D2.4. ADVANCED 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)	2022-12-31
Actual Delivery Date	2022-12-30
Abstract:	This text presents the Advanced (and final) CoRoSect's System architecture, evolved from the initial version (D2.3) and enhanced with technical WPs' achievements during last year. Its objective is to fully support all the functional and technical requirements needed to deploy and perform the CoRoSect's pilots and evaluate the CoRoSect's performance. It also details each layer's architecture and the CoRoSect's components integration architecture. This will support the CoRoSect's full system implementation and the integration of all its functional subsystems, addressed in WP9. It includes the guidelines for the system's scalability and replicability.

Document Revision History

Date	Version	Author/Contributor/ Reviewer	Summary of main changes
25/10/2022	0.1	ATOS	Table of contents
13/12/2022	0.2	ATOS, OAMK, CERTH, HSEL, AGVR, ROB	Contributions for components' architecture
15/12/2022	1.0	ATOS	Version for internal review
21/12/2022	2.0	ATOS, UM, CERTH	Version for delivery

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 document **evolves** the Initial System Architecture for the CoRoSect System, presented in D2.3, **towards the final (and advanced) architecture** that will lead the **CoRoSect System Implementation**. This evolution is the final step to provide a functional architecture for the CoRoSect scenario, which also be portable to other Smart Manufacturing environments.

Here is presented an overall and independent concept architecture for smart manufacturing environments, customised for the CoRoSect's scenario (insect's farms and rearing processes) and according to the project's requirements and objectives. This proposal follows the **RAMI4.0** guidelines to obtain an I4.0 compliance architecture, easily **replicable** and **adaptable** to similar scenarios and **extendable** to other manufacturing areas.

The Initial version of the CoRoSect Architecture was detailed in D2.3 and provides a quite reliable and coherent approach. This hasn't required deep modifications during this working year to cover all the addressed functionalities and integrate all the CoRoSect System components, so the content is focused on the details that **guide the reader through the final functionalities and integration capabilities of all the components and sub-systems of the main CoRoSect system**.

The intention is also to support the deployment of a functional instance of the whole system and enable the way to add new OT and IT components, expanding the system's capabilities.

This Advanced CoRoSect System architecture is directly connected with all the technical CoRoSect WPs, as it details how the integration of all the developed servers and adaptors that **interconnect** all the **robots at the shop floor**, and it supports the **WP9 integration** and implementation activities.

Following the **M18 reviewers' recommendations**, this document is part of the set formed by **D2.4** (Advanced System Architecture); **D9.1** (Integration Plan and Interfaces); **D9.2** (Integrated CoRoSect Platform R-I); and **D9.3** (Test cases for the CoRoSect Ecosystem); which has been aligned and revised to provide a joint and **concise description of the technical approaches** selected to build, deploy, and evaluate the **CoRoSect full system**.

List of Figures

List of Abbreviations and Acronyms	
AAS	Asset Administration Shell
AGV	Automated Guided Vehicle
AI	Artificial Intelligence
AR	Augmented Reality
CIA	Confidentiality, Integrity, and Availability
DMS	Data Management System
DoW	Document of Work
D-Robot	(Stacking) De-Stacking Robot
DS	Decision-Making System
DSS	Decision Support System
DX.Y	Deliverable X.Y
ERP	Enterprise Resource Planning
HRC	Human-Robot collaboration
ICF	Italian Cricket Farm
ID	Identification
IdM	Identity Manager
IIoT	Industrial Internet of Things
IMS	Information Management System
IoT	Internet of Things
IT	Information Technology
JSON	JavaScript Object Notation
Mx	Month X
MES	Manufacturing Execution System
ML/DL	Machine Learning / Deep Learning
MQTT	Message Queuing Telemetry Transport
M-Robot	Manipulation Robot
OPC-UA	OLE (Object Linking and Embedding) for Process Control-Unified Architecture
OSI	Open Systems Interconnection
OT	Operations Technology
PDP	Policy Decision Point
PEP	Policy Enforcement Point
Q&R	Query & Retrieve
RA	Reference Architecture
RAMI4.0	Reference Architecture Model for Industry 4.0
RBAC	Role-Based Access Control
RCS	Robot Control System
RM	Route Manager
ROS	Robot Operating System
SFM	Shop Floor Manager
SLAM	Simultaneous Localization And Mapping
SQL	Structured Query Language
WP	Work Package

1 Introduction

1.1 Scope and objectives of the deliverable

This document describes the evolution of the initial architecture, proposed in D2.3 and presented in M12, throughout one year long (till M24). This evolution is derived from the different technical WPs' achievements in terms of d-cells development and integration patterns, also considering the fulfilment of the functional and technical requirements initially identified (in M12) and raised during these developments.

As its scope, the document will so describe the final (called Advanced) architecture of the CoRoSect system that integrates all the components involved in the project's development: at the shop floor level (Operational Technologies); at the Management Execution System (Information Technologies); and at the CoRoSect's layers for Human and Robot safe collaboration environment and the Robot's actions planning and control.

The modelling of this system's architecture relies on the State-of-the-Art analysis done in D2.3 and the architectural views presented there, as they are perfectly applicable to the Advanced Architecture, to then focus on providing detailed logical views for the global advanced CoRoSect's architecture as well as for each of its layers. It reflects the chosen approach for building the CoRoSect's system, after one year of research and validation. It also analyses all the CoRoSect's components integration architectures, so, at the end, the reader will get a clear view of both: the overall architecture proposed for the CoRoSect system implementation (to be carried out in Task 9.2), addressing the main layers, functionalities, building blocks and WPs where these are to be developed; and then, the architecture of these CoRoSect layers, proposing common schemas for their components to get integrated according to RAMI4.0 guidelines.

In addition to present and describe the CoRoSect's Advanced System architecture, this document has two more relevant targets: a) to provide the reader with a pattern to integrate a new RAMI4.0 compliant solution that can interact with the rest of the CoRoSect components, and b) identify the endpoints and capabilities of the system to get connected and exploit the information sets captured from the shop floor level, useful to integrate CoRoSect with any other existing ERP software at the farms.

1.2 Relationships with other deliverables and tasks

The CoRoSect's advanced architecture proposed within this document collects information from all technical WP and those tasks that analyse different project's requirements and, in turn, creates and retrieves the schema where all the functional and technical CoRoSect components are mapped and integrated. Figure 1 summarises the inputs required and outputs provided by this Advanced System Architecture.

