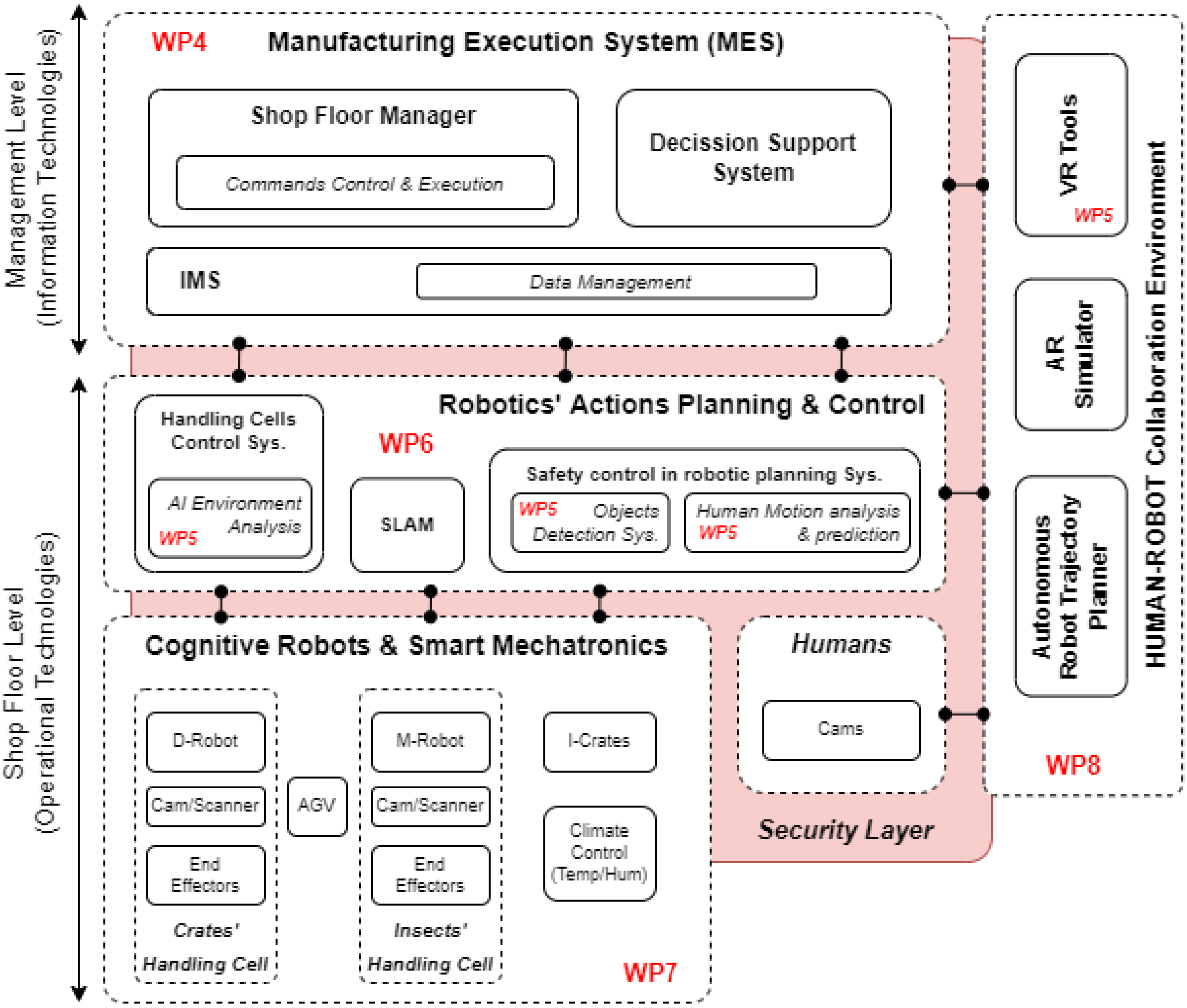


Figure 15. CoRoSect Reference Architecture

3.3.1 Cognitive Robots and Smart mechatronics

The CoRoSect system comprises three closely cooperating robotic systems with dedicated functions:

- The Crates Handling Cell is set up to handle heavy loads such as filled crates for insect rearing.
- The Insects Handling Cell focusses on fine manipulation of lightweight and small objects such as food trays and insects and supports the automated insect rearing process by handling material, feed, and insects
- The AGV autonomously transports stacks of crates from and to the robot cells.

Furthermore, the CoRoSect system contains intelligent crates for insect rearing, the I-Crates. These I-Crates are being formed by integrating smart embedded distributed sensor and processing nodes into crates. I-Crates communicate the selected sensor readings to the system. Insect and material handling is realised by the interaction of the dedicated robotic systems and distributed sensing.

3.3.1.1 Crates' Handling Cell

The Crates Handling Cell implements software and hardware for stacking and de-stacking of crates. The cell is protected by safety fences and light-barriers for safe operation.

D-Robot

At the core of the cell, the D-Robot is operating. The D-Robot is set up to manipulate heavy loads and is equipped with force-adaptive end-effectors for picking and placing crates.

Cams & Scanners

The Crates Handling Cell comprises cameras/scanners for detecting and for the safe manipulation of crates.

End Effectors

Custom-made force-adaptive end-efforts attached to the D-Robot ensure safe operation and handling of crates.

The Crates Handling Cell is provided with stacks of crates by the AGV. The Crates Handling Cell de-stacks crates from a pallet and places single or pairs of crates onto a support structure or conveyor belt to be further processed by the Insects Handling Cell. The Crates Handling Cell accepts processed crates from the Insects Handling Cell and stacks them again on a pallet to be transported by the AGV. The Crates Handling Cell is controlled by a dedicated Cell Controller (section 3.3.2.1).

3.3.1.2 Insects Handling Cell

The Insects Handling Cell implements software and hardware for manipulating insects and the material required for rearing insects. At the core of the cell, the manipulation robot (M-Robot) is operating. The M-Robot is equipped with high-resolution cameras/scanners for closed-loop control and custom-made end-effectors for precise, force-adaptive manipulation of insects and material.

M-Robot

The M-Robot is a collaborative robot capable of safe operation close to and in direct collaboration with human workers. The robot arm is equipped with high-resolution cameras/scanners, force sensors and custom-made end-effectors for precise and safe closed-loop manipulation.

Cams & Scanners

High-resolution cameras and scanners provide precise 3D information for the detection and identification of objects required for precise motion planning and safe closed-loop control for insect manipulation and safe material handling. Combined with artificial intelligence these cameras are used also in a variety of quality management tasks where the camera system autonomously assesses the well-being and growth of the insects.

End Effectors

CoRoSect develops and integrates custom-made end effectors for force-adaptive precise manipulation of insects and material. Overall, the Insects Handling Cell fulfils the following functionalities:

- Placement, replacement, and removal of support structures, food trays, and drinking devices into and from crates for insect rearing.
- Placement and removal of insects.
- Camera guidance for quality control of the insect rearing process.
- Feeding.
- Waste removal.
- Manipulation of crates.

The Insects Handling Cell is provided with crates by the Crates Handling Cell. Both cells collaborate closely by delivering crates between each other. The Insects Handling Cell is controlled by a dedicated Cell Controller (section 3.3.2.1).

3.3.1.3 AGV

The under crawler AGV (Autonomous Guided Vehicle) is capable of autonomous navigation and carries stacks of crates and containers with material on pallets autonomously between storage and insect rearing areas and the Crates and Insects Handling Cells. The AGV is equipped with high-resolution scanners for autonomous navigation and simultaneous localization and mapping (SLAM, section 3.3.2.3).

3.3.1.4 I-Crates

I-Crates are intelligent crates for insect rearing. I-Crates contain embedded sensor and processing nodes that can obtain and transfer sensor data to the CoRoSect MES through a dedicated gateway. This sensor data is crucial for continuous monitoring of the insect rearing process and for continuous quality management.

3.3.1.5 Climate Control

Automated climate control is provided by the building of the insect farms. Climate control plays a crucial role in the well-being and efficient growth of insects during the rearing process.

3.3.2 Action Planning and Control

This section includes subsystems related to the action planning and control for the insect farm. This includes the handing of crates (addressed in sections 3.3.2.1 and 3.3.2.2), the Simultaneous Localization and Mapping (SLAM) (addressed in section 3.3.2.3) and the safety control to avoid collisions between people and robots (addressed in section 3.3.2.4).

3.3.2.1 Handling Cells Controllers

All Crates Handling Cell and Insects Handling Cell are controlled by dedicated cell controllers. These cell controllers directly communicate with the CoRoSect MES through the IMS. The cell controllers fulfil the following main operations:

- Obtaining sensor data from the corresponding robot cells, AGV, and I-Crates.
- Transferring data to the IMS.
- Requesting/receiving data from the IMS (when needed)
- Receiving and executing instructions from the IMS.

The handling cell controllers in particular:

- Obtain and provide environmental data from their dedicated cameras/scanners and embedded sensors.
- Obtain and provide data from the sensors integrated into the M-Robot and D-Robot including their state of operation, position data of joints and end-effectors, data from force sensors embedded in the robots (M-Robot) and end effectors (D-Robot and M-Robot).
- Process data for the detection of crates (Crates Handling Cell, Insects Handling Cell), insects and material for insect rearing (Insects Handling Cell).
- Obtain instructions for operation from the IMS and execute these operations autonomously.
- Handle data from cameras/scanners and force-feedback sensors for autonomous and safe operation.
- Plan and execute actions with the dedicated M-Robot and D-Robot.
- Detect and collaborate with human co-workers (Insects Handling Cell).

The Crates Handling Cell and Insects Handling Cell controllers implement motion planners for the autonomous and collision-free operation of the robot arms (D-Robot and M-Robot).

3.3.2.2 I-Crates Controller

I-crates integrated sensors created data is read through the I-Crate controller. I-Crate controller consist of dedicated gateway + included client software) and it directly communicates the I-Crate provided data with CoRoSect MES through the IMS.

I-Crate controller is responsible for the following operations:

- Reading the sensor data sent by an integrated intelligent sensor (IIS) in the I-Crates
- Sending the data to IMS via its software client
- Receiving and executing instructions from IMS (in certain cases)

3.3.2.3 SLAM

The AGV is equipped with its own safety control system. This system consists out of a safety PLC and safety LiDAR scanners. The safety control system is isolated from the AGV control software and will always be active. Although the safety functionalities are isolated, the raw data of the safety LiDAR scanners can be shared and used for localization.

Contour localization and navigation is used during the operational phase of the AGV. For this type of localization and navigation, the raw data of the safety scanners is matched with a map of the contour. This map is stored on the AGV.

To build the map in the commissioning phase, Simultaneous Localization And Mapping (SLAM) [15] on the AGV is used.

During the SLAM operation, the AGV will autonomously explore the environment. By driving around and processing the raw LiDAR data, a map of the environment is created. The complete operation is performed on the AGV.

This map will be the base of the coordinate system of the AGV and can be shared with other systems.

3.3.2.4 Safety Control

The safety control is composed by three different components that are complementary: a) the obstacles' detector; b) the routes manager; and c) the routes planner.

The last two components (manager and planner) have different functionalities, but, as they both directly deal with routes, in a future implementation they may be included in the same software component.

The Safety control covers the following requirements:

- To manage the routes of all moving elements of the factory
- To avoid any possible collision between robots and people
- To avoid, if possible, stopping the moving robots.

This Safety Control component will never override or substitute current safety measures of the robots. It will "extend" these safety measures in order to predict possible collisions and to avoid stopping the AGVs whilst ensuring the safety of people (primarily) and of the machines.

Detector of obstacles

The detector of obstacles will be thoroughly described in D6.7 (M12), D6.8 (M24) and D6.9 (M30) *Safety concept for robotic systems.*

The component has the objective of detecting any static or moving "element" that are present on the factory floor and that are not part of the factory baseline. These "elements" include all kind of objects, including robots, and people. The idea is to detect any potential obstacle that is present in the passing areas (the aisles of the factory). The system detects these “obstacles” and in the case of moving elements, the projection of their trajectory, and then sends the detection to the Routes manager.

The detection is made using fixed IP cameras (as seen in Figure 16) that cover the entire factory avoiding “blind” areas. The footage detected by these cameras is analysed in near-real-time to detect:

- People and people movement: the algorithms calculate the position of the people in the factory space and project their possible movement
- Any moving element and the projection of their movement: this includes any moving AGV for example
- Any static obstacle that is not part of the factory “baseline”: This includes any object that is blocking an aisle, for example, a fallen box or crate.

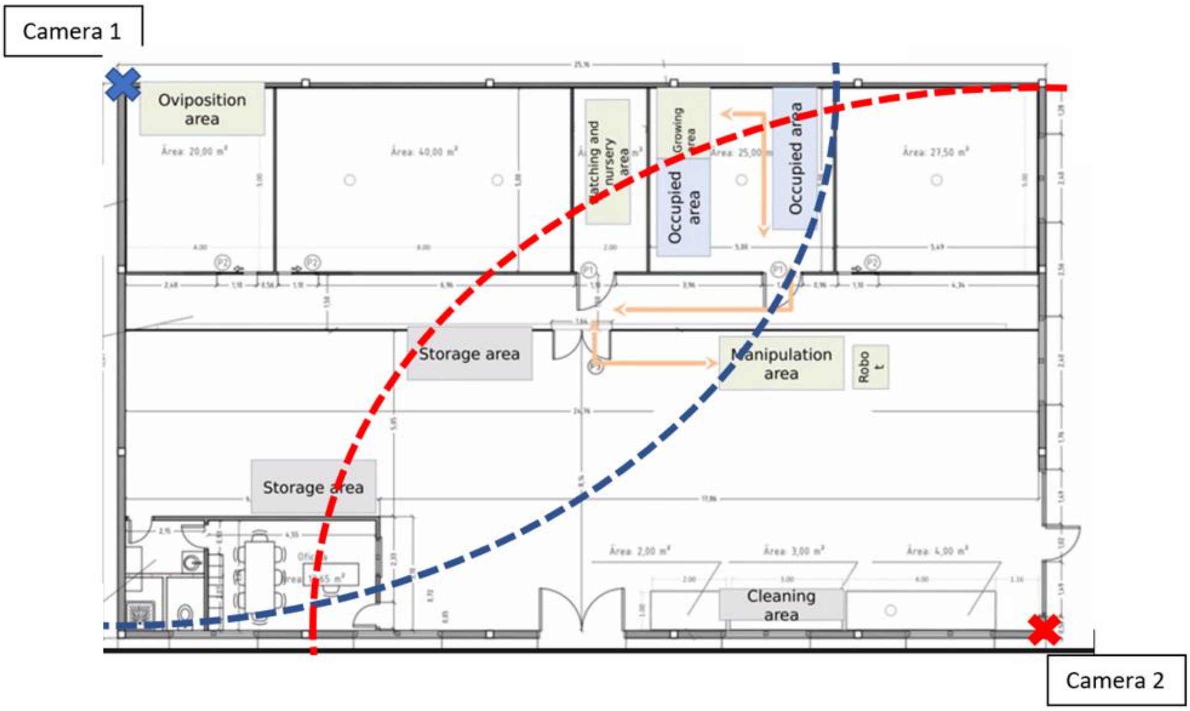


Figure 16. Detection of obstacles, example of camera layout

After the analysis, the obstacles detected are sent to be consumed by the Routes manager.

Routes’ manager

The Routes manager is the central piece of the Safety control. This component is responsible for checking all the routes of all the moving robots in order to avoid any collision with other robots, unexpected obstacles and people.

The component will receive information about a) Routes from the Routes planner; b) Position of all moving robots (AGVs); and c) Obstacles detected and the projection of the “moving” obstacles from the Detector of obstacles

The objective of the manager will be to check if the routes and positions can create any future collision. If a potential collision is detected, an order of “reroute” will be sent to the Routes planner.

This component has two main missions:

- Validation of Routes: The Manager will receive any new route from the Routes planner for any AGV. The manager will check the planned route against all the other known routes and obstacles to detect any problem. As a response, the manager will validate or reject the new route. In the case of rejection, the Routes planner will send a new route to restart the validation process.
- Dynamic detection of obstacles: the manager will receive updated information on the current position of all moving elements and information about any static or moving obstacle and its future projection. With this information, the Manager will be able to detect any possible future collision. As a response, it will command reroute orders to the Routes planner for any AGV.

Routes' planner

The Routes planner will be aware of the factory layout, that is the aisles where the AVGs can circulate. It will create optimal or sub-optimal routes for any AGV between two or many points.