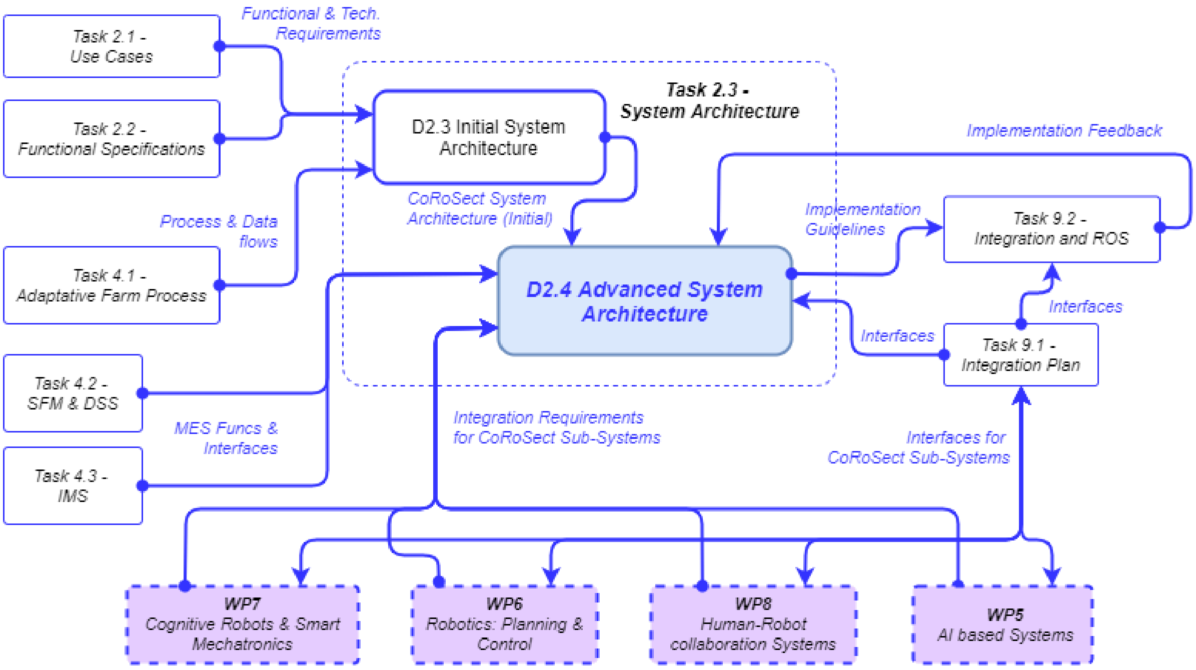

Figure 1: D2.4 dependences with other CoRoSect's tasks and work packages

Considering Figure 1, these contributions (inputs and outputs) and relationships are mainly derived from and contribute to:

- **WP2 (Use-cases, user requirements and system architectures)**: T2.1 defined the layers in which the tasks will be executed; T2.2. defined the functionalities (functional requirements) and T2.3 elaborates on this and provides a structure (architecture) for incorporation of functionalities across the layers. All of these are shown in:
    - D2.3 Initial System Architecture (M12) [1] and D2.4 Advanced System Architecture (M24) [2].
- **WP4 (Farm-level modelling and orchestration)**: provides the incomes for the SFM and DSS. In turn, D2.4 introduces the architecture for the IMS.
    - D4.2 Data analytics to obtain the prediction models (M18) [3].
    - D4.3 IMS Implementation (M30).
- **WP5 (AI-enabled perception methods)**: D2.4 interacts with WP5 and introduces the architecture for the Objects detector and the VR tools for the Augmented Reality (AR) simulators.
    - D5.1 Object detection methods for environment analysis (M30).
    - D5.2 Tools for natural interaction with the VR environment (M30).
- **WP6 (Robotic actions planning and control)** and **WP7 (Cognitive robots and smart mechatronics)**: provide requirements and support the architecture for Shop Floor components integration.
    - D6.1 Documentation of control for handling of crates (M12) [4].
    - D6.2 Documentation of control for insect handling (M12 and M24) [5].
    - D6.4 Safety concept and control in robotic systems (M12 and M24) [6].
    - D7.1 Sensoring solution to support insect rearing process automation (M30)
    - D7.2 Report on and documentation of robot cell for handling crates (M12 and M24) [7].

    - D7.3 Report on and documentation of robot cells for object manipulation and feeding (M12 and M24).
    - D7.4 Report on and documentation of AGV for insect farms (M30)
- **WP8 (Human-robot collaboration schemes)**: requirements and architecture for the Route Manager.
    - D8.2 Autonomous and human-aware robot trajectory plan for safe and efficient HRC (M32)
- **WP9 (Secure platform integration)**: architecture of the CoRoSect system instances.
    - D9.1 Integration plan and definition of the interfaces (M24) [8].
    - D9.2 Integrated CoRoSect Platform (M24 and M34) [9]

1.3 Structure of the deliverable

The main target of this document is to present to the reader the final and advanced architecture designed and developed within CoRoSect to fulfil all its objectives at project's level, which in turn will be used to deploy de CoRoSect system. **This architecture is the evolution of the one introduced in D2.3** and so, the text will rely on that initial architecture and its basis to focus on clearly expose the final results. According to this, the structure of this deliverable is as follows:

- Section 2 (**functionalities**) will present and update the set of capabilities addressed by the CoRoSect system to cover all functional and technical requirements extracted from WP2 and which must be mapped by the CoRoSect's architecture. This will produce and updated CoRoSect's Reference Architecture (RA).
- Section 3 (**advance system architecture**) takes the previous (initial) architecture from D2.3 and the new CoRoSect's RA and describes the final functional views for the different layers and for the components within, showing the architecture required to integrate new components and the entry points to interact with the system, always relying on RAMI4.0 and SoA paradigm. It also paves the way to extend the system and introduce new technologies.
- Section 4 (**conclusions**) closes the deliverable and the CoRoSect's Architecture proposal with the summary of the achievements and lessons learnt extracted from these two years working on a RAMI4.0 compliant architecture for CoRoSect system, providing the guidelines for the CoRoSect's system deployment in WP9 and evaluation in WP10.

2 CoRoSect System's Functionalities

Taking up the approach already presented in D2.3 derived from the previous work done in Tasks 2.1 and 2.2 related to the end users' (farms) requirements and initial DoW targets, this section summarises the generic scenario to be covered by the CoRoSect's system and the requirements it should address to build on top the final CoRoSect's Reference Architecture. This will reflect the system's layers and depict the integration among them, as starting point to later provide the advanced CoRoSect's system architecture from the initial one.

2.1 Usage scenarios

As written in the DoW, *CoRoSect project provides 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*. The integration of heterogeneous cyber-physical systems, according to a homogeneous digital model (known as Digital Twin) and managed by a specifically developed set of software components which controls and improves the insects' rearing processes, all according to RAMI4.0, is, in very main lines, **the scenarios' set up** to run the WP10 CoRoSect pilots. These will be conducted through the four involved farms: **ICF** in Italy; **Entocycle** in UK; **Nasekomo** in Bulgaria; and **Invertapro** in Norway, plus the l configuration of the pre-pilot hosted in Entomotech in Spain.

Figure 2 illustrates this CoRoSect scenario's baseline and sets the layout where all the CoRoSect's pilots will be executed. Industry 4.0 has come to transform the manufacturing processes by implementing new digital technologies and novel IoT infrastructures, creating major optimisations in terms of manual labour minimisation, precise monitoring and control and new capabilities of planning and management, hence improving business activities outcome. I4.0 also pursues to converge the traditional IT (information technology) and OT (operations technology) layers, what is also a target of the CoRoSect's architecture.

CoRoSect's **OT layer** is composed by the project's shop floor, the corresponding hardware (with embedded manufacturers software) controllers and the shop floor network infrastructure that enables the interconnection between shop floor components themselves and with the IT layer.

- The project's **shop floor** contains all the robots, mechatronics, hardware devices and IoT infrastructures needed to build and perform the pilots. These are:
    - The **I-Crates**, as the place covered by IoT sensors that monitor the status of the environmental parameters for the insects rearing.
    - The **insects' handling cell** comprises i) the visualization and inspection module to capture images of the i-crates; ii) the tools attached to the core robot to help feeding and actuating with the insects in the crate; and iii) the Manipulation robot (M-Robot) to support the cell.
    - The **crates' handling cell** moves a selected i-crate from the rearing shelve to the AGV to harvest or to get cleaned. It is articulated by the stacking/de-stacking robot (D-Robot) with its attached gripping tool.
    - The **Automated Guided Vehicle (AGV)** transports the crates between the rearing area to the harvesting area or to the cleaning zone. It uses the SLAM component to map the trajectories between the zones and interacts with the IT Layer Route Manager to get the paths.
- The **controllers** for each component at the shop floor directly connect the hardware components to the network and open the door for the digital data gathering and commands

injection. These hardware/software components enable the remote digital command and control of the shop floor. One single controller can manage only one or several connected devices, depending on the manufacturer and on the applied configuration.

The **shop floor network** connects the OT and IT layers, allowing interactions between them. In the CoRoSect's use cases, this will be an ethernet network that will also support wireless (WiFi) connectivity for those shop floor components that requires it, as the AVGs.

In turn, the scenario's **IT layer** covers the components and binds them with the Data Layer, defined in D.2.1. Use Cases Description, to homogeneously digitalize the shop floor, to control the manufacturing processes and to exploit the collected information according to CoRoSect's objectives. These are:

- The shop floor's (and any other system component's) **digital twins**, in the form of I4.0 Asset Administration Shells, which, among others, define the interfaces to interact (commands) with the shop floor asset and to get information from the resource (properties or attributes).
- The **management level** implements all the shop floor management and control functionalities specific to the manufacturing process plus the new enhanced ones and those directly addressed to the CoRoSect's objectives:
    - The **data management** module to collect, store and distribute all the information from the shop floor components and from any other integrated data source.
    - The **process management** component that executes and controls all the shop floor processes
    - The **decision-making** module reads and evaluates the shop floor status to assist the process management module on the efficiency enhancement of the manufacturing process.
    - Then, the rest of the specific CoRoSect's modules read data from the data management module to execute concrete functionalities to interact with the shop floor: Object's detection, routing management, ML/DL algorithms development and training for simulation tools and predictive analysis, etc.

On the top of the CoRoSet's scenario, the system provides a set of services and endpoints (this is a SoA architecture) for the farms' ERP (and any other existing data exploitation software) application to get integrated and extract and exploit information managed by the CoRoSect's system. These are the System's external interfaces.