The planner will create a route and will send it to the Route manager to validate it. In case of rejection, it will create a new route. In the case of receiving a reroute order, it will create a new route (different to the previous one) and will send it to the Route manager in order to validate it.

If a route is declared valid, it will send to the specific AGV to be followed.

3.3.3 Manufacturing Execution System (MES)

Industry 4.0 (I4.0) relies on cyber-physical production components that connect the physical and digital world of production and proposes a fully digitized system that covers all levels (devices, controllers, managers and information systems) within a factory. CoRoSect's Objective 1 (O1) aims at an I4.0 compliant system that will require digital transformations and reorganizations of targeted factories' production methodologies. The I4.0 Maturity model [16] provides a guide for companies to implement this required digital transformation process. It comprises a six-stage maturity model focused on the four key structural areas of resources, information systems, organisational structure and culture:

1. Computerization: directly related to digitalisation, different information technologies are used here, in principle, isolated, to assist on different production processes.
2. Connectivity: the isolated deployment of information technology is replaced by connected components, but Information Technology (IT) and Operational Technology (OT) layers are not still fully integrated.
3. Visibility: production chain events and states are recorded in real-time throughout the entire company. Digital models of factory's components are created to keep, update and share information.
4. Transparency: the company is able to understand why something is happening and uses this understanding to produce knowledge using root cause analyses.
5. Predictive capacity: company can simulate different future scenarios and identify the most likely ones. This is a fundamental requirement for automated actions and automated decision making.
6. Adaptability: the company is able to delegate certain decisions to IT systems so that it can adapt to a changing business environment as quickly as possible.

The main challenge to implement Industry 4.0 is to put these principles into practice, which will allow generating knowledge from data, enabling rapid decision-making and adaptation processes throughout every part of the company.

The Manufacturing Execution System (MES) is defined within I4.0 as the component that monitors, controls and optimizes the manufacturing processes [17]. Information managed by the MES is used by decision-makers to understand how the subsystems involved in production are interlinked, and the generated knowledge enable the continuous improvement of factory processes. In this sense, MES is the key component of stages 2 and 3, supporting then stages 4, 5 and 6 of the I4.0 Maturity model. According to this, the basic MES functional requirements [18] would guarantee:

- **Support computerization** and digitalization: use computer-based control throughout the whole production chain
- **Improve connectivity**: support information sources that automatically sends information about themselves to the MES in real-time. The overall state of the whole production chain should be monitored and so the traceability of the products
- **Ensure visibility**: process, share and show information about the current status of the production chain. Interconnect with Product lifecycle management (PLM) systems and Enterprise Resource Planning (ERP) systems.
- **Ensure transparency**: homogenise, store and distribute product chain information to feed knowledge data bases and big data tools oriented to process' improvement.
- **Increase predictive capacity**: feeding CoRoSect simulation and optimization functionalities and tools, fostering the virtual factory concept and the use of AI for process control (Digital Twin concept).
- **Improve adaptability**: enable the usage of (almost real-time) data to support adaptative real-time decision making.

This way, MES links the company's planning system (like its ERP) with the controlling systems (such as sensors, robots, etc.), using the manufacturing information (such as devices' status, resources, orders and commands) to support the manufacturing processes. CoRoSect's MES is being developed by WP4 (Farm-level modelling and orchestration) and its performance will be driven by WP9 (Secure platform integration). It will be split into three main subsystems that address the presented requirements.

3.3.3.1 Information Management System

All the information produced within the OT layer will be driven to the IT layer, and vice versa, through the Information Management System (IMS). This component will act as the node that serves the information required to manage the manufacturing processes, also providing the connection with the shop floor controllers. It will support these main functionalities related to data management:

- Data modelling: it will provide data models to map the entities (devices, robots, processes, tasks, etc.) including identified attributes based on current IoT and I4.0 standards. This way, all the information managed by the MES will be homogenous whilst facilitating the integration with new information systems or I4.0 compliant devices.
- Data gathering and storing: it will classify and store all the information gathered from OT layer, using the data models, to be served on-demand to the IT layer.
- Data distribution: it will implement mechanisms to serve all the required information (from the OT layer and/or introduced by an operator, such as process definitions, configurations, or tasks) to the demanding systems in two main ways, a) in an on-demand fashion, implementing a query/retrieve API; and b) in a publish/subscribe fashion, providing information in close-to real-time.

CoRoSect's IMS is developed within Task 4.3 Service-Oriented Information Management System.

3.3.3.2 Decision Support System

CoRoSect's Decision Support System (DSS) will support SFM in carrying out the task by taking all the decisions such as matching real-time values with the values stored from process knowledge. The DSS is subscribed to the IMS so it is permanently aware of the status of the production chain. Task 4.2 (Shop Floor Management and decision support system) is defining the context to be monitored, the actions it should automatically trigger and the processes it should supervise.

3.3.3.3 Shop Floor Manager

The CoRoSect's Shop Floor Manager (SFM) monitors the robot devices, storage, inventory, and manufacturing processes by connecting to the IMS. With this information, it will manage:

- the proper routing of materials on the shop floor
- the processes and procedures on the shop floor
- the scheduling of the required resources (materials, workforce, etc.), and operations
- the monitoring of shop floor status to detect deviations from planned processes, to correct them swiftly

DSS and SFM will be also connected, so the SFM can request DSS evaluation to correct a given deviation and the DSS can act on a running process in case a specific event is detected. Together with the DSS, the SFM is being developed by Task 4.2 (Shop Floor Management and decision support system).

3.3.4 Human-Robot collaboration environment

The factory space has to be shared between robots and people avoiding any collision while keeping the robots moving autonomously as much as possible. This section includes subsystems aimed to enhance this collaboration. This includes the creation of an autonomous robot trajectory planner (section 3.3.4.1), the AR simulator (section 3.3.4.2) and the VR tools (section 3.3.4.3).

3.3.4.1 Autonomous Robot trajectory Planner

As described in section 3.3.2.3 we have a Trajectory planner that is part of the Safety control component.

The trajectory planner needs as a baseline to have a "map" of the insect farm. This map can be obtained in different ways; for example, can be passed as a 3D or 2D map to the tool. In CoRoSect the idea is to obtain this map using SLAM algorithms or a 3D projection using other tools/components and then pass it to the Trajectory planner as a "static" element. We don´t need to send the map to the tool, but it can be part of the tool from the beginning (Figure 17)

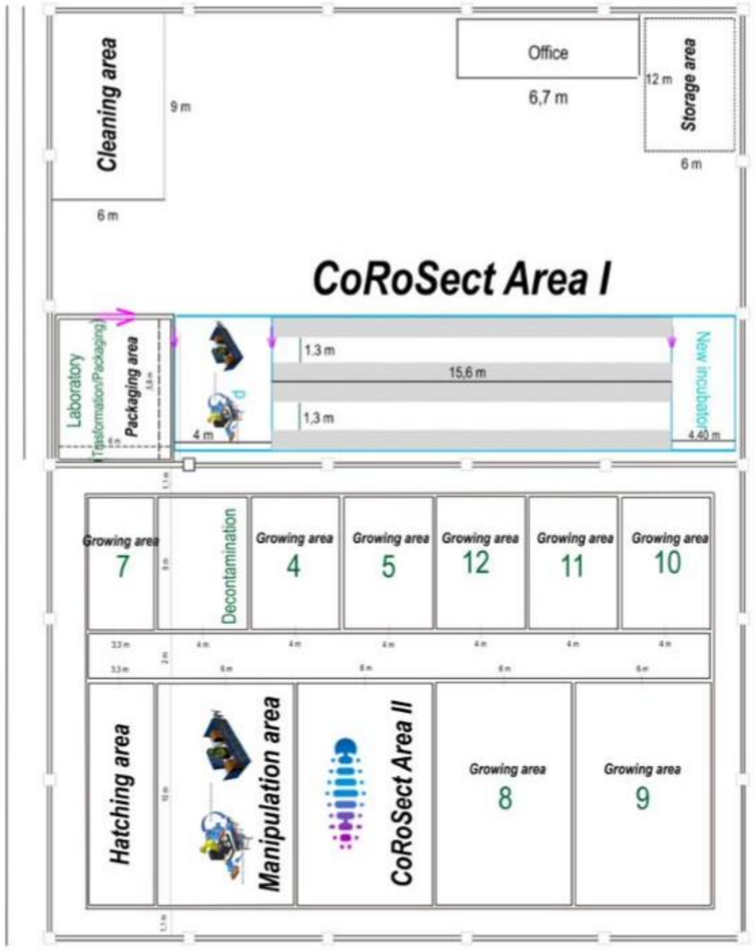


Figure 17. 2D plan of a insect farm

With this map, the tool is aware of the fixed elements of the factory and the passing areas (the aisles) and also of the coordinates of all elements.

The tool implements two main data fluxes:

- CreateRoute (a,b): [”a” and ”b” are points] As a response, the tool will create an optimal or suboptimal route from point ”a” to point ”b”. This route will be, then, sent to Routes Manager to be validated. If the route is rejected, it will create a new route. This process will be repeated until a valid route is created or the process is cancelled.
- CreateRoute (a,b,c,d...): this is similar to the previous one using many waypoint points.

3.3.4.2 AR Simulator

For the CoRoSect system, a safe human-robot collaboration environment is needed to ensure the safety of the workers and the robots. To achieve the environment the workers situational and spatial awareness will be enhanced using Microsoft's HoloLens 2. It is an optical see-through head-mounted device on which the necessary AR-based mechanisms for human-machine interactions for situation awareness will be realized. To create this environment, a multi-level AR-based worker-robot communication system is being developed that provides the following functionalities to the worker:

1. Worker with indications of the robot's next step

2. Status updates about the ongoing task and warnings
3. Farm floor task assignments

Regarding the first functionality, as can be seen in Figure 18, the robot follows a 3D augmented reality line that represents the trajectory created by the planning of the robot's movement. The colour of the line changes according to the distance of the HoloLens 2 user from the line itself. If the distance between the user and the line becomes less than a predefined value its colour changes to red, as seen in Figure 19, in order to inform the user that he is positioned too close to the path that the robot will follow.

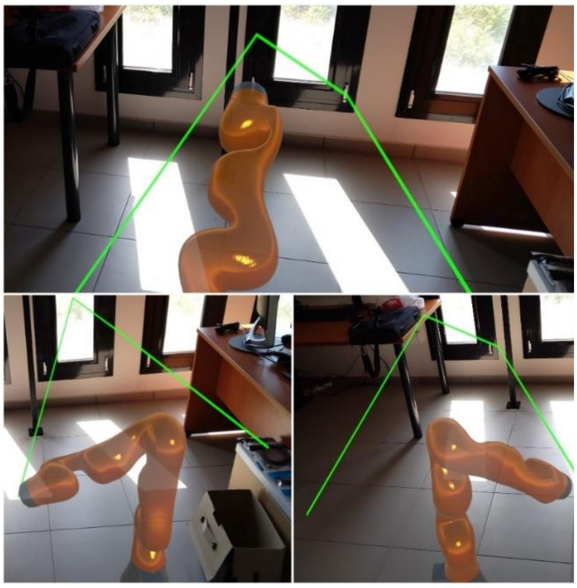


Figure 18. The robot is moving along the green line.

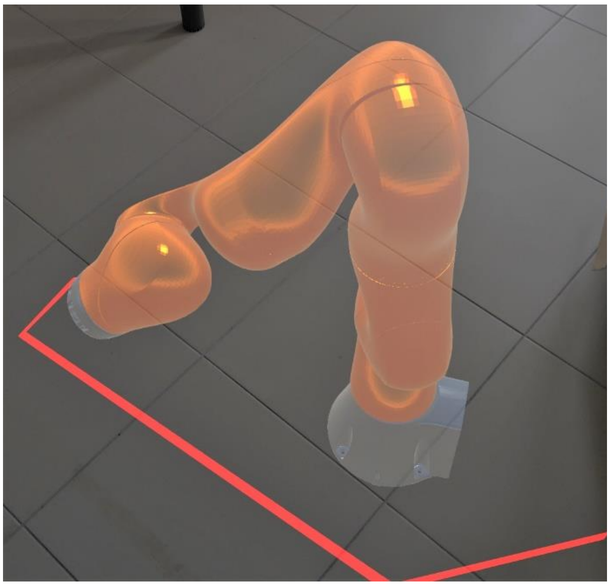


Figure 19. Red trajectory based on the user's distance.

Regarding the second functionality, the 3D AR augmented sphere around the robot, as depicted in Figure 20, acts as a safety border that signifies the range of movement of the robot. As before, the colour of the sphere changes according to the distance between the sphere and the HoloLens 2 user. Moreover, an augmented-based warning message, as can be seen in Figure 21, appears to the user's interface to inform that he is positioned too close to the robot or the robot's movement path.

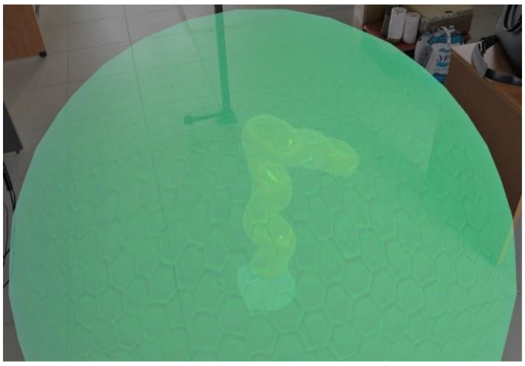


Figure 20. 3D Green Sphere around robot.

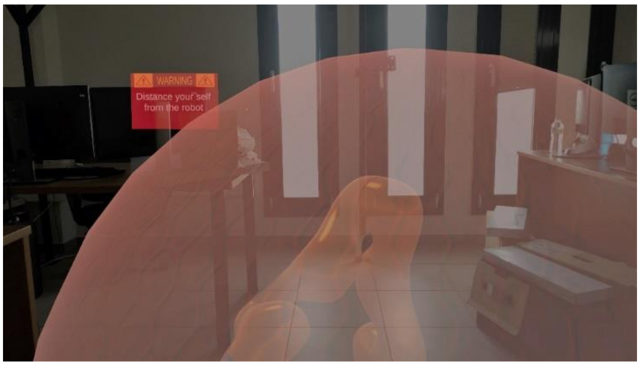


Figure 21. Warning user regarding the distance.

The third functionality is another essential point, the ability provided to the user to communicate with the robot through menu options in the interface of the HoloLens 2, as can be seen in Figure 22. For instance, by pressing augmented reality buttons the user will send a signal to the robot, and accordingly, to the signal, the robot will carry out a predefined command.

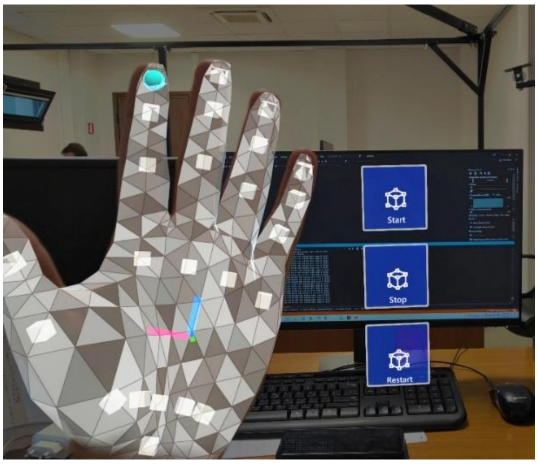


Figure 22. Hand menu with palm facing up.

3.3.4.3 VR Tools

In the context of CoRoSect, VR modalities will be explored, especially the use of wearable devices that offer force and haptic feedback. For this purpose, a physics-based VR environment needs to be developed in the Unity game engine. This environment will be used to facilitate the onboarding of new workers and their training at the end-users since it will represent a similar environment to the one that can be found at the end-users' facilities. In order to develop the aforementioned environment, a wearable device that fulfilled all the requirements was chosen, the SenseGloves, as can be seen in Figure 23. The SenseGloves have the ability to provide both force and haptic feedback (Figure 24). Regarding the haptic feedback, the SenseGloves user will be able to have the feel of touch, while interacting with digital objects within the VR environment. Regarding the force feedback, the user will feel realistic forces on the phalanges, when interacting with the digital objects, as if interacting with physical ones.