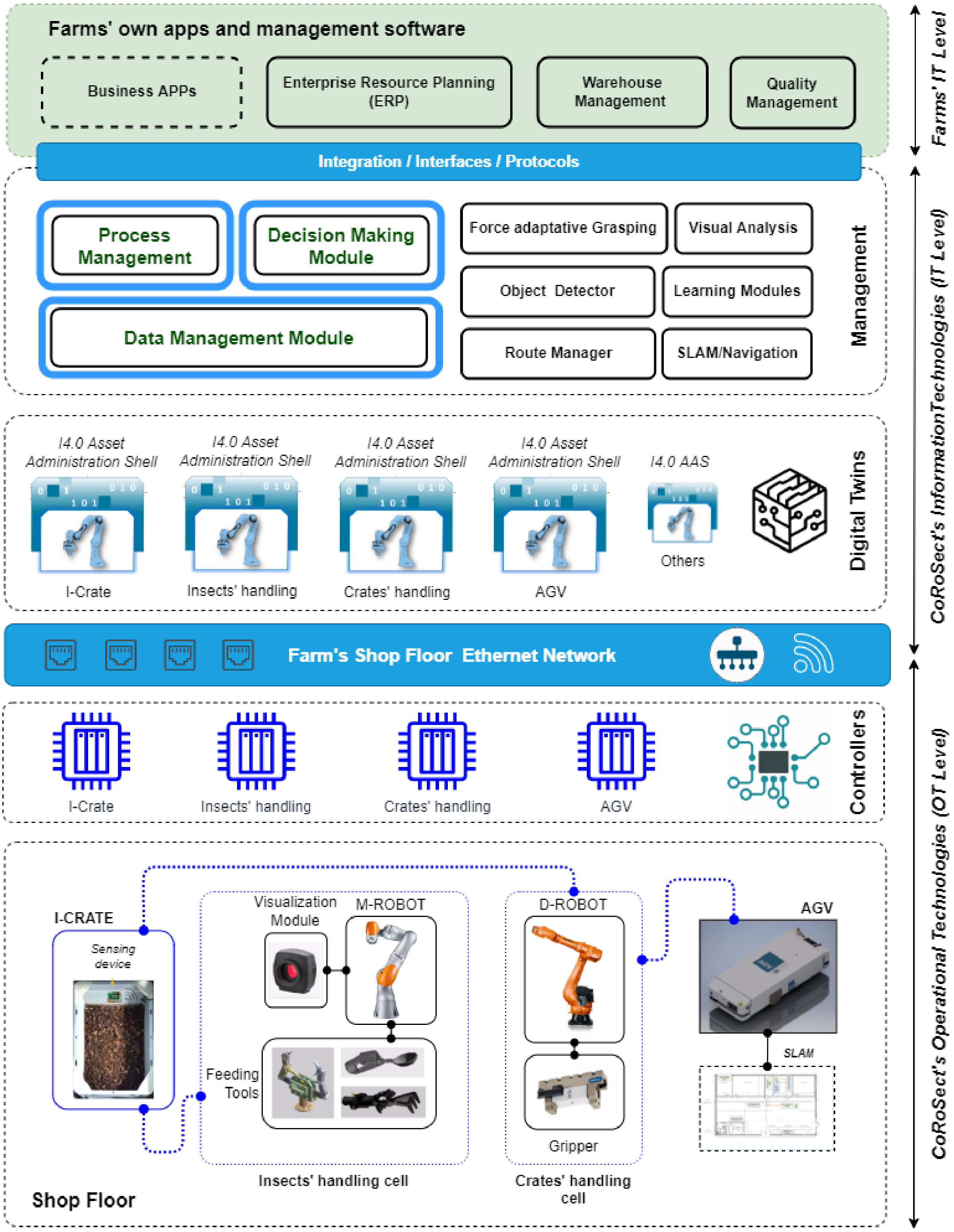

Figure 2: CoRoSect's common usage scenarios' set up

2.2 Reference Architecture

The usage scenario in Figure 2 has no deep variations from the one described in D2.3. The progresses in the pilots' definition and the end-users scenarios (farms), as well as the in the d-cells development and IT level building blocks during this one-year period (M12-M24) have no significant deviations from the Reference Architecture (RA) depicted in D2.3. This section just introduces minor updates on this original RA to match the final system implementation and remark the WP where each component is developed. The final CoRoSect's RA is presented in Figure 3.

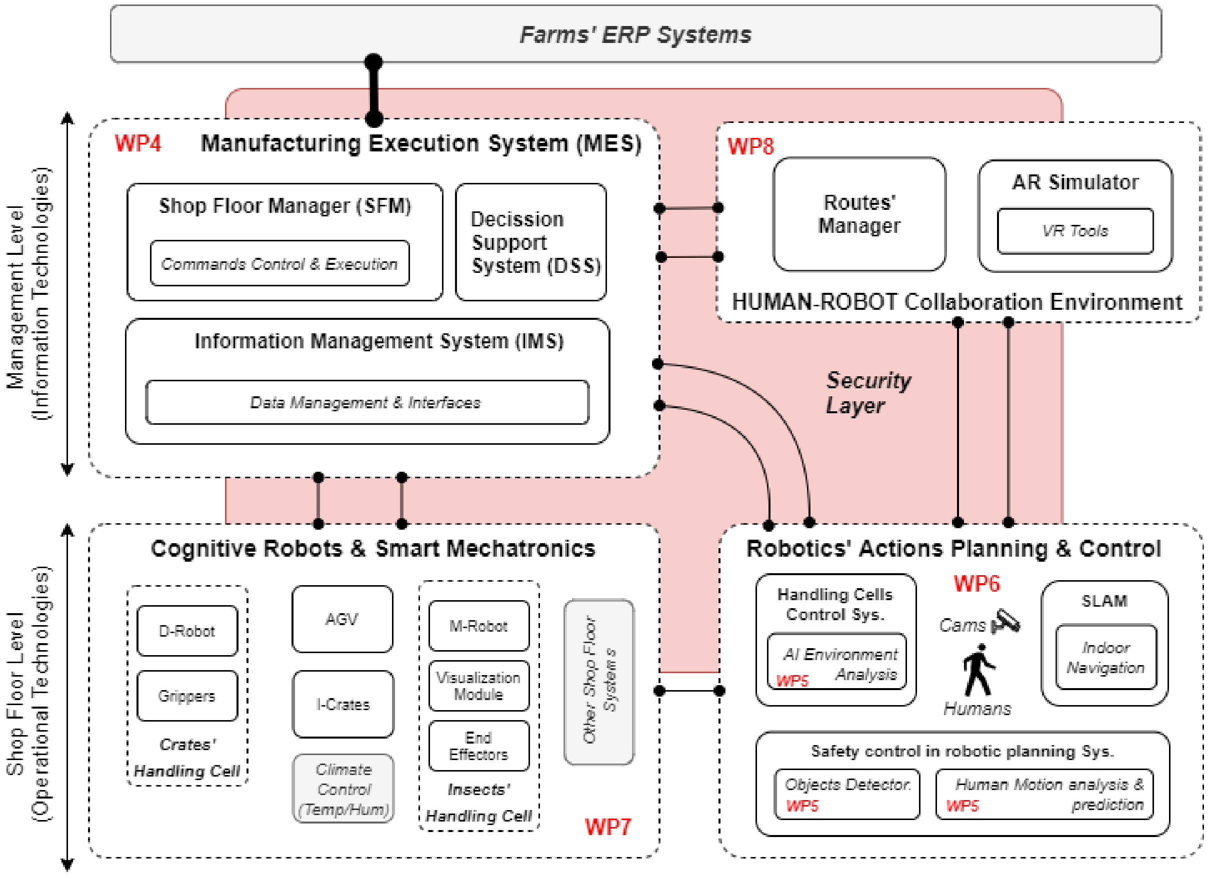

Figure 3: D2.4 CoRoSect Advance Reference Architecture

This reviewed CoRoSect's RA version aggregates the technical WP outcomes into specific building blocks, directly related to the CoRoSect's usage scenario, and depicts the interconnections, from a high-level point of view, between them. This approach provides a structure to easily describe the components of the advanced architecture and present the detailed integration of each asset in the following sections. The main building blocks are the same described in D2.3, so detailed descriptions can be found there.

Cognitive Robots and Smart mechatronics block covers the OT layer with the d-Cells and assets that compose the Shop Floor. It includes the Crates Handling Cell, the Insects Handling Cell, the AGV and the I-Crates. Any other additional asset, information system or new IoT infrastructure should be included here. The development of these CoRoSect's components is mainly carried out in **WP7**.

Manufacturing Execution System (MES) within I4.0 is the responsible of monitoring, controlling, and optimizing the manufacturing processes. In CoRoSect, this building block also collects and concentrates all the system's information to be later distributed in a homogeneous way and according to I4.0 Asset Administration Shells (defined in D9.1). All this information is used internally to the MES to analyse the shop floor status and processes carried out to support decision-making and assist the shop floor management in processes efficiency improvement. All tasks related to the MES development are reported in **WP4**.

Robotics' Actions Planning and Control level develops and implements the controllers and additional CoRoSect's components to execute the commands received from the MES and those automatic actions inherent to the shop floor level that guarantees the proper control and planning of the manufacturing process. These components include the digital specific controllers for the handling cells plus assisting components, such as the AI models, objects' detector or human movement predictor that enhance this planning and control activities. Here is also included the SLAM [10] navigation module for supporting the AGVs and the routing management. This all is described in **WP5** and **WP6** crossed tasks. This layer is connected to the MES to both, receive commands from the shop floor manager and read data from the IMS to get its current context (status of the surrounding devices needed to complete an action).

Human-Robot collaboration environment permanently monitors the status of the whole CoRoSect's systems through the MES to assist on the interactions between robots themselves and between robots and humans at the shop floor. In CoRoSect project, this level develops the routes manager to plan the routes for the AGV according to the current context of the shop floor and the envisioned human obstacles; and the augmented reality simulation environment to train both, users and robots I the manufacturing process. This is all developed in **WP8**, but in close collaboration with WP5, WP6 and WP7.

2.3 Functional and Non-Functional Requirements

Functional requirements group the set of core functionalities the CoRoSect solution must provide and/or support to guarantee the proper performance of the envisioned pilots and accomplish project's objectives. These were mainly extracted from Task 2.2 (M10) in a first row and shown on D2.3 (M12), to be later checked and validated through the development of technical WPs. The full list in D2.3 has no relevant variations since no new functional requirements appeared during this period (M12-M24). In summary, and addressed to CoRoSect functional components, there are:

Process Management System (PMS) → MES Shop Floor Manager (SFM)

1. The SFM implements an interface to trigger specific processes or tasks (can be triggered from any operator/system)
2. The SFM requests the respective list of tasks and its subtasks from the DSS
3. The SFM controls and executes the list of tasks
4. The SFM receives and manages error messages from controllers at the Shop Floor level
5. The SFM generates commands IDs to track commands flow through the IMS

Decision-Making System (DS) → MES Decision Support System (DSS)

1. DSS collects/receives data from the IMS related to other system components to be internally evaluated.
2. DSS provides the logic behind decisions made when comparing values, examining conditions within the system.
3. DSS instructs SFM to execute/trigger certain processes/tasks depending on the outcome of a condition, i.e., decision made.
4. DSS has a Graphical User Interface (GUI) to show decisions made to the users.

Data Management (DMS) → MES Information Management System (IMS)

1. The IMS manages the Digital Twin structure of a defined system component
2. The IMS stores data stemming (context information) from Shop Floor Components (OT Level)
3. The IMS stores historical data produced by Shop Floor Components (OT Level).

4. The IMS stores threshold values entered by an operator.
5. The IMS can record and serve the ID of the task and the status of a Shop Floor controller
6. The IMS can store lists of tasks and subtasks
7. The IMS provides data to other components (IT Level) upon request (synchronous data query/retrieve)
8. The IMS implements publish/subscribe mechanisms to distribute data among other components – IT Level (asynchronous data query/retrieve)
9. The IMS implements commands delivery and tracking, using Digital Twins
10. The IMS implements commands history record

For ALL Shop Floor Components (OT Level)

1. Implement integration with IMS (RAMI4.0 compliant)
2. Capture data from embedded sensors and sub-systems and send them to the IMS
3. Receive from IMS and execute supported commands
4. Implement and support RAMI4.0 Asset Administration Shell
5. Implement a heartbeat messaging service

Non-Functional requirements related to architecture have to do with technical details (also known as technical requirements) of the whole system deployment and performance. These have been also presented in D2.3 and here, two specific ones: **scalability** and **replicability**, are further described in section 3.6.

Further details and explanations of the CoRoSect's functional requirements are provided in the reviewed D10.2 Pilot Preparation and planning [11].

3 Advanced CoRoSect System’s Architecture

The CoRoSect’s system architecture proposed in D2.3 was the result of an exhaustive analysis of the existing smart manufacturing approaches and the matches with the CoRoSect’s functional and technical requirements. Together with the RAMI4.0 compliance, as one of the main technical targets of the project, this initial architecture has demonstrated to be perfectly valid as we progressed towards its instantiation during this last period (M12-M24), and flexible enough to integrate all the CoRoSect’s components plus any new system to interact with. This section goes a step beyond this D2.3 initial architecture, relying on the presented architecture’s views, to show the final integration schemas (Figure 4) for all the system’s components. It also includes the CoRoSect System components’ deployment and integration architectures.