Figure 23. SenseGlove

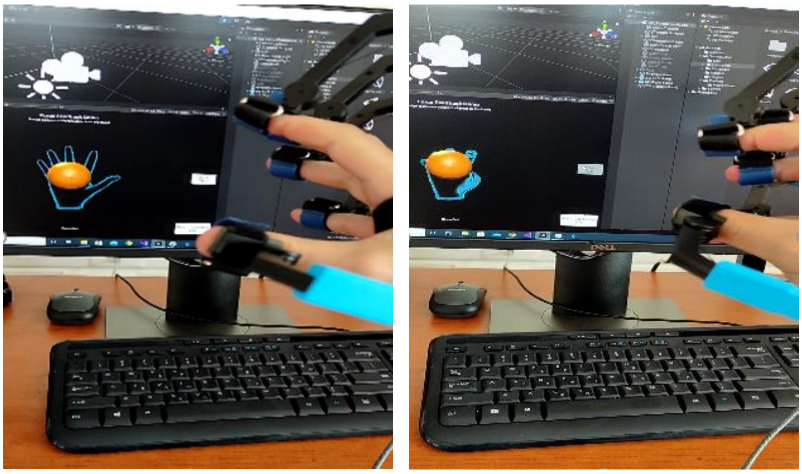


Figure 24. Force and haptic feedback with deformable object

4 CoRoSect System Architecture

CoRoSect's architecture is defined to deploy a whole cyber-physical and integrated system intended to support the functionalities addressed to a general insects' farm production process. It is guided by the Industry 4.0 standards for the digitalization of the robotic components and processes but also considers merging the growing IoT technologies and infrastructures to combine these two environments into a RAMI4.0 compliant system that integrates factory robots and IoT sensors. This section takes the requirements and the reference architecture presented in section 3 of this document to expand this approach and to present a first version of the different components that compose the baseline for the CoRoSect System implementation, which will later support the development of CoRoSect novel subsystems and assets carried out within each corresponding Work Package. This system is divided into layers demarcated by functionalities, operational areas, digitization level and services offered. The architecture is so aligned with the CoRoSect RA (Figure 15) and the identified functional areas, making easier the assignment of defined requirements and functionalities.

4.1 Logical View

CoRoSect's architecture logical view presents a high-level schema of all the involved components and their existing relationships in terms of incomes and outcomes (dataflows). Starting from the CoRoSect RA derived from the project's Description of Work (Figure 13), the logical view identifies the set of overall components detailed in Figure 25.

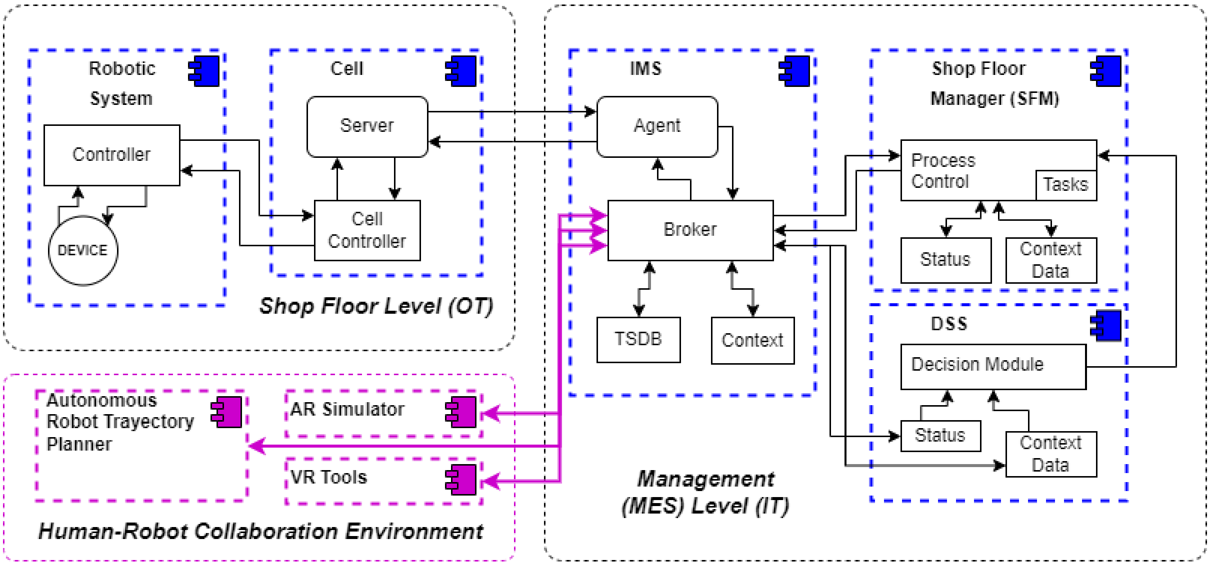


Figure 25. CoRoSect's System Logical View

From this logical perspective, and considering only the functionalities and objectives that the CoRoSect project pursues (this is, leaving the Enterprise Resource Planning – ERP - level out of the scope), there are three principal levels grouping components: i) the Shop Floor Level, where the robots, mechanical components and IoT devices will perform the production actions; ii) the Management level, where all data is stored and processed and from where production processes are controlled and quick decisions are taken; and iii) the Human-Robot collaboration level, where CoRoSect's AI-assisted planners and enhancement tools for the safe collaboration between workers and devices are done. These logical levels are depicted in the following subsections.

4.1.1 Shop Floor Level

Within the CoRoSect system and in CoRoSect pilots, the Shop Floor level covers the area where the insect's production and rearing activities are carried out, involving an automated system, workers and combination of both. From the architecture point of view, we are considering here the robotic systems and devices (what is called OT-Operational Technologies) to be integrated within the proposed CoRoSect system (and which will be involved in the pilots) but from a holistic perspective that allow expanding this logic to other automated systems. This way, all Shop Floor robotic components (AGV, M-Robot and corresponding end-effectors, D-Robot and corresponding end-effectors, I-Crates and Visualization system) will fit on the logic proposed in Figure 26.

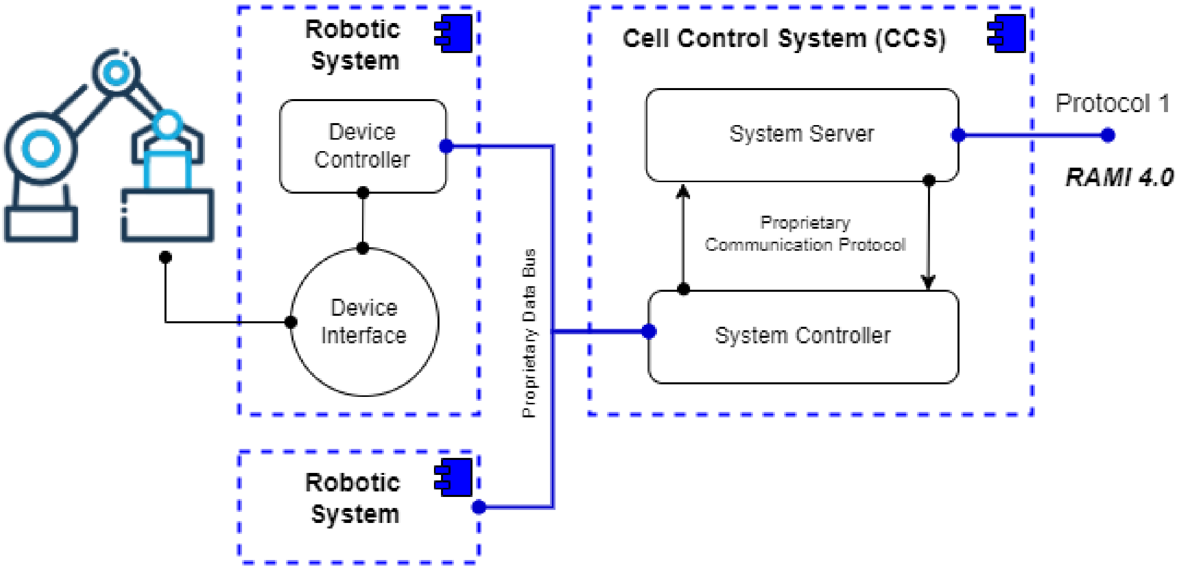


Figure 26. Shop Floor's general component logical view

Each integrated Shop Floor component will consist of two building blocks:

- **Robotic System**: usually a proprietary software that interprets the instructions addressed from the MES through the Cell Control System and executes them on the end device. It also captures raw data from the end device and send it to the Cell Control System. Its Device Interface directly connects the end-device with the Device Controller, which processes the information and manages the data flow.
- **Cell Control System**: connects and integrates the Robotic System within the CoRoSect main system. Its main functionality is to standardise the commands and data flows according RAMI4.0, supporting the digitalisation of the managed devices. The system controller connects with the end device through its specified data bus and translates the RAMI4.0 compliant protocols (commands and query/retrieve instructions) into end-device protocols and vice versa. The System Server exposes the robotic system to the CoRoSect environment, enabling its integration and its digitization. A unique Cell Control System may map several robotic systems, as they become an operative part of the same "Cell" as a subsystem.

Corresponding components will be developed in WP6 and WP7.

4.1.2 Management Level

This level covers the main IT (Information Technologies) that controls the processes to be carried out within the Shop Floor (by the OT layer). The CoRoSect perspective corresponds to the implementation of the I4.0 Manufacturing Execution System (MES) introduced in section 3.3.3.

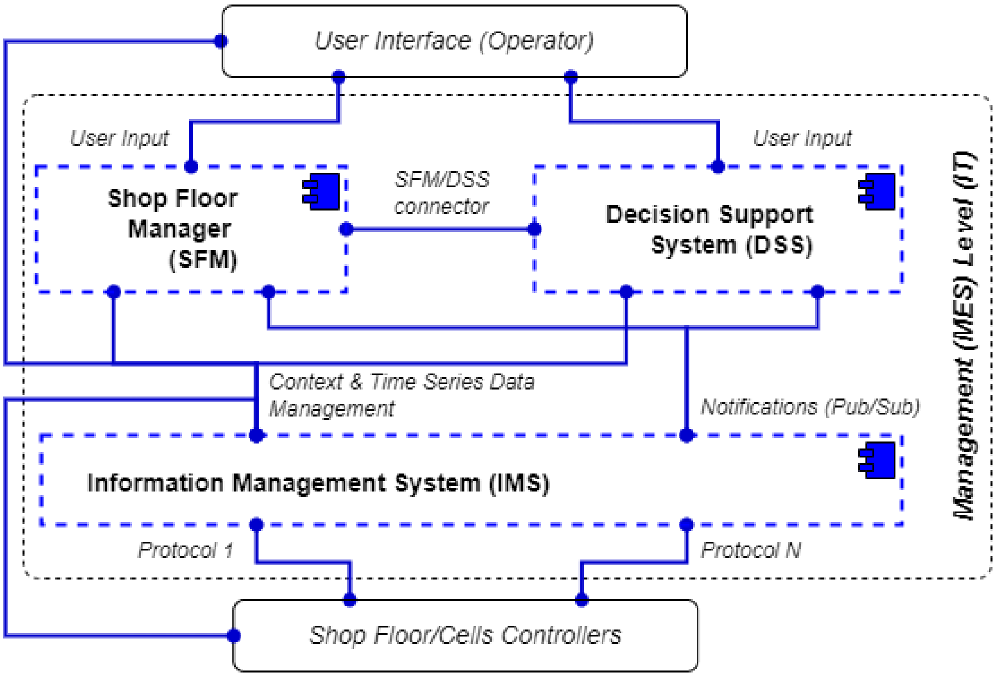


Figure 27. Management Execution System (MES) Logical view

The MES (Figure 27) supports the process management, the decision-making operations and the data management activities introduced in the functional requirements (section 3.2.2) by implementing the IMS, the SFM and the DSS.

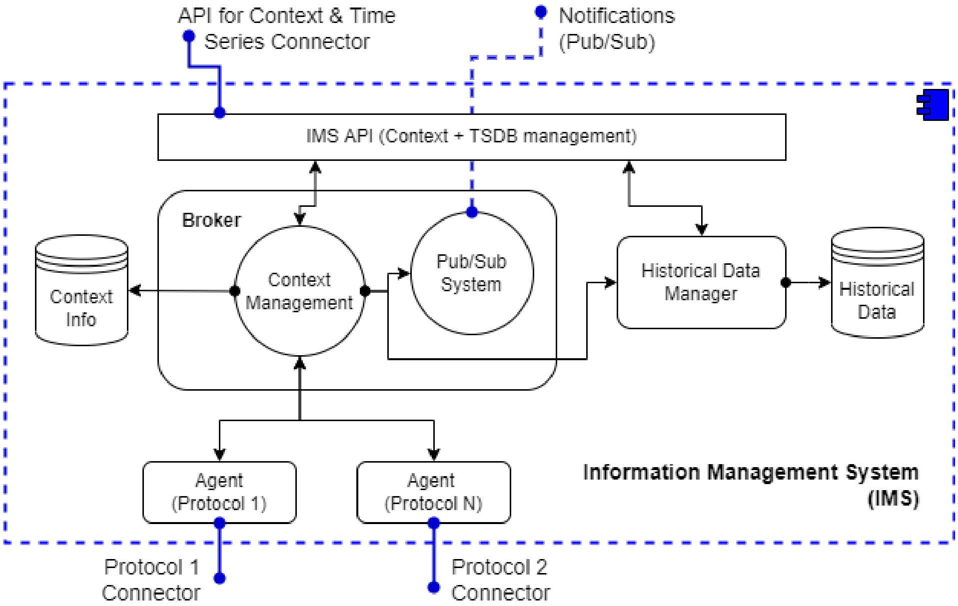


Figure 28. Information Management System (IMS) logical view

The Information Management System (IMS), shown in Figure 28, supports all data management and interconnection functionalities, storing and serving all data required to perform the production processes. It also manages the Digital Twins, as digitalized information models, of all system components. This is to be developed in T4.3 and It contains:

- A context broker that serves context information to all the CoRoSect's subsystems. Context Information provides the status, with last measurements and reports from all components, through their corresponding information models. This information models also provide the links to manage commands with their digitized devices. These models will also be used to map and manage processes and tasks to be managed by the SFM. This broker will support data query/retrieve and publish/subscribe APIs to provide context information on demand and in an asynchronous way.
- A historical data manager, that receives all reports from system components and keeps classified historical records for these measurements. It provides an API to quickly serve time series datasets with information from requested devices.
- Data repositories for Context Information and historical datasets, core of the system's data persistence layer.
- In order to get connected with the Shop Floor, it also implements data adaptors to integrate the system with the Robotic Systems.

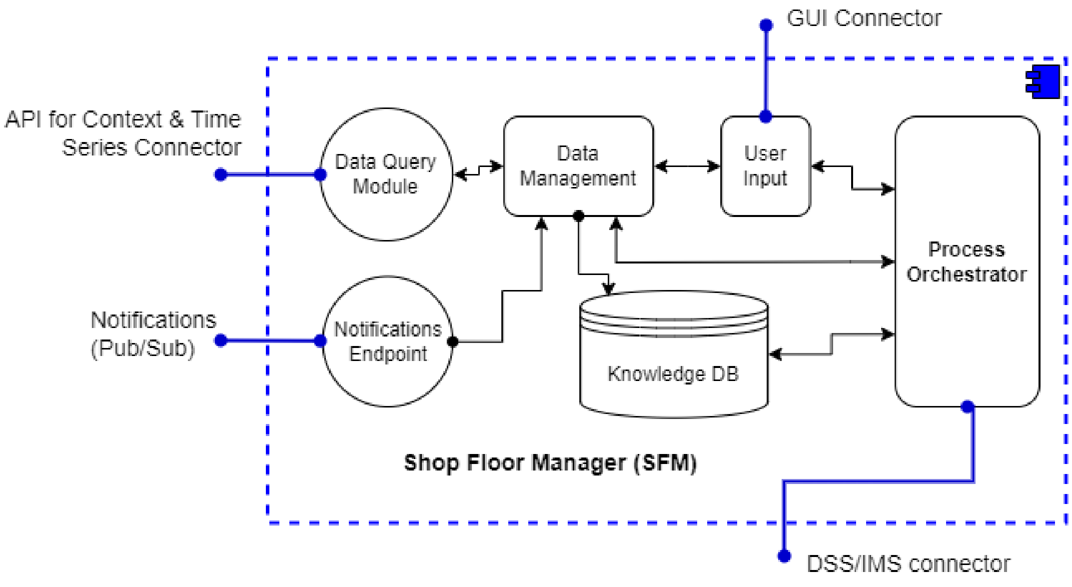


Figure 29. Shop Floor Manager (SFM) logical view

The Shop Floor Manager (SFM) is the CoRoSect System orchestrator. This is part of the T4.2 objectives and will support the process management functionalities (Figure 29). It is built by:

- A set of end-points and modules that connects the SFM with the rest of the CoRoSect System: i) a Notifications end-point to connect with the IMS Pub/Sub service and receive all data updates as they happen; ii) a Data Query module to connect with the IMS query/retrieve API and request information on-demand; iii) a link with the DSS to receive further instructions, and iv) a GUI connector to get inputs and provide outputs to an operator.
- An internal SFM data management module to pre-process received notifications and retrieved data and serves this to the process orchestrator core. It also builds the query commands to be sent to the IMS and stores data into the SFM knowledge data base.
- A Knowledge data base that stores relevant information used for the Process Orchestrator to enhance efficiency when managing production processes.
- The process orchestrator core that loads the production processes to be executed (from the IMS), collects all required information, connects with the DSS and triggers the corresponding

tasks in the corresponding devices. Through the IMS, it also monitors the status of the Shop Floor and the running processes execution.