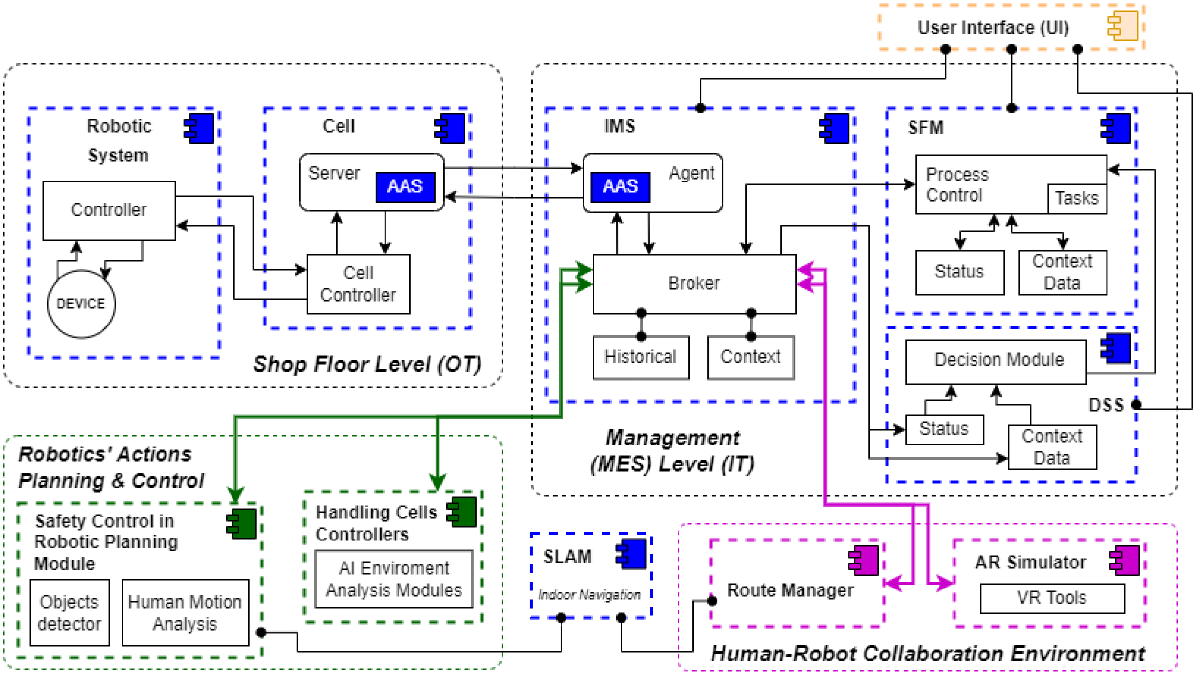

Figure 4: Updated (from D2.3) CoRoSect’s Advanced System Logical View

3.1 Shop Floor level components

As described in D2.3, the Shop Floor in CoRoSect comprises, according to the traditional manufacturing definition, the area where all the insects’ rearing processes take place plus all involved the physical systems (robots, mechatronics, vehicles, etc.) and workers. Going further, the CoRoSect’s Smart Shop floor includes also IoT infrastructures (such as the I-Crates) and supports interconnected devices to build a safe I4.0 Human – Robot collaboration environment.

To achieve this, the CoRoSect’s architecture proposes a common approach to connect and integrate the heterogeneous OT (Operational Technologies) layer of the Shop Floor (Figure 5) with the IT (Information Technologies) and defines two main integration paths: data gathering (Figure 6) and command/response flows (Figure 7).

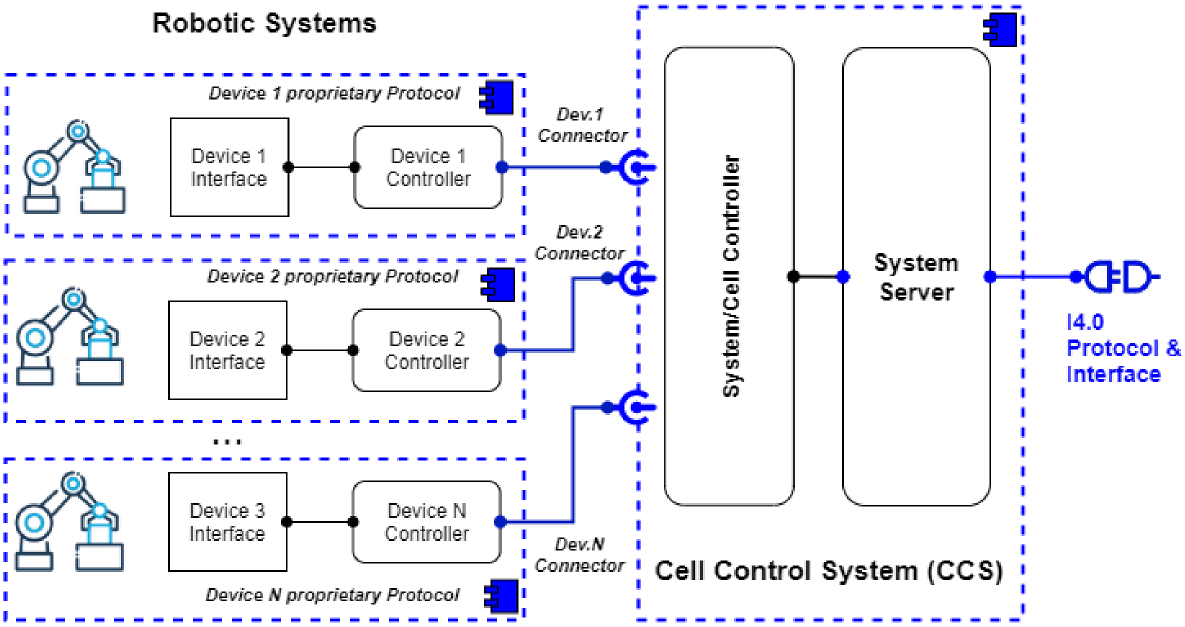

Figure 5: General Integration architecture for CoRoSect's Shop Floor components (Logical View)

From the CoRoSect's perspective, each shop floor component (included here the robots, mechatronic devices, IoT components or specific data sources) can be depicted as the connection of two main systems: the Robotic System and the Cell Control System. This approach, also followed by D2.3, is used by CoRoSect to describe the architecture and the integration of its assets at this level. In the same way, it is also proposed to map and integrate any new shop floor device into the CoRoSect architecture. According to this, a CoRoSect's shop floor component (e.g. handling cells, IoT system or AGVs) would be composed by one or several devices (robot, camera, gripper…) managed by a single control system:

- **Robotic System**: is the combination of the N devices required to fulfil all the component's functionalities. As in D2.3, each device is defined by the physical functional device (e.g., the robot or the gripper); the interface, including cable, wireless connection, plug and the protocol (levels 1-3 of the OSI architecture [12]) to connect the physical device to the specific controller (provided usually by the manufacturer); and the own device controller which enable the control of the device using the manufacturer's protocol. This device controller would implement the proprietary command and control interface for the physical device and provide a connection with the whole cell (or system) controller.
- **Cell Control System**: concentrates the management of the whole set of devices that conforms the cell in a common single point, enabling the interconnection with the CoRoSect's MES. It has two main building blocks: the proper system or cell controller; and the system server:
    - The **Cell Controller** is connected to all the single device controllers to manage all of them. On one side, it receives the information from all its devices and aggregates all into a data packet to be delivered to the system server. On the other side, is able to receive, format and forward to the corresponding device a given command to execute a concrete action. It works as a hub for the N-devices and as a data homogenisation point.
    - The **System Server** is developed within CoRoSect and collects from the cell controller all the data from its devices. It formats all the data according to the interface (Asset Administration Shell) defined for the whole cell (or asset) and sends this to the MES using a RAMI4.0 compliant protocol. On the other way around, it implements the

RAMI4.0 channel to receive a command form the MES, using its AAS, and process it using its cell controller.

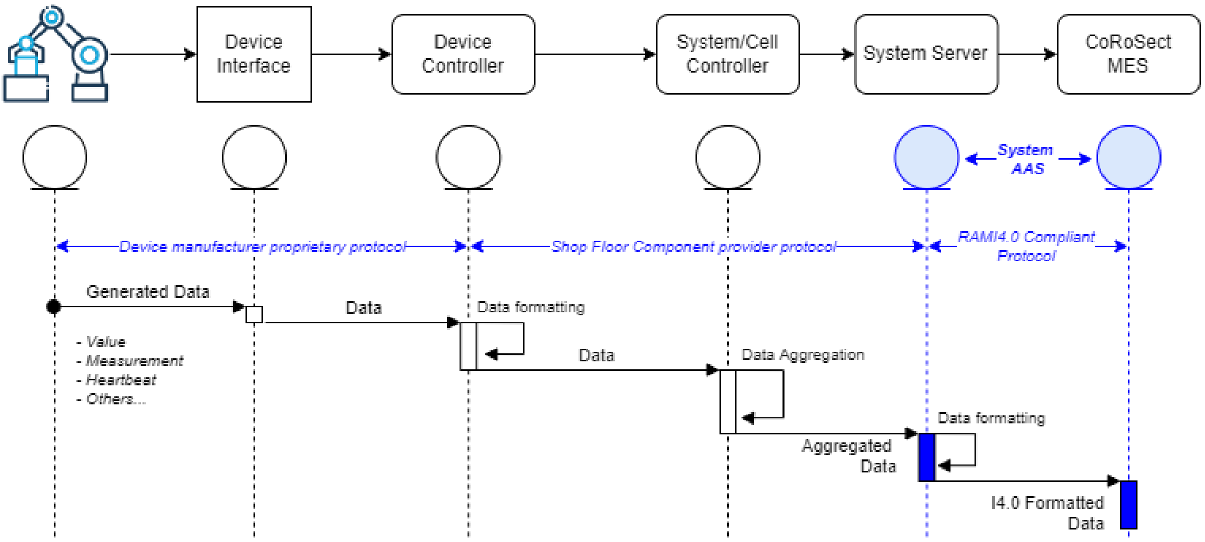

Figure 6: Common data updating process for shop floor components (process view)

Figure 6 depicts the common example about how data is updated in the MES from the Shop floor level: i) data is first generated at the physical device (e.g. a measured temperature value, a rotation status, a signal of error, etc.) and sent to the device controller through the device interface; ii) this raw value is formatted into a digital value at the device controller and sent to the cell controller; iii) then, in the Cell controller, this data is aggregated with other devices' data and with other context information (such as date and time) and stored in a single registry to be sent to the System Server; iv) the System Server updates its Asset Administration Shell information and sends it to the CoRoSect's MES, where the information is stored and distributed where requested.

Figure 7 complements this process view with the commands flow example: in this case, i) the command is generated in the CoRoSect's MES (by the Shop Floor Manager), forwarded by the IMS to the corresponding System Server (the one connected to the asset in charge of executing the final action). This is done using the defined AAS model and the selected RAMI4.0 compliant protocols that supports that AAS. ii) Then system server processes the command and extracts the commands parameters to send this to its cell controller, which, in turn; iii) formats the command's payload of the according to the device's proprietary and drives it to the corresponding device controller. iv) The device controller uses the device's proper interface to send the addressed command directly to the hardware device where it is finally executed. Once the corresponding response of the command is ready (at the hardware device), it travels back to the MES following a similar path to Figure 6. This response can be the execution results, an error, an Ack message, or any other dataset covered in the device's provider manual. In any of the previous steps, and Ack message can be configured (this is optional) to report the MES the proper addressing of the command.