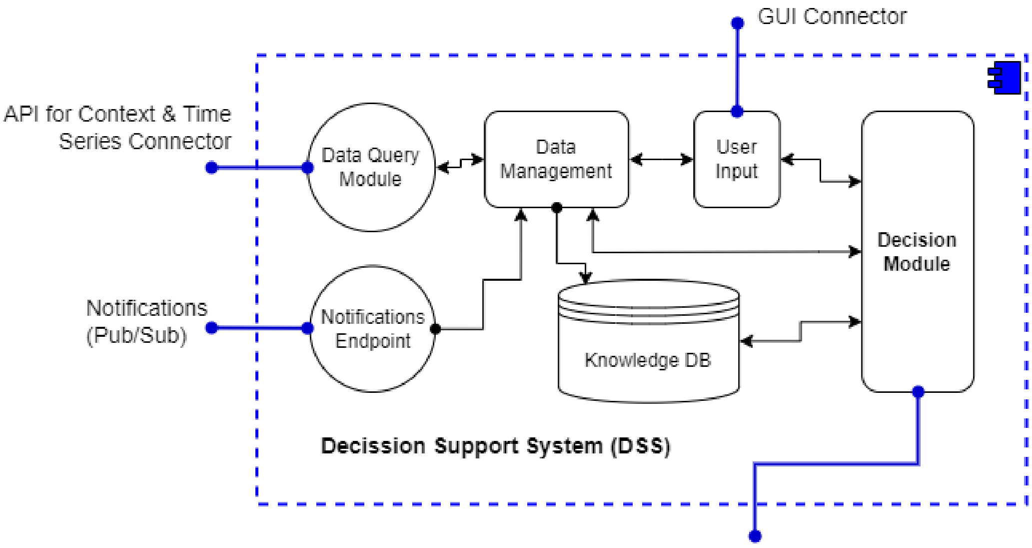


Figure 30. Decision Support System (DSS) logical view

The Decision Support System (DSS) shown in Figure 30, and also to be developed by Task 4.2, will implement the decision-making layer to be addressed by CoRoSect System which will be defined by its WP4. Its internal logic composition is similar to the SFM by connecting and interacting with the same set of subsystems and it will consist of:

- A set of end-points and modules to interconnect the DSS: i) a Notifications end-point to connect with the IMS Pub/Sub service (asynchronous data update); ii) a Data Query module to connect with the IMS query/retrieve API and request information on-demand; iii) a link with the SFM to interact with the running processes, and iv) a GUI connector to get inputs and provide outputs to an operator.
- An internal DSS data management module to pre-process received notifications and retrieved data and serves this to the Decision module. It also builds the query commands to be sent to the IMS and stores data into the internal DSS knowledge data base.
- A Knowledge data base that stores relevant information for the decision-making process.
- The Decision Module combines and process the current Shop Floor status, the historical datasets and its knowledge with the user inputs to assist the operator on the process control and to interact with the SFM to modify a process execution.

4.1.3 Human-Robot collaboration level

This level houses those CoRoSect components that are not essential to the proper production process performance but represents the innovations that CoRoSect brings to the Smart Industry 4.0 insect farming. These have two main objectives: a) focused on safety, to build an enhanced human-aware collaborative environment, where mechatronics and human workers can safely and effectively cooperate; and b) focused on learning, provide tools for humans to operate and train with robotic systems and ML/DL powered algorithms to let robotic systems learn from human behaviour. These components will be connected to the CoRoSect MES to gather information from the Shop Floor and to interact with the SFM when needed. The developments related to this level will be addressed in WP8 and WP5 respectively and will consist of the following components.

Autonomous Robot Trajectory Planner, which receives its inputs via publication/subscription using the Notification Endpoint; these inputs are commands/orders to calculate routes (new route or rerouting) including two or more points (waypoints or coordinates). The Route Calculation calculates a new route taking into account the Factory Map and checking with internal memories of Previous Routes that is not a route already calculated (for the case of re-routing). After this, the planer checks the route validity via request/response (it needs an answer). If the route is valid it is published.

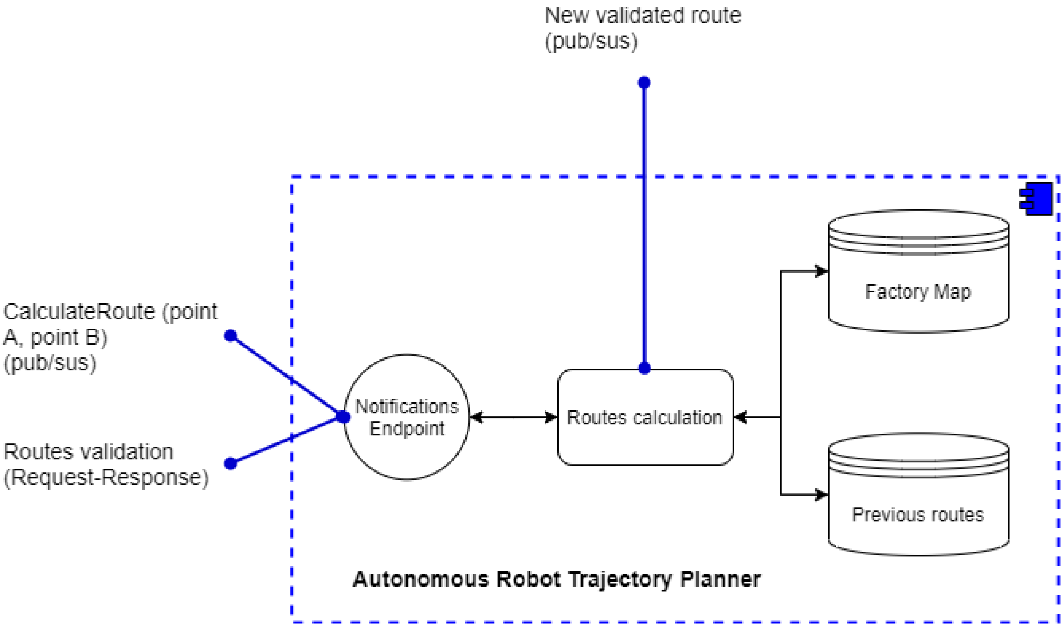


Figure 31. Autonomous Robot Trajectory planner logical view

It is composed of the following modules (Figure 31):

- Routes' calculation: calculates a new route between points. Request route validation. Output is the new validated route.
- Notification endpoint: receives inputs of new routes and validation of routes.
- Factory map: static map of the factory. Used to calculate the new route.
- Previous routes: temporal storage of "routes under use" used to avoid repetition in case of a petition of "re-route".

AR Visualizer, which receives input through the MES and also transmits output to the MES utilising ROS Interface; these inputs are transferred via the ROS Network to the Subscriber/Publisher and are used to visualize synthetic data in the Device Interface. The Device Interface provides input to the operator and likewise receives input from the operator. Through the User Interface, the operator's output is used as input the Subscriber/Publisher and via the ROS Network is transferred to the ROS Interface and subsequently to the MES.

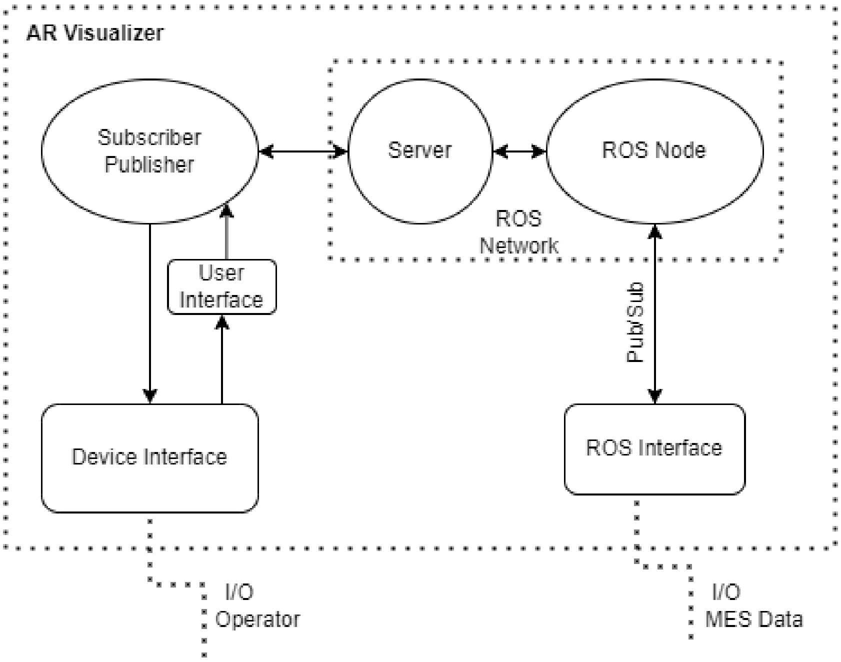


Figure 32. Augmented Reality Visualizer logical view

It is composed of the following modules (Figure 32):

- ROS Interface: data receiver/transmitter.
- ROS Node: transfers data between ROS Interface and the server.
- Server: facilitates message passing between HoloLens 2 and ROS.
- Subscriber/Publisher: transfers data between the device interface and the server.
- Device Interface: I/O functions for the operator of the device.
- User Interface: sends data from the HoloLens 2 to Subscriber/Publisher

VR Tools is a component that is developed around the Wearable Gloves. The operator of the device receives and likewise shares data with the VR Environment using as a medium the Wearable Gloves themselves. The data is translated to haptic and force feedback for the operator and composite information for the VR Environment.

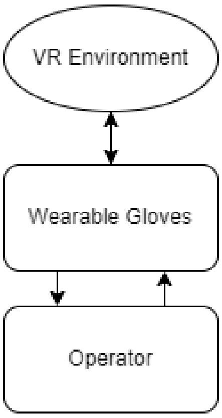


Figure 33. Virtual Reality tool logical view

It is composed of the following modules (Figure 33):

- VR Environment: simulates end-users' environments.
- Wearable Gloves: provide input and output between the operator and the VR Environment.
- Operator: Utilizes wearable gloves.

4.2 Process View

This section provides the general descriptions of the dynamic behaviours of the CoRoSect System components for data collection and process execution. These are intended as templates to later assist WP4 on defining in detail the components, interactions and data flows involved on each of the processes to be carried out to perform CoRoSect pilots and so the CoRoSect orchestration, as wel as WP9 on guiding the integration processes that supports the pilot's deployment and so the CoRoSect system's evaluation. In particular, here are presented the common CoRoSect data collection/data update process and a simple subsystems orchestration.

4.2.1 Data Gathering/Updating process

This process, represented in the Figure 34, illustrates the way that new data is reported from the Shop Floor Level and distributed among all CoRoSect subsystems. It is divided into three main steps: i) data production and formatting, ii) data uploading; and iii) data distribution, and, although here the data flow is shown from a Shop Floor device, the process would be similar to this if e.g. the source was another CoRoSect subsystem updating a Robot controller.

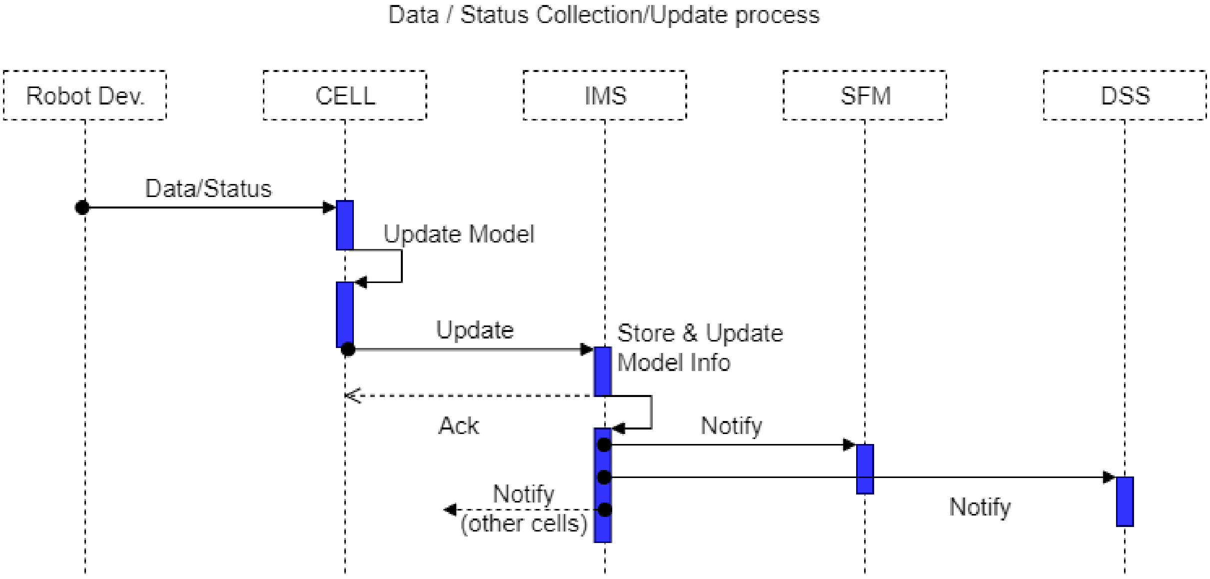


Figure 34. Data gathering/updating process view

Data production and formatting: the raw data is generated by the hardware device. This data may contain asynchronous sensor measurements, periodical status reports or a message to the MES raising an alarm or notifying a completed task. This is sent to its corresponding Cell Controlling system, where this data is collected, updated locally, and formatted according to the corresponding device information model. Once this is done, the Cell server will generate an update message with the updated model as its payload to the IMS.

Data uploading: the IMS receives the Update message and checks the payload. Internally updates the new context information from the identified device into its corresponding digital twin (the master copy of the device's information model) also updating the historical records of this device with the new data.

Data distribution: once the updated data has been processed and properly stored, the IMS automatically sends a notification message (Pub/Sub mechanism) to all subsystems registered to the updating device data source. These subsystems would include the SFM, the DSS and any other Cell/Component that may require to track the device. This way, CoRoSect System is aware of what happens at all levels in real-time.

4.2.2 Process Orchestration

This process (detailed in the Figure 35) presents a template for the common integrated process execution and orchestration of all the involved tasks and subtasks, and how the CoRoSect components interact together to keep the information updated and consistent. It is divided into several stages that represent the common flow of this orchestration from the simplest possible way, to make this easy to understand and to be expanded to address more complex orchestration processes. This will be managed within WP4 and tasks 4.2 and 4.3.

4.2.2.1 Process loading and set up

Conceiving a CoRoSect farm process as a set of tasks and subtasks to be executed and coordinated by one or several Shop Floor components (robots and humans working together), each of these processes will be mapped according to a defined process information model and stored into de IMS by a system operator (an entity that defines and details the specific process). This model will contain so a list of tasks, timings, involved assets and actions to be carried out for the proper execution of the represented process. Tasks would be also represented by models stored in the IMS that contain the commands and templates to be executed by corresponding assets.

When a process (named e.g. PA) is started (by an operator or automatically by the production system) through the SFM, it requests for PA process file to the IMS. SFM then loads this PA process into its Process Orchestration module and starts capturing required information to execute the assigned tasks: i) it requests first for the status of all assets involved in PA to check if all of them are ready and ii) request them all context information (temperatures, insects' numbers, robots' location, storage availability, etc.) needed to set up the first task. If all of this is correct, the first task may be ready to start.

4.2.2.2 Context evaluation

Before triggering this first task, the system first queries the DSS, in order to modify the orchestration process in case there are other conditions that may improve whole production process or any unusual condition or user intervention that recommend altering (or stopping) the process execution. For this action, the DSS queries the IMS for extra information that complements the one it already has, coming from the user interface, its knowledge database or previously provided by the SFM. With all this, its Decision Module will evaluate the situation and return a response to the SFM. In turn, the SFM processes this response and proceed according to it: rearranging the planned tasks, stopping the process or starting it as programmed.

4.2.2.3 Tasks' execution

Once everything has been validated and the required information is collected, the SFM initiates the PA process execution, starting with the first task (T1.PA). If needed (in case that this was not loaded during the first stage), the SFM requests IMS for the T1.PA description, which includes all the commands to be sent to all involved Shop Floor assets. Now, the following steps are repeated for all tasks within the PA process. For the sake of simplicity, these are listed subsequently, but their execution could also be done in parallel, depending on the way the different tasks and production processes are planned:

- Task execution request: the SFM reads the Information Model of the Cell System that will execute the T1.PA from the IMS and, through the IMS it sends the proper command, with required data, to its Cell Controller. This receives the command and internally connects with the corresponding device controller to check the status and get (if required) further information. Here, further information from nearby devices may be required. In this sense, it can request the IMS for this needed data. If everything is fine and ready, the device controller starts the proper physical required actions.
- Actions' executions: the execution of T1.PA may require running several related actions (or subtasks) at the Cell Controller level. In this sense, the Driver Controller and the Cell Controller will manage this set of actions, reporting to the IMS any change and/or update that happens within the device status, updating (at the IMS level) its information model. Any change here will be automatically reported to the SFM and the DSS.
- Task update: once all actions/subtasks have been completed at the Cell level, the Cell controller reports this to the IMS (T1.PA executed ok) and updates the related context information. The IMS automatically reports this to the SFM and the DSS.
- New Context evaluation: SFM gets ready to execute the next task in the list, but previously updates its context information by requesting the IMS and collects all required information for the next task. In turn, it also requests the DSS for any possible modification on the normal process.

4.2.2.4 Process' end

Previous steps are required till all tasks of the PA process have been completed. When the last task (TM.PA) has been properly executed, the corresponding cell controller updates the IMS which in turn notifies all the subsystems registered. With the collected information, SFM and DSS updates their own knowledge databases and reports the operator. PA has been properly executed and the system is ready to load and execute a new process.

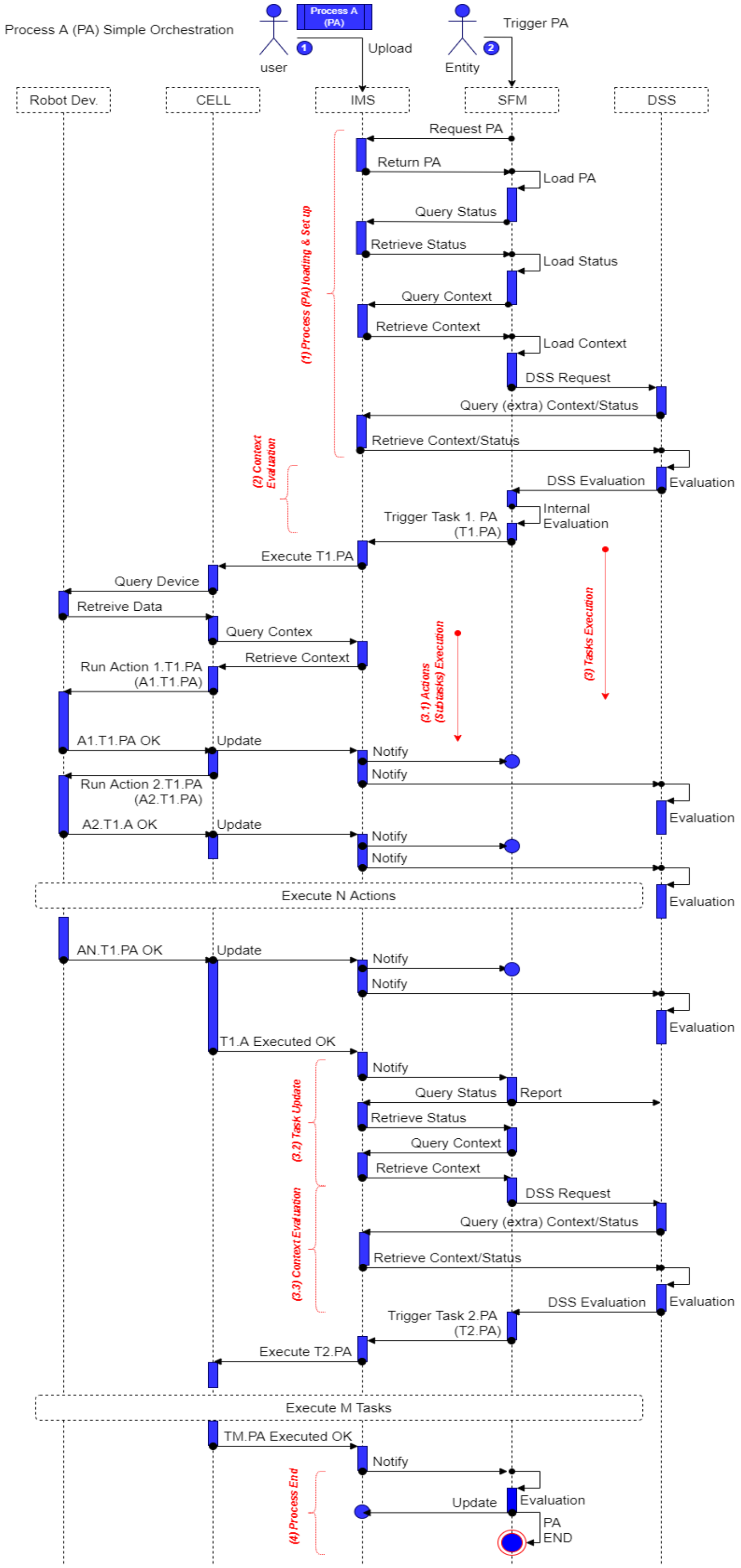


Figure 35. Process orchestration view

4.3 Development View

Combining the logical and process view of the proposed CoRoSect System Architecture, here is introduced a high-level development view of this architecture (Figure 36). The figure presents the structure of the main CoRoSect modules to be developed, including the interfaces and the main properties of each of them.

Starting from the Shop Floor level, each device (robotic system, AVG or I-Crate) is represented by a set of attributes, to map sensors measurements and status of the device and functionalities, that are translated into commands. These compose the Information Model that maps the device into an entity, with attributes and methods, building its Digital Twin (DT). This DT is exposed and managed by the system server and the MES through the IMS interfaces: Context; Time Series; and Publish/Subscribe (notifications). These three interfaces will drive main integration between MES components, Shop Floor and Human-Robot collaboration layers, as well as with the Enterprise Resource Planner (out of the project's scope). SFM and DSS, in turn, will connect with the User Interface to allow the operator to interact with the CoRoSect System.

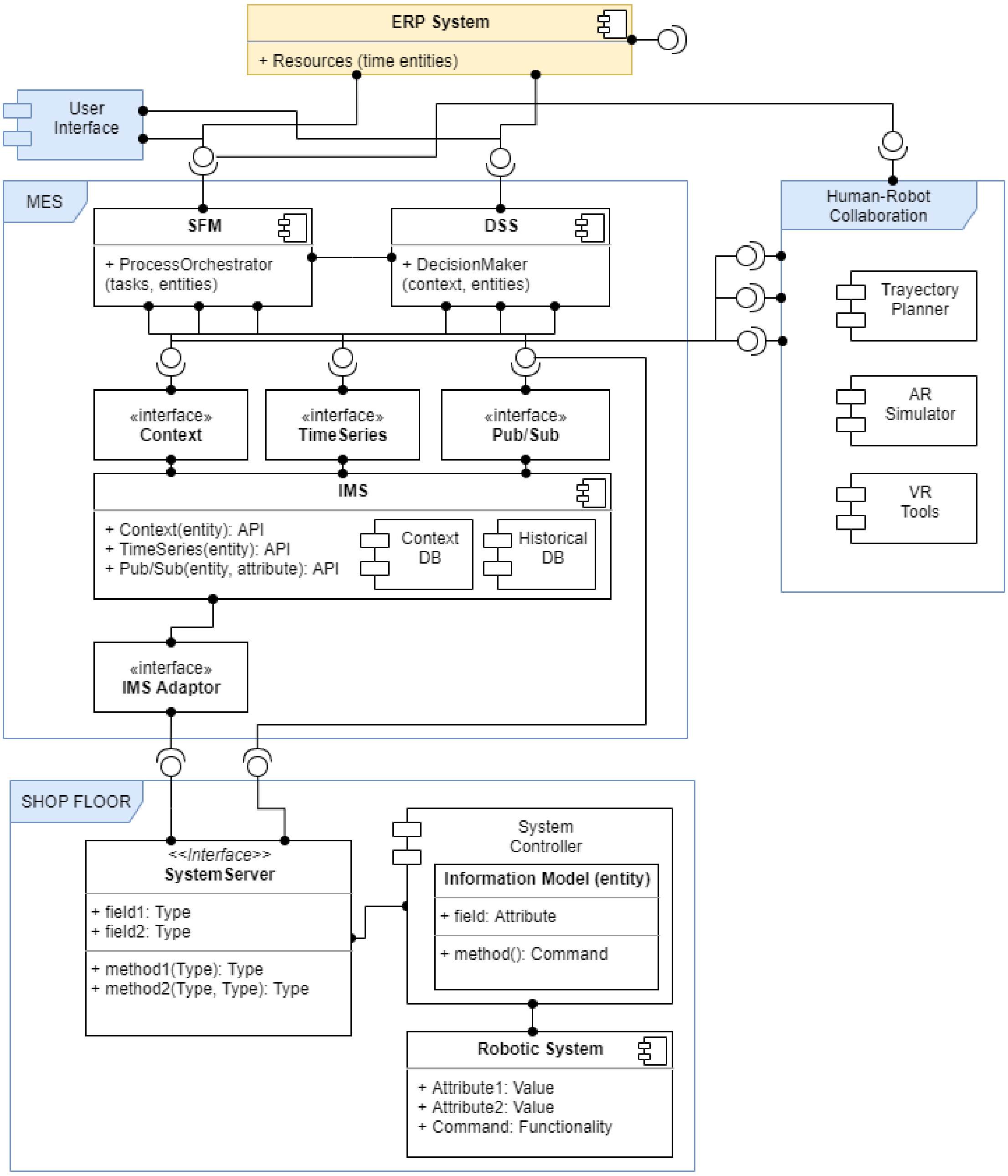


Figure 36. CoRoSect's development view

4.4 Building CoRoSect 4.0 Architecture Implementation

4.4.1 RAMI 4.0 compliant CoRoSect 4.0

As of today, the various components present in the IT and OT layer of the insect production system operate and integrate in the form of a CIM pyramid or Industry 3.0. This approach has notable weaknesses. These weaknesses run across functions, processes, components and data and processing communications. A brief overview of these weaknesses is given below:

a) Weakness in Functions: In the current insect production system, the functions are associated with the Hardware. This means that the technical implementations of MES or SCADA can only be available in the said software and on the same level as given in the ISA-95. This in terms means that the functionality is not portable and is rigid along the HW implemented.

b) Weakness in the Process: With the current insect production system, the process of producing insects lies with the associated HW present in the insect production system. Though it is well suited for the mass production of insects, it is not for mass customization. This would mean that the same hardware and software cannot be used for producing any other type of insect without requiring substantial changes [6].

c) Weakness in Components: The components present in the insect production system be it HW or SW are tied to their respective levels in the ISA-95. They do not offer any modularity and plug and play features. Effectively, this results in a constant divide between the IT and OT components. Furthermore, because of this, the addition, or deletion of components from the system results in breaking changes [6].

d) Weakness in Communication: As the components are often tied to their respective level, they can only communicate with their adjacent levels of the ISA-95. This potentially means that the AGV present in the Level 2 of the ISA-95 will not be able to directly communicate with the functions of the MES present in Level 4. Moreover, each component (HW/SW) is implemented using a set of proprietary interfaces. The management of these many different interfaces is not only cumbersome but also enhances the chances of readaptation of the system for the component.

e) Weakness in Data Processing: The data models and structure used by each of the components present in the production system is not standardized and does not cover all the phases of the lifecycle of the component. With this, the information islands are created, and the sole source of information is missing.

All these weaknesses combined result in loss of features like Flexibility in operating processes, Real-time data capture of process and component, Dynamic planning and optimization, Continuous track and trace and transparency of manufactured products (in this case insects) [19] [20] [21].

Worthy to note that these set of features are required to improve the efficiency (ensuring the constant quality of insects produced) and to improve productivity (ensuring constant quantity of insects [22]).

Thus, to have the possibility to add those missing features and overcome the weakness listed above, it is required that the Digitalization & Networking of the various components present in the IT & OT levels of ISA-95 is performed.

The Digitalization will be responsible for bridging the gap between the IT & the OT creating an information exchange flow. This would lead to better connectivity and a tighter integration between the various components present [23].

There are numerous architectures like the IIRA, IoT etc. given above. All these architectures help in performing digitalization. But it is the RAMI4.0 that is most near to the manufacturing domain and captures the aspects of I4.0 as a whole. Its hierarchical axis helps in identifying the assets or group of assets based on IEC 62264 and the vertical axis helps in giving out methods to perform the digitalization.

For the digitalization to happen based on RAMI 4.0, the vertical axis of the RAMI 4.0 needs to be traversed. The vertical axis can be traversed bottom-up or top-down. The “product” word in the Figure 37 refers to the physical asset be it AGV or the group of assets i.e., the D-Cell.

Figure 37. Vertical Axis of RAMI 4.0 and its descriptions [24]

The Top-Down approach is described under:

a) Business: The is the layer where the answer to the question 'What business would I like to perform with the services offered by the asset?' will be mapped.
b) Functions: The layer is concerned with answers as to what functions are required to perform that business.
c) Information: What are the digitalized & standardized information and data models that will be required for exposing those functions?
d) Communication: What is the technology or protocol used to access the digitalized data?
e) Integration: How will the information transition from the physical world to the real world?
f) Asset: What will be the asset? Will it be a single or a group of assets?

For the bottom up, the approach starts with the first layer:

a) Asset: This is the physical thing that will be digitalized.
b) Integration: This is the layer where the transition of physical data to digital data takes place.
c) Communication: This layer provides access to the data and information coming from the integration layer. This is where the internet-based communication lies (IIoT and IoT)
d) Information: The information layer is involved with representing the data that is coming from the integration layer in a correct and standardised way. This is where the information is modelled using a suitable representation.
e) Function: The functions layer maps which functions can be conducted with this digitalized data that is exposed as services in IoS.
f) Business: This is where what business and application can be performed with these set of functions that are made available from the digitalized asset in the form of services.

To realize the concept of digitalization and turning assets or components into I4.0 components, DIN SPEC 91345 RAMI 4.0 recommends creating an Asset Administration Shell (AAS) for each component.

The AAS covers Layer 3 and Layer 4 of the RAMI 4.0 to perform functions (Layer 5), associated with the business to be performed in Layer 6.

The AAS helps in implementing the principles for Industry 4.0 for each asset. It helps in turning assets into modules that use services to exchange data and information with other assets that have AAS.

It helps in the decentralization of the production process and decouples process execution from the assets. By standardizing the way, it models the data and communicates it to other modules, favours interoperability (one of the main precursors for Industry 4.0 [5]).

The next section will now identify which are the assets from the CoRoSect that needs to be digitalized and map them to the hierarchical axis of the RAMI4.0. These are the first steps in making CoRoSect 4.0

4.4.2 Identification of Assets

Industry 4.0 intends to represent assets as an individual in the information sphere to enable meaningful interaction with IT systems instead of interpreting the state of a production system by reading individual sensors and actuators. An "asset" can be a physical or logical object owned by or under custodial duties of an organization, having either a perceived or actual value to the organization.

Industry 4.0 contemplates assets are more than just machines, production modules, or systems; but also, individual products, software installations, intellectual property, or human resources.

Assets may be material or immaterial and of various natures:

- Physical objects, for example, equipment (Robot, AGV, Sensor, Actuator, Connectors, Camera…), part components or products (final or intermediate)
- Software (firmware, applications, engineering tools…)
- Documents (data media, life cycle documentation…)
- Immaterial (licence, plan, process definition, standards, recipe, general procedure…)
- Information
- Human (service technician, programmer, operator…)
- Service

Depending on the distinct natures, the traits of assets can differ (for a piece of equipment, the serial number is relevant; for a person, the gender may be relevant). In other words, a production asset can be a plant, a machine, a station, a motor etc.

In CoRoSect 4.0 seven assets are identified in the production facility which is mentioned below:

- D Cell
- M Cell
- IC (Intelligent Crate)
- AGV (Automated Guided Vehicle)
- IMS (Information Management System)
- SFM (Shop Floor Manager)
- DSS (Decision Support System)

To perform the digitalization of the assets noted above in the context of Industry 4.0, the assets can be positioned/mapped to the Hierarchical Levels axis of DIN SPEC 91345 RAMI4.0. A description of the Hierarchical levels is given below:

- Connected World – It describes the relationship between an asset or combination of assets (such as an installation or company) and another asset or combination of assets (another installation or company), in other words, for example, a network of factories.
- Enterprise – It is a system which is responsible for managing resources for the entire enterprise.
- Work Centers – It is a system responsible for manufacturing operations management.
- Station – An individual machine or cell of assembly equipment responsible for a process.
- Control Device – It is defined as the brain of industrial control systems and is usually represented by programmable devices.
- Field Device – Represents the functional level of an intelligent device for example a smart sensor or actuator. They are electronic devices used to detect and identify sensor components and technologies.
- Product - Denotes the cooperating or collaborating product to be manufactured as an integral part of an Industry 4.0 value-added process.

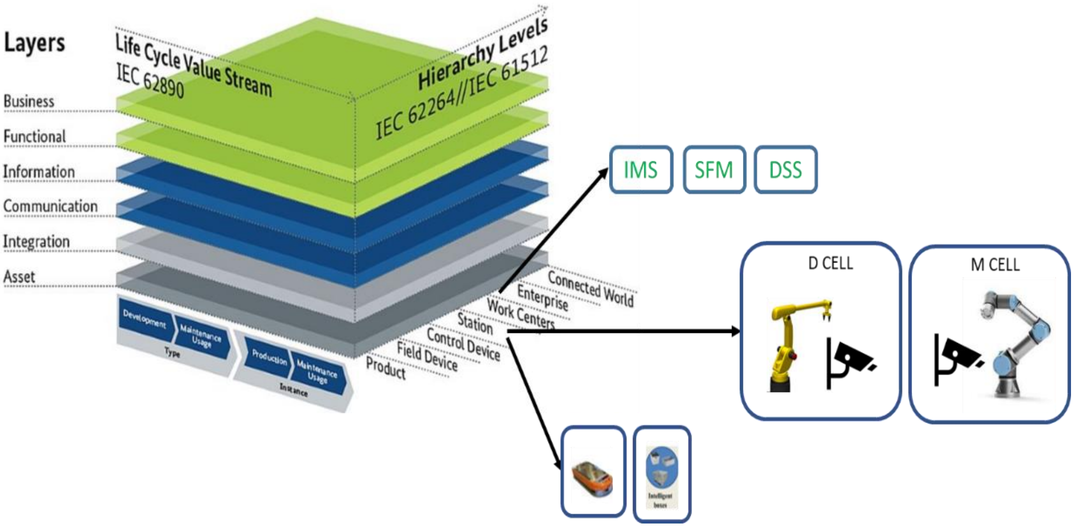


Figure 38. Position of assets in RAMI4.0 Architecture

The assets of CoRoSect 4.0 when functionally seen can be mapped (Figure 38) to the Hierarchical axis RAMI4.0. The IC, AGV, D Cell which consists of destacking robot and camera/scanner and M Cell which contains the manipulating robot, and a camera/scanner are positioned at the Station level because these assets are either machines or cells that are responsible for conducting the process. The MES modules Information Management System (IMS), Shop Floor Manager (SFM) and Decision Support System (DSS) are positioned on the Work Center level of RAMI 4.0 Hierarchy levels.

The digitalized form of each asset can be mapped with the vertical layer of RAMI 4.0 and the digitalized form of the asset is referenced in the layers above the Integration layer. The digitalization of each asset will be achieved through implementing Asset Administration Shell (AAS) for each asset which will be covered in the upcoming sections.

4.4.3 RAMI 4.0 compliant Digitalization Architecture.

Based on RAMI 4.0, for the assets that are identified in section 4.6.2, the assets are transformed into modules. This modularization or in other words creation of the Digital Twin of the asset is based on the application of Asset Administration Shell (AAS) Methodology. The AAS covers Layer 3 and Layer 4 of the RAMI 4.0. Layer 1 corresponds to the asset that is identified and mapped to the hierarchical axis of the RAMI 4.0. Layer 2 is the integration layer where the corresponding technology implemented by the asset provider to convert the physical values to the digital values is present. The Blue Box from the Figure 39 corresponds to the RAMI Layer 4.0 where the standardized data models for monitoring and executing the functions shall be specified and implemented. The Arrow above, corresponds to the Industry 4.0 compliant communication protocols like the OPC-UA, MQTT or the Webservices. This would help in communicating the data and information from the asset to the I4.0 network. Through these communication capabilities, other modules in the I4.0 network can access the services provided by the corresponding module. These services would then be used to perform functions (Layer 5 of RAMI 4.0) and associated business (Layer 6 of RAMI4.0).

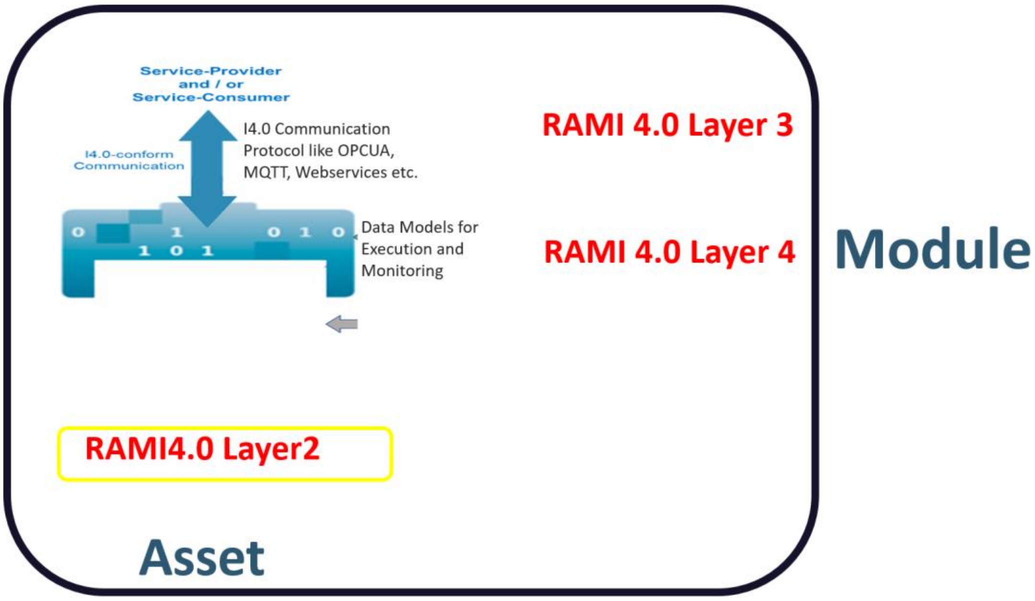


Figure 39. Migrating Assets to Modules

4.4.3.1 Logical View of Digitalization Architecture

As described above, all the assets that have been identified as part of CoRoSect would be transformed into modules. These digitalized modules will communicate their services and necessary data in the I4.0 network. Logically considering, this means that the division between the IT & OT will be diminished, and all the assets be it in IT or the OT become part of a single coherent I4.0 network.

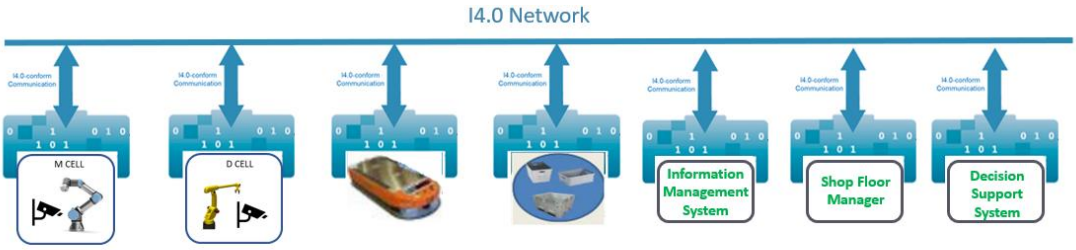


Figure 40. Logical View of the RAMI 4.0 compliant CoRoSect 4.0

As seen in the Figure 40, the digitalized modules are denoted by a blue box. The blue boxes help the assets turn into modules by digitalizing the necessary data and function in a standardized format. The arrow over the box is exposing their necessary data and functions in the form of services using the I4.0

complaint communication technology. Moreover, this approach would help in achieving the plug in and plug out of the necessary modules without breaking changes to the system or the process itself.

4.4.3.2 Functional View of Digitalization Architecture

For the CoRoSect 4.0 to conduct the process of insect rearing with the use of digitalized modules, it is important to functionally view the different modules (Figure 41) of the CoRoSect 4.0 and what will be their respective functions.

The module AGV will be responsible for the transportation of crates from one cell to another. The AAS of AGV will be responsible for standardizing the necessary data and functions associated with the AGV. The Blue arrow will have the function to expose the standardized data from the asset in the I4.0 network.

The module Intelligent Crate will be responsible for housing the product i.e., the insects and measuring the environmental conditions that are present in the crate for the insects to breed. When the IC will be digitalized and AAS will be responsible for standardizing the necessary data and functions associated with the IC. The Blue arrow will have the function to expose the standardized data & functions in the form of services from the IC in the I4.0 network.

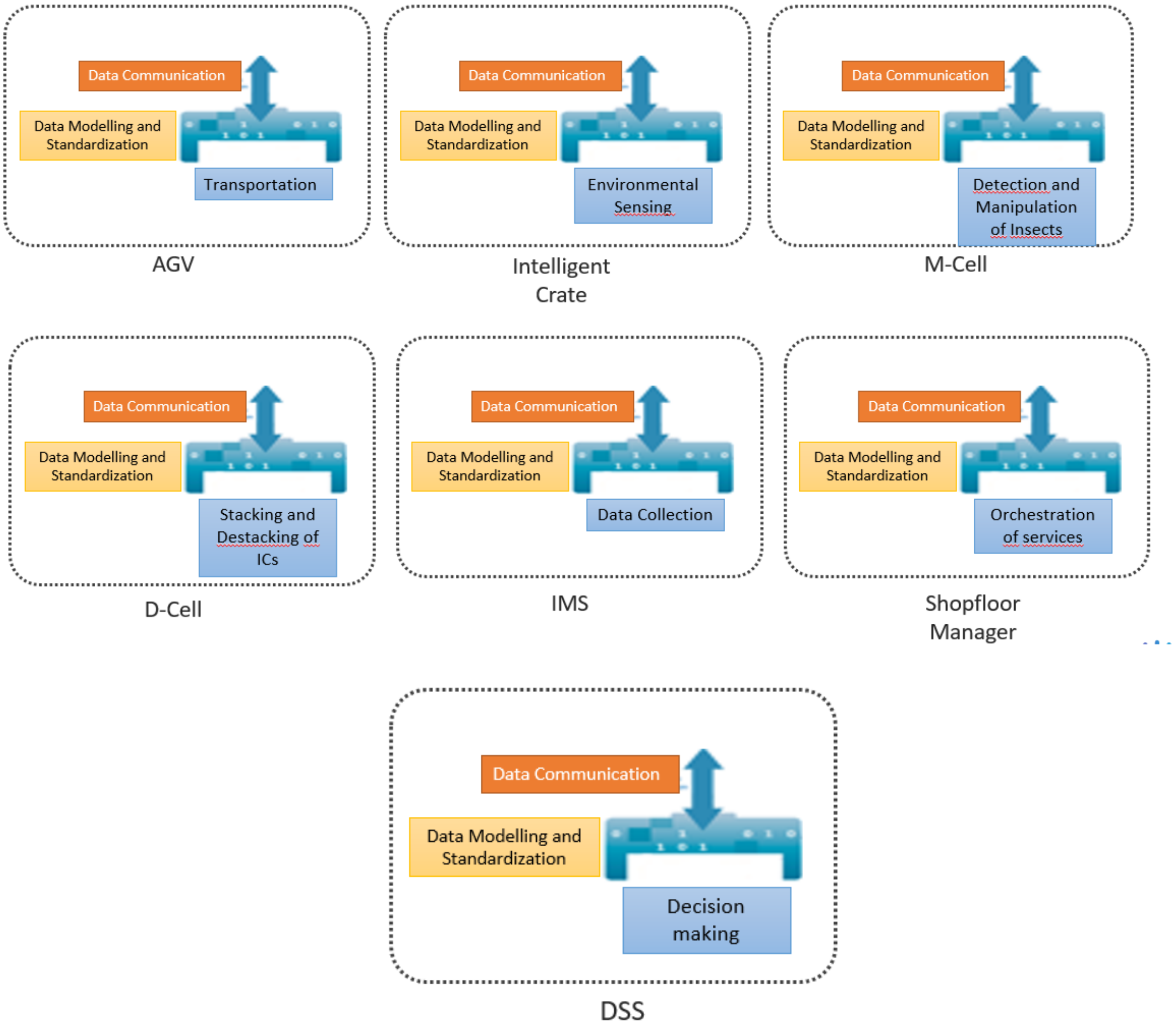


Figure 41. Functional View of the RAMI 4.0 compliant CoRoSect 4.0

The module M-Cell will be responsible for detection and fine manipulation of insects. When the M-Cell will be digitalized the AAS will be responsible for standardizing the necessary data and functions associated with the M-Cell, which will enable the whole M-Cell as such to receive commands from the Shopfloor Manager. The Blue arrow will have the function to expose the standardized data & functions in the form of services from the M-Cell in the I4.0 network.

The module D-Cell will be responsible for stacking and destacking the crates of insects. When the D-Cell will be digitalized the AAS will be responsible for standardizing the necessary data and functions associated with the D-Cell. This will enable the whole D-Cell as such to receive commands from the Shopfloor Manager. The Blue arrow will have the function to expose the standardized data & functions in the form of services from the D-Cell in the I4.0 network.

The module of IMS will have two functionalities. First, it will be responsible for collecting and storing the data from all the modules listed above and present in the I4.0 network. Secondly, a context broker inside the IMS will allow to interconnect all the modules listed above and manage the entire lifecycle of context information including updates, queries, registrations, and subscriptions. When the IMS will be digitalized, the AAS of the IMS will be responsible for standardizing the necessary data and functions associated with the IMS. This will enable the whole IMS as such to receive commands and send data to other modules. The Blue arrow will have the function to expose the standardized data & functions in the form of services from the IMS in the I4.0 network.

The module of Shopfloor Manager (SFM) has the functionality to orchestrate the numerous services provided by the different modules and conduct the insect rearing process. The functionality of this module corresponds to the Execution Management function of the MES as described in the ISA-95 [1]. When the SFM will be digitalized, the AAS of the SFM will be responsible for standardizing the necessary data and functions associated with the SFM. This will enable the whole SFM as such to receive commands and send data to other modules. The Blue arrow will have the function to expose the standardized data & functions in the form of services from the SFM in the I4.0 network.

The module Decision Support System (DSS) will be responsible for doing Performance Analysis, Tracking, Scheduling and Dispatching of Job Orders for the Shopfloor Manager. This implies that the single module of DSS will be responsible for covering 4 functions of MES as stated in the ISA-95 [1]. When the DSS will be digitalized, the AAS of the DSS will be responsible for standardizing the necessary data and functions associated with the DSS. This will enable the whole DSS as such to receive commands and send data to other modules. The Blue arrow will have the function to expose the standardized data & functions in the form of services from the DSS in the I4.0 network

4.4.3.3 Technical View of Digitalization Architecture

The technical view of the CoRoSect 4.0 expands the functional architecture in a more technical oriented manner. This is depicted in Figure 42, the RAMI 4.0 Layer 1 is now expanded and with their respective controllers. Like in the case of Intelligent Crate, each intelligent crate will send the data to the IC Controller, hence for digitalization the asset would include the crate and it respective gateway. This controller will then use any proprietary communication/ digital technology that is used till now. This corresponds to the RAMI4.0 Layer 2 (Integration).

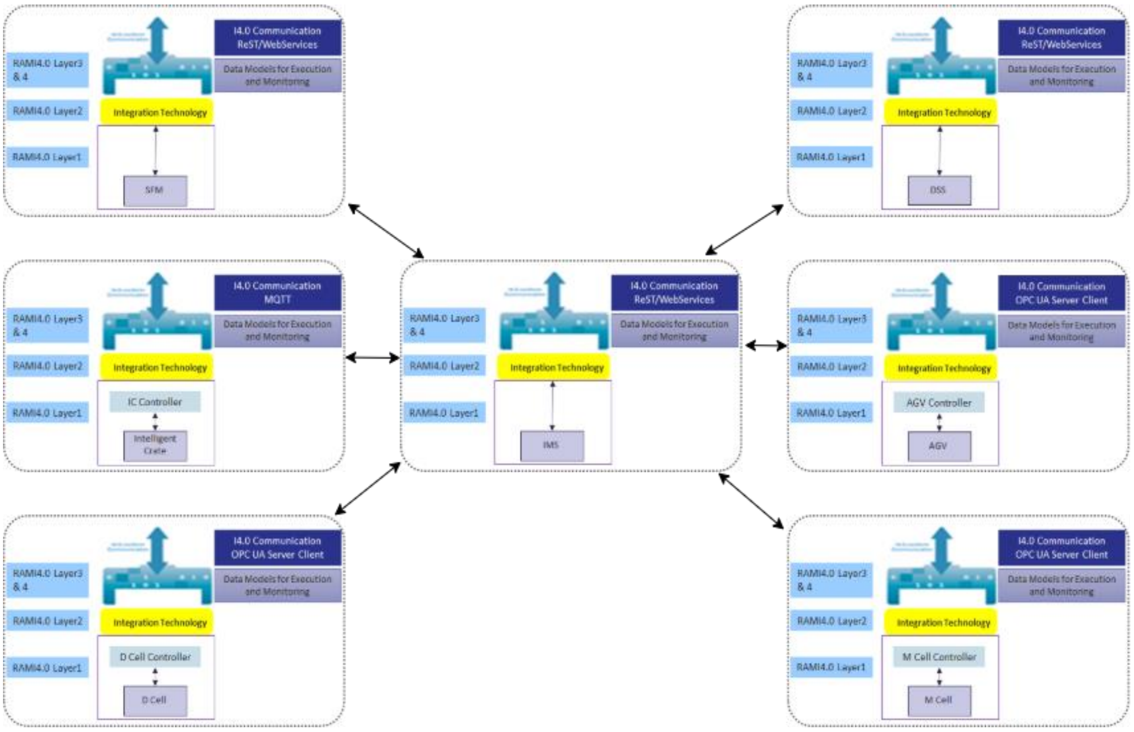


Figure 42. Technical View of the RAMI 4.0 compliant CoRoSect 4.0

For Layer 4, the Submodels of the AAS where the data and functions will be specified pertaining to functionalities of execution and monitoring. These Submodels will be converted into a suitable exchange format. This format will depend on the communication protocol chosen for a particular module. But the choice of communication protocol is restricted to either MQTT, OPC-UA or Webservices to be compliant with I4.0. If MQTT/Webservices is chosen, the AAS is converted into JSON format. For the OPC-UA, the data is serialized into OPC-UA Information Model which is based on the XML format.

As from the data gathered till now, the IT components i.e., IMS, SFM and DSS would use webservices as the communication technology. The intelligent crate would use MQTT as communication technology. The AGV module, D-Cell Module, M-Cell Module proposed to use OPC-UA communication technology. It should be noted clearly that the use of proposed communication protocols is subject to testing and validation and can be changed in the further course of the project. The details of the same shall be updated in Task 9.1 and 9.2.

It is important to note that in this case, the modules would communicate to each other through the context manager provided in the IMS.

5 Candidate implementation technologies and tools

This section proposes a set of suitable technologies to develop and integrate the different components of the CoRoSect System architecture, defined here on its first version and in a technology-agnostic way. These technologies apply to different levels of the architecture and are a) firstly oriented to support components' functionalities in terms of communications, integration and data management; and b) secondly to create the common environment to code, deploy and test the prototypes of these components. Considering that the proper development of CoRoSect's assets will be done in their corresponding work packages, the text here just provides implementation insights for CoRoSect's developers aligned with the CoRoSect RA and the Industry 4.0 standards, with no further restrictions to other RAMI4.0 compliant approach.

5.1 Implementation Technologies for Data Management

From a data management overall perspective, we consider here, and so propose potential technologies and standards, the functionalities that deals with a) integration, in terms of data modelling, data sharing and routing; b) data storage and persistence layers; and c) data access with two different approaches: the visualization tools that assist on the system management and the security mechanism to control those accesses.

5.1.1 Data Routing and Data Modelling

Datasets within CoRoSect will be managed in terms of information models or data models, as documents that organise and represent in a standard way all the parameters (attributes) and functionalities (commands) linked to a given asset (entity) of the system. These models digitalise all the elements involved in the production chain (as Digital Twins) and are used as payloads to query/retrieve information or as references to access a device and trigger actions by all the proposed interfaces (OPC, MQTT or NGSI) that route the information. For these purposes, CoRoSect adheres to RAMI4.0 and relies on the Asset Administration Shell to implement the Digital Twin concept that covers the AAS Information Model and AAS Communication system.

5.1.1.1 AAS Information Model

The AAS information model is the part where the general structure and how to model information is stated. (Figure 43)

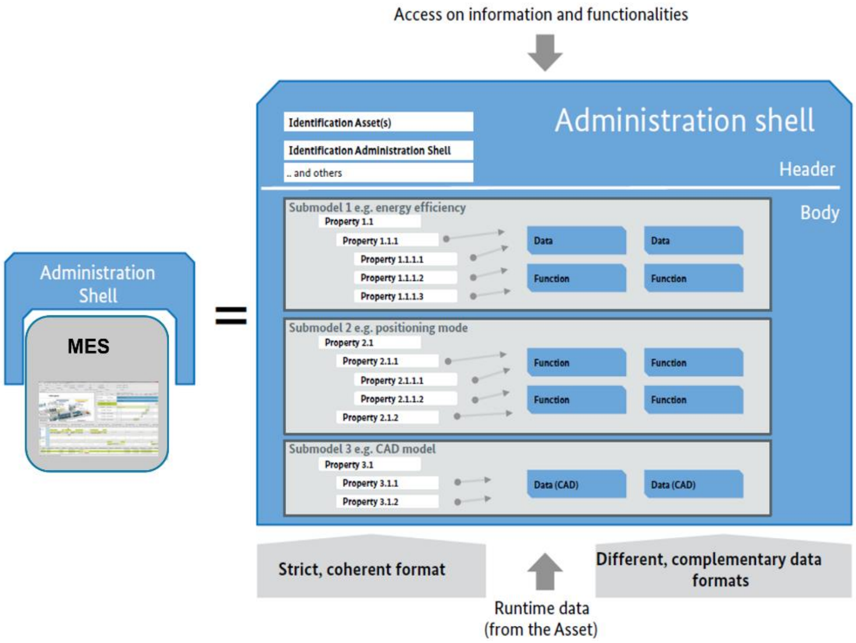


Figure 43. General Structure of the AAS [25]

An AAS Information Model has two parts, the header, and the body. The header contains the information for the identification of the asset and its AAS. This helps in uniquely identifying the AAS in the I4.0 network. The AAS can be accessed using this address.

The AAS Information Model is a technology-neutral information model. It does not specify what information to model, but it specifies in which way to model. The AAS Body is in this regard.

In the AAS, there is a concept of submodels. These submodels are the actual places where data and information will be stored with the properties. These submodels will contain submodel elements that will be of type property, operation, files etc.

The specifications on how to create AAS and what contents it needs to have can be found in the document [26]. The Platform Industrie4.0 is continuously doing partnerships with asset providers both hardware and software to standardize the contents of these submodels.

Apart from the submodel, there is a concept of dictionaries. It is called a concept dictionary in the context of AAS. These dictionaries contain the reference to the definition of the properties defined in the submodels. This ensures that a property like 'MaxRotation' will have an associated meaning to it. This helps in bringing interoperability among the several modules that will be communicating to each other using their AAS.

5.1.1.2 AAS Communication

The AAS must communicate themselves in this novel I4.0 Network. The I4.0 is composed of various AA Shells that are communicating with each other in a Service-Oriented Manner.

The AAS can be classified into three types in terms of communication, a passive AAS, a reactive AAS or a proactive AAS. The difference between these three is that the passive AAS in the form of QR codes have no power or deployment of its own but has valuable information stored. This method is used in an exchange of information utilizing AAS. It is described in labelled part 1 of Figure 44.

The other type of AAS communication is of nature "Reactive" where the data from the AAS is requested on-demand using the set of APIs. The external partner need not have its AAS but has access to the API, which exposes the AAS of that asset.

The third form of AAS communication is of the type "Proactive". This means that the AAS can itself send some information to the other AAS using the I4.0 network and language. Where events are concerned, the AAS of one asset can refer to the AAS of the other asset and pass on the data, whenever required.

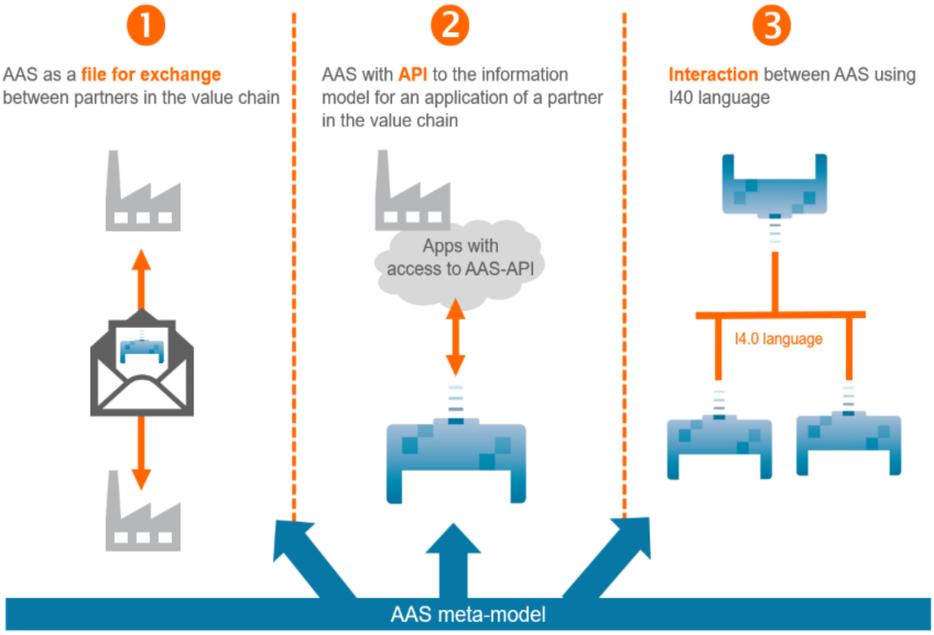


Figure 44. Various Types of AAS Communication [27]

For the Type 2 and Type 3 AAS communication, the Industry 4.0 compliant communication protocol is to be used. This would be Layer 3 of the DIN SPEC 91345 RAMI 4.0.

As per DIN SPEC 91345 RAMI 4.0, the various Layer 3 communication protocols are OPC-UA, MQTT and Webservices (Figure 45). All these protocols support service-oriented architecture as they are built considering this architecture.

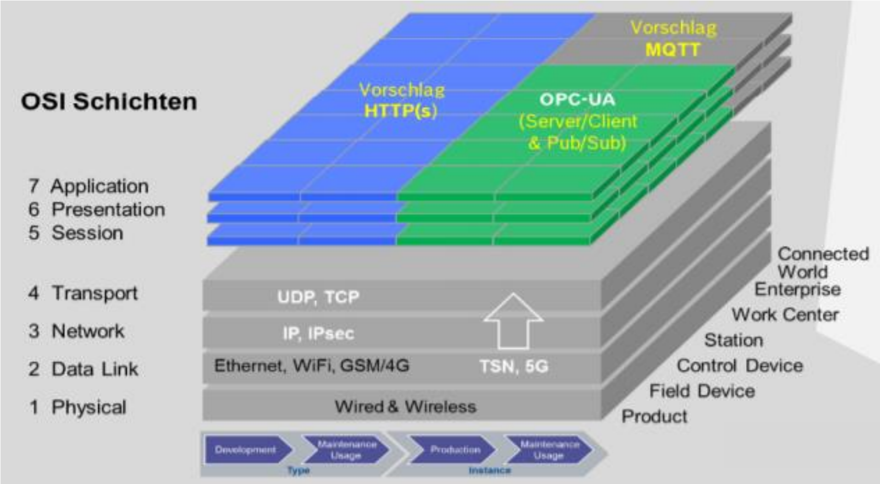


Figure 45. Communication Protocols in the RAMI 4.0 [3]

OPC-UA is mostly used in the production instance. It is because OPC-UA supports information modelling. OPC-UA can support data communication over web-service or binary. It is one single protocol that can be used to connect any asset to any system in IT.

One another communication protocol is MQTT. MQTT stands for Message Queuing Telemetry and Transport. It works in Pub/ Sub fashion and is mostly used to add constraint devices like sensors providing data like Humidity, Temperature etc to the system. Hence for adding sensors like in the case of Intelligent Crate to the system and providing a digitalized intelligent crate, the MQTT can be used. For MQTT to relay the data stored in the AAS, the AAS is converted into the format of JSON and then sent as payload using MQTT.

Hence the application of the AAS is the requirement for doing digitalization following the Layers of the RAMI4.0. It is reinforced that each AAS will always have two parts, information modelling and communication and both are necessary for complete digitalization of the asset.

5.1.2 Data Persistence

Within a hypothetic production deployment, and likewise in CoRoSect's pilots, on a lesser scale, the system shall be able to manage large volumes of data to control and monitor the production processes and then, feed the factory ERP system and support AI tools that enhance the efficiency. Within the proposed architecture, the CoRoSect persistence layer relies mostly on the IMS, as part of the complete MES. In this sense, the IMS will work with two main datasets to be stored:

- **Context information**, which will cover the models and current status of each of the Shop Floor components (robots, crates, AGVs and any other involved device) with last reported values, plus the processes and tasks that will define the production chain. All this information will be stored as documents that represent the different models defined for each of these entities. For this purpose, a **document-oriented database** that supports semi-structured data is recommended. MongoDB is the proposed open component, also utilised by the F4I architecture implementation.
- **Historical information**, that stores all the records from the system components containing attributes' values with time and location reference. Due to the nature and structure of these information set which becomes bigger as time goes by, a **scalable SQL Data Management System** (DBMS) is recommended.

The final implementation of the Data Persistence layer will be developed in Task 4.3 and presented in D4.2 and D4.3.

5.1.3 Management, Configuration and Visualization Tools

In the context of the MES, a complete management system is essential; a system that supports capabilities of configuring, monitoring and visualizing flows of data streamed by submodules or internally between the different levels of the MES. This system aims to provide operational effectiveness and efficiency, and it can be conceived as a higher-level for enhanced observability (Figure 46) and interpretation of semantic information introduced from logging, metrics and tracing of the CoRoSect. For satisfying such a need, a software tool that offers a complete featuring of all the former is the Grafana platform [28]. Grafana is an open-source solution for running data analytics, pulling up metrics that make sense of a massive amount of data, while also monitoring applications with the assistance of customizable dashboards (see Figure 8). This makes Grafana highly tailorable since it enables users to create a self-driven (self-sufficient) observation strategy, while also preserving user-friendliness. Among many, some of the most significant features rely on the fact that compatibility with many standard database frameworks are supported (i.e Graphite, Prometheus, Influx DB, ElasticSearch, MySQL, PostgreSQL, etc). Moreover, users are enabled to extract, transform, correlate, query, analyse, visualize and monitor data while also featuring transparency across team members. Integration with the software is simple, and thus suitable for rapid familiarization, and it is briefly described as follows. Having an activated account and the localhost ports configured for the application to run, by writing database-compatible queries, users are enabled to import data and make use of any of the offering features, flexibly and collaboratively. As a result, all these key features enable adaptation and adjustability for building a comprehensive layer, aiming to satisfy supplementing necessities of the MES.

time	flow1	flow2	flow3
Jun 7th, 2021 14:46:16	50%	50%	50%
Jun 7th, 2021 15:01:16	49.4%	44.9%	49.2%
Jun 7th, 2021 15:16:16	45.3%	47.5%	53.9%
Jun 7th, 2021 15:31:16	47.9%	52.9%	59.3%
Jun 7th, 2021 15:46:16	51.4%	58.5%	56.0%
Jun 7th, 2021 16:01:16	54.3%	61.9%	58.5%
Jun 7th, 2021 16:16:16	56.0%	63.3%	65.3%
Jun 7th, 2021 16:31:16	56.7%	66.9%	59.6%
Jun 7th, 2021 16:46:16	56.1%	59.5%	59.1%
Jun 7th, 2021 17:01:16	58.3%	57.0%	65.7%

Figure 46. Grafana General Panel

The applicability of Grafana can be described using the Quality Management use case depicting the steps in which adaptation within the MES system can be achieved. At first, by collecting values from sensors, and having them stored within IMS in a predefined form, the capability of creating a Grafana data source according to the type of the values stored, is offered. Proceeding on such action, saving, testing, and initialization of an empty dashboard is been made automatically. Successful definition of the data sources of interest requires configuration on querying tables, columns, values of interest, filters and grouping methods. Therefore, such queries can be used to request selected fields and values, while filtering using thresholds stored by the DSS is also feasible; and thus, establishing self-driven control operations for validating that the obtained values are conforming to the logic defined by the DSS. Furthermore, Grafana features multiple queries, hence enabling comparisons between newly recorded and pre-stored parameters pumped from the database stored in IMS. Visualization of data is automatically realized on the application's panel and it is highly customizable. Its usability could be exploited on various steps of the data flow. For instance, ranging values, occupation states of the shop-floor, running, resting, discontinued or pending processes could reasonably consist of individual panels for boosting readability and interpretation. Consequently, the benefits obtained from such analysis could enhance both developers during implementation, debugging and optimization, and also end-users to assess services or even to define innovative benchmarks based on statistical measurements; since currently accessible.

5.1.4 Implementation Technologies for the Security Layer

The security layer considered for this first version focuses on the MES level and the integrity of the data management process within a Service Oriented infrastructure. In this sense the security framework to be implemented must guarantee two main requirements:

a) Only authorised entities (either information systems, operators or end users) should be able to access (request) data from the MES. This also means that different entities may have different granted accesses to different sets of information, what requires a role-based access control service.
b) Only identified and authorised entities (controllers, systems components, or end-users) should be able to write, update, modify or delete information to or from the MES. This guarantees that all data in the MES comes from authenticated and authorised sources, which protects the system against potential attacks that injects corrupted datasets and allows the temporal banning of damaged assets.

These requirements need a security framework able to manage different identities and define specific roles that provide configurable identification and authentication access controls, and, in turn, easy to be integrated within a SOA based infrastructure. An Identity Manager supporting OAuth 2.0 [29], the industry-standard protocol for authentication and authorization will be the right choice.

5.2 Technologies for Continuous Integration and Configuration of Components

Deploying, sharing and merging of continuously changing working copies to a shared codebase is a key practice among multiple programming entities for developing integrated content. Therefore, the collaborative implementation of various types of functions made by individual technical partners, as defined in CoRoSect, requires the validation of building and subsequent testing of every code change made to a working package. While avoiding integration challenges, Continuous Integration (CI) puts a great emphasis on this, by providing features for automated building, testing and deployment. This serves the process of verifying that a new commit pushed to a shared hosting server is not broken, in this way alleviating the risk of introducing issues in the build lifecycle, and also ensuring that the code changes and builds were never done in isolation. However, the implementation framework of CoRoSect does not exclusively implicates code-based software but also implementations based on API definitions of hardware-centric firmware (i.e. device configuration or data acquisition based on specific classes of software that provide low-level control of robots, sensors, body-mounted devices such as gloves and glasses, etc.). In those cases, for serving the needs of online testing, the adoption of CI tools further demand complex integration for wrapping or emulating hardware functionalities, consequently introducing impracticality. Thence, compiling and deploying stages as parts of the CI workflow could be validated using CI tools, whereas mutual and consistent testing of code structures among contributors, is considered to be achieved on a local level. Consequentially, two complementary solutions are proposed for the cooperative development of both implementation types of software (code-based and firmware-based) respectively.

For cooperatively developing code-based software modules, a hosting server capable of providing compilation, deployment and testing services among technical partners, is required. For serving this aim, GitLab server is thus recommended. GitLab is an open-source software, web-based, adequately documented, which enables collaborative plan, build, secure, and deploy software while further introducing transparency, consistency and traceability. Controlling can be achieved using both a graphical and a terminal-based control system, hence making it flexible for technical partners of CoRoSect to become acquainted with. Along with CI services, a plethora of additional features related to core software development stages are further supported. The integration process is briefly summarised as follows. Code updates are saved and uploaded on the shared codebase upon a commit and push. The inclusion of a YAML file within the code changes committed, generates a pipeline of sequential instructions called stages. Each stage is exclusively related to a certain production stage and is consisted of a collection of jobs that a runner executes in parallel to automatically build, test or deploy the current application, ensuring the validity of the code submission. A single job failing, results in the entire pipeline to fail, indicating an invalid code submission. On the other hand, successful submissions enable developers to open a merge request, so as to incite(/encourage) coworkers to comment on the implementation and suggest improvements while additionally creating artifacts for each job that can be downloaded and tested locally; thence reinforcing Continuous Integration.

On the contrary, synergistic elaboration among individual developing carriers on shared code involving the usage of hardware-centric firmware, although conventionally necessitates native testing using compatible-if not identical- connected hardware on both sides (/clients), CI tools are enabled exclusively for compilation purposes. Therefore, code changes could be committed on a shared codebase passing through CI's building stages, whereas testing using the actual hardware could be accomplished in a local environment. Among conventional solutions, on occasions where the is a lacking of compatible hardware setups on both contributing sides, if feasible, simulated testing by implementing dump data generators exclusively for testing purposes by the committers could be ventured(/attempted). As for that, for exceeding compatibility issues related to the platform of execution and ensuring compliance during testing, the Docker application is proposed. Docker is an open-source project for automating the deployment of applications as portable, by packaging applications and their dependencies in a virtual container that can run on the cloud or on-premises. Using Docker, technical partners of CoRoSect will be able to enhance incorporation by creating Docker images that contain executable application source code as well as all the tools, libraries, and dependencies that the application code needs to run as a container. Docker containers are the live, running instances of Docker images. While Docker images are read-only files, containers are live, ephemeral, executable content. Users can interact with them, and administrators can adjust their settings and conditions using docker commands. The creation of an image can be achieved by using already existing base images obtained from the Dockerhub repository, or by committing changes applied on containers that run older versions of the same image. Sharing of images among cooperators can be done by pushing them on the Docker hub server. A significant benefit that ought to be mentioned relies on the fact that having configured Gitlab Runner to use a Docker executor, Docker images can be built online on for serving the CI requirements, while also preserving a unified framework (GitLab) for sharing implementation resources between technical partners. Figure 47 represents the schema of the proposed collaborative CI/CD environment.

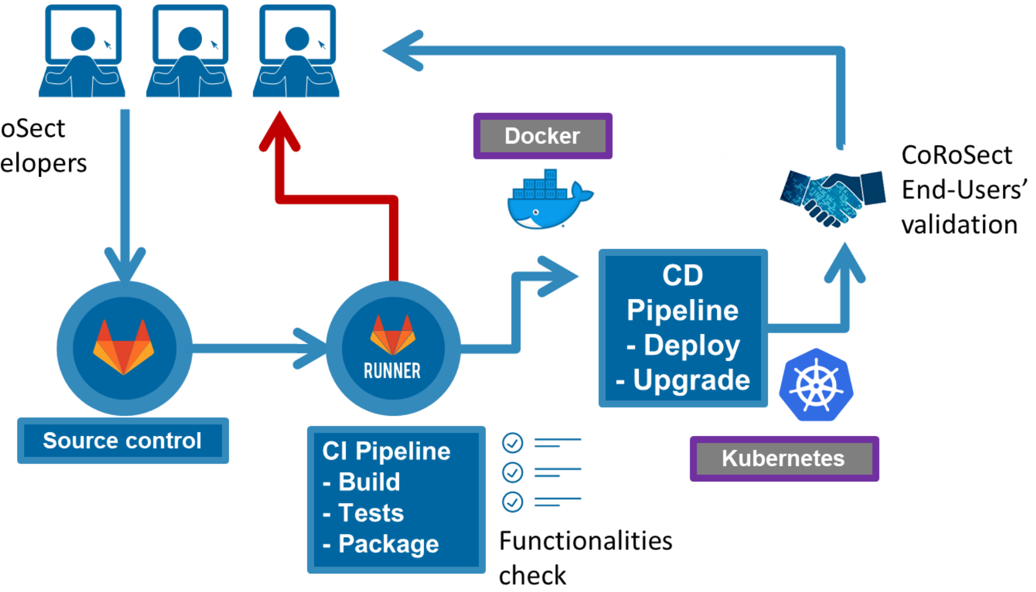


Figure 47. CoRoSect CI/CD proposed environment

6 Conclusions

This document has provided the initial System Architecture for the CoRoSect project, within the targets of Task 2.3 in WP2. This identifies the main components for CoRoSet implementation to be developed in the project's work packages and drives their integration. The specifications of this architecture have been driven by a set of requirements and insights extracted from standards-based reference architectures, IoT+Industry frameworks and CoRoSect's usage scenarios, and these are intended to foster the integration and interoperability of the heterogeneous mechatronics and IoT-based devices within a RAMI4.0 compliant framework.

This initial version of the architecture sets the guidelines to develop the digitization of CoRoSect's building blocks and starts the process for their whole integration, which will be carried out by WP9. As a first step to present the CoRoSect System, here is introduced a high-level reference architecture, which presents the Operational and Information Technologies (OT/IT) involved in CoRoSect's scenario and introduces the common functional layers that defines the system. Derived from this, the text goes a step beyond and develops this RA by identifying the relationships and dataflows between these main components. This way it provides the first versions of its logical, process and development views, which includes the needed integration interfaces for WP4, WP6 and WP7 components.

The alignment of the proposed architecture to the IoT and Industry standards has led to the reuse of some relevant concepts and building blocks from them. In particular, and aiming again at the RAMI4.0 compliance, it is remarked the concept of the Asset Administration Shell (AAS) for the digitization of each component to be integrated. In this sense, and coming from novel Industry 4.0 + IoT running frameworks, the deliverable suggests several open technologies to implement some functionalities addressed to specific components.

As mentioned, the outcomes of this deliverable: the RA, the system views, the interfaces, the AAS guidelines, and the proposed technologies will be used by technical and integration work packages (mainly WP4, WP5, WP6, WP7 and WP9) to develop the components and build the system. During these development and integration processes, new requirements, data flows and connections will arise, evolving this initial system architecture to its advanced version, depicted in D2.4 (Advanced system architecture)

7 References

[1] IEC 62264 / ISA-95, “Standard for Enterprise-Control System Integration”, 2013.

[2] “Enterprise-control system integration — Part 3: Activity models of manufacturing operations management”, *IEC 62264-3:2016,* 2016.

[3] D. K. Schweichhart, “Reference Architectural Model Industrie 4.0 (RAMI 4.0),” [Online]. Available: https://ec.europa.eu/futurium/en/system/files/ged/a2-schweichhart-reference_architectural_model_industrie_4.0_rami_4.0.pdf. [Accessed 23 12 2021].

[4] Industrial-process measurement, control and automation - Life-cycle-management for systems and components, IEC 62890:2020, 2020.

[5] D. S. 91345:2016-04, Reference Architecture Model Industrie 4.0 (RAMI4.0), 2016.

[6] P. D.-I. A. W. Colombo, Course Content Digitalization of Industrial Cyber-Physical Systems, University of Applied Sciences Emden/Leer, 2020.

[7] “Industrial Internet Consortium Reference Architecture (IIRA),” [Online]. Available: https://www.iiconsortium.org/IIRA.htm. [Accessed 23 12 2021].

[8] Bauer M. et al. (2013) IoT Reference Model. In: Bassi A. et al. (eds) Enabling Things to Talk. Springer, Berlin, Heidelberg..

[9] Refererence Architecture Model for the Industrial Data Space, Fraunhofer-Gesellschaft, 2017.

[10] “FIWARE Smart Industry,” [Online]. Available: https://www.fiware.org/community/smart-industry/. [Accessed 23 12 2021].

[11] “FIWARE,” [Online]. Available: www.fiware.org. [Accessed 23 12 2021].

[12] “FIWARE Technologies enabling Industry 4.0 (FIWARE for Industry),” [Online]. Available: https://www.fiware4industry.com/#. [Accessed 23 12 2021].

[13] “DIH^2 Accelerating factories through robotics,” [Online]. Available: http://www.dih-squared.eu/. [Accessed 23 12 2021].

[14] F. Melendez, “DIH2 Digital Platform 1.0,” [Online]. Available: http://www.dih-squared.eu/sites/default/files/D1.3.pdf. [Accessed 23 12 2021].

[15] AGVR, “Natural Navigation Automated Guided Vehicles,” [Online]. Available: https://www.agvnetwork.com/natural-navigation-automated-guided-vehicles. [Accessed 23 12 2021].

[16] R. A. R. D. A. K. M. t. H. Günther Schuh, “Industrie 4.0 Maturity Index. Managing the Digital Transformation of Companies,” 2020. [Online]. Available:

https://en.acatech.de/publication/industrie-4-0-maturity-index-update-2020/. [Accessed 23 12 2021].

[17] Meyer, H., Fuchs, F., Thiel, K., Manufacturing Execution Systems (MES): Opti-mal Design, Planning, and Deployment, McGraw-Hill Education, 2009.

[18] Szilárd Jaskó, Adrienn Skrop, Tibor Holczinger, Tibor Chován, János Abonyi, “Development of manufacturing execution systems in accordance with Industry 4.0 requirements: A review of standard- and ontology-based methodologies and tools,” *Computers in Industry,* vol. 123, 2020.

[19] “Industrie 4.0 Plug-and-Produce for Adaptable Factories,” 2017. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/Industrie-40-20Plug-and-Produce.html. [Accessed 23 12 2021].

[20] Xun Ye, Seung Ho Hong, “Toward Industry 4.0 Components: Insights Into and Implementation of Asset Administration Shells,” *IEEE Industrial Electronics Magazine,* vol. 13, no. 1, 2019.

[21] Erdal Tantik, Reiner Anderl, “Potentials of the Asset Administration Shell of Industrie 4.0 for Service-Oriented Business Models,” *Procedia CIRP,* vol. 64, pp. 363-368, 2017.

[22] J.A.Cortes Ortiz, A.T.Ruiz, J.A.Morales-Ramos, M.Thomas, M.G.Rojas, J.K.Tomberlin, L.Yi, R.Han, L.Giroud, R.L.Jullien, “Chapter 6 - Insect Mass Production Technologies,” *Insects as Sustainable Food Ingredients,* pp. 153-201, 2016.

[23] X. Ye, J. Jiang, C. Lee, N. Kim, M. Yu and S. H. Hong, “Toward the Plug-and-Produce Capability for Industry 4.0: An Asset Administration Shell Approach,” *IEEE Industrial Electronics Magazine,* vol. 14, no. 4, pp. 146-157, 2020.

[24] “Reference Architectural Model Industrie 4.0 (RAMI4.0) - An Introduction,” [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.html. [Accessed 23 12 2021].

[25] P. I. 4.0, “The Structure of the Administration Shell: TRILATERAL PERSPECTIVES from France Italy and Germany,” 2018. [Online]. Available: https://www.de.digital/DIGITAL/Redaktion/EN/Publikation/the-structure-of-the-administration-shell.pdf?__blob=publicationFile&v=3. [Accessed 23 12 2021].

[26] P. I. 4.0, “Details of the Asset Administration Shell,” 2020. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part1_V3.pdf?__blob=publicationFile&v=5. [Accessed 23 12 2021].

[27] P. I. 4.0, “Part 2 – Interoperability at Runtime – Exchanging Information via Application Programming Interfaces,” 2020. [Online]. Available: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Details_of_the_Asset_Administration_Shell_Part2_V1.html. [Accessed 23 12 2021].

[28] “Grafana: The open observability platform,” [Online]. Available: https://grafana.com/. [Accessed 23 12 2021].

[29] "OAuth 2.0," [Online]. Available: https://oauth.net/2/. [Accessed 23 12 2021].

CoRoSect

Maastricht University
CERTH
CENTRE FOR RESEARCH & TECHNOLOGY HELLAS
University of Applied Sciences
HOCHSCHULE EMDEN•LEER
Luke
LUONNONVARAKESKUS
tecnova
CENTRO TECNOLÓGICO
KU LEUVEN CiTiP
CENTRE FOR IT & IP LAW
Atos
Robotnik
AGV R
NASEKOMO
ENTOMOTECH
ENTOCYCLE
Italian Cricket farm
invertapro
FieldLab ROBOTICS
f/h
AgriFood DIH Lithuania
CIHEAM BARI
OAMK
OULU UNIVERSITY OF APPLIED SCIENCES

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101016953