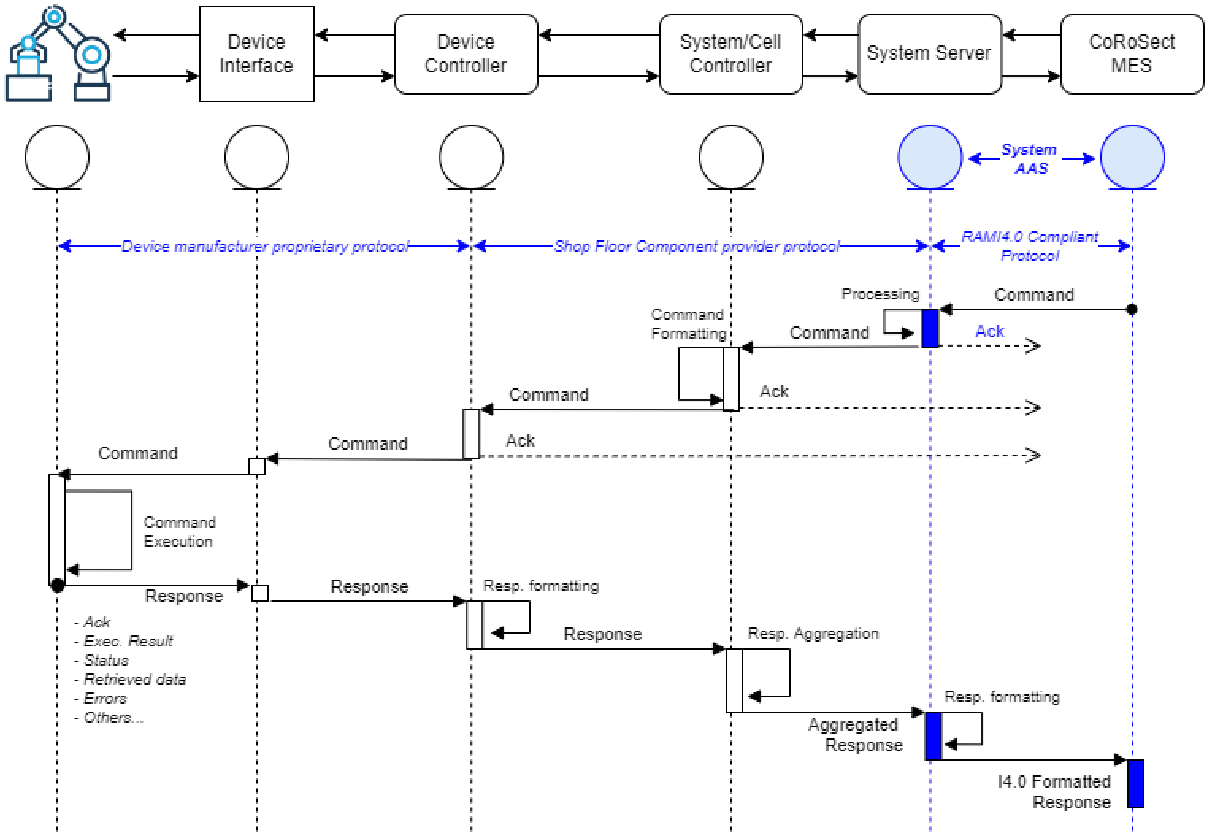

Figure 7: Common command/response process for shop floor components (process view)

3.1.1 Stacking/De-staking Robot – Crates' handling cell

In end-user facilities, crates filled with insects are stored in stacks to optimize the space required and the transportation needs. Crates must be individually de-stacked before the insects' handling cell (with the Manipulation Robot as its core) proceeds with its tasks. In the same way, they must be stacked again before they are transported to the storage area by the AGV. The Crates' handling cell's main objective is to move crates and boxes securely and robustly from the pallet with stacked crates, into the table of operation, and back again into the empty pallet to be picked by the AGV.

In this sense, the Staking/De-stacking Robot (D-Robot) is a core component in the CoRoSect's Robotic Cell and is equipped with sensors and robust controllers to manipulate crates filled with insects in a safe way. To make the system robust the D-Robot is equipped with a custom compliant gripper, sensors, and visual servoing algorithms that enable it to overcome small deviations in crate positioning. These all components conform the Crates' handling cell (also known as the CoRoSect's D-Robot), which main **functionality** is divided into two procedures:

- **De-stacking procedure**: D-robot picks a crate/box from the input's pallet and places it in operation table. The AGV transports small crates and big boxes stacked in pallets, which are picked by the D-Robot one at a time and placed in the area of manipulation. The input to the de-stacking procedure is the grid position (x,y,z) of the crate to de-stack from the input pallet. Once the D-Robot is commanded to de-stack a crate -> Robotic arm will move to a position close to the input pallet –> Robotic arm will move to a position close to the desired (x,y,z) position in that pallet -> Visual servoing will be used to locate and centre the arm with respect to the desired crate –> Gripper will open to grab the crate -> Robotic arm will move to a position close to table -> Robotic arm will place crate in table and close gripper to release crate -> Robotic arm will move away from table.

- **Stacking procedure**: D-Robot picks a crate/box from the operations' table and places it in output pallet. Once the M-Robot manipulation operation is finished, the D-robot picks the crate or box from the area of manipulation and stacks it again in the output pallet. Once the output pallet is filled it will be transported by the AGV to the storage position. The input to the stacking procedure is the grid position (x,y,z) where to place the crate in the output pallet. Once the D-Robot is commanded to stack a crate -> Robotic arm will move to a position close to the table –> Visual servoing will be used to locate and centre the arm with respect to the crate -> Gripper will open to grab the crate -> Robotic arm will move to a position close to the output pallet –> Robotic arm will move to a position close to the desired (x,y,z) position in that pallet -> Robotic arm will place crate in position and close gripper to release crate -> Robotic arm will move away from position.

The Crates' handling cell (or D-Robot) is **composed** by:

- A robotic arm.
- A pneumatic gripper with custom jaws for each crate type.
- Sensors for visual servoing.
- Computer with HMI.

The **D-Robot integration architecture** is show in Figure 8 and follows the general approach proposed for CoRoSect Shop Floor devices. It is divided in:

- **D-Robot Device System**, composed by the Kuka KR70 2100[1], an industrial robot arm with a reach of 2100mm and a rated payload of 70 Kg, enough to handle the small crates and big boxes at the required distance. It is complemented with:
    - Schunk PSH52[2] pneumatic gripper to grab the crates. Custom compliant jaws to handle each type of crate securely have been developed. The gripper is actuated by a pneumatic system, it is controlled by a standard IO system.
    - Lasers and RGB-D Cameras are used as visual sensors for servoing, where the data stream produced is connected through USB to the visual sensor controller.

  Further details on the D-Robot Device System and its implementation within the CoRoSect System is described in D9.2 Section 3.2.1.

- **D-Robot Control System (D-Robot CS)** which manages the cell status (number of crates in each stack, safety signals, emergency stop, robot moving...) and the different available D-Robot operations. It is composed by:
    - The D-Robot Controller, which has been implemented using ROS [13] and MoveIt [14]. It is a custom manipulation software which computes and executes collision free trajectories to pick and place the crates has been developed. A visual servoing system to handle small deviations in the positioning of the crates is also integrated. Trajectory and correction commands are sent by the D-Robot controller to the KUKA arm through the proprietary RSI interface.
    - OPC-UA [15] Server: OPC-UA is the communication protocol selected to integrate the D-Robot, according to RAMI4.0 recommendations and CoRoSect's targets. D-Robot OPC-UA Server, implements and exposes the D-Robot AAS defined in Task 9.1 (D9.1).

---

[1] https://www.kuka.com/-/media/kuka-downloads/imported/6b77eecacfe542d3b736af377562ecaa/0000332117_en.pdf

[2] https://schunk.com/de/en/gripping-systems/parallel-gripper/psh/psh-52-1/p/000000000000302152

It also supports the OPC-UA protocol implementation on the Server side. The D-robot provides one AAS for the whole Control System, as the internal components (gripper and sensors) are managed internally and therefore not interfaced. Robotnik has developed the D-Robot OPC-UA server using ROS. The server exposes to the IMS the different command actions available in the D-Robot, it also updates the status of the D-Robot dynamically.

Further details on the D-Robot Control System and its implementation within the CoRoSect System is described in D9.2 Section 3.2.1.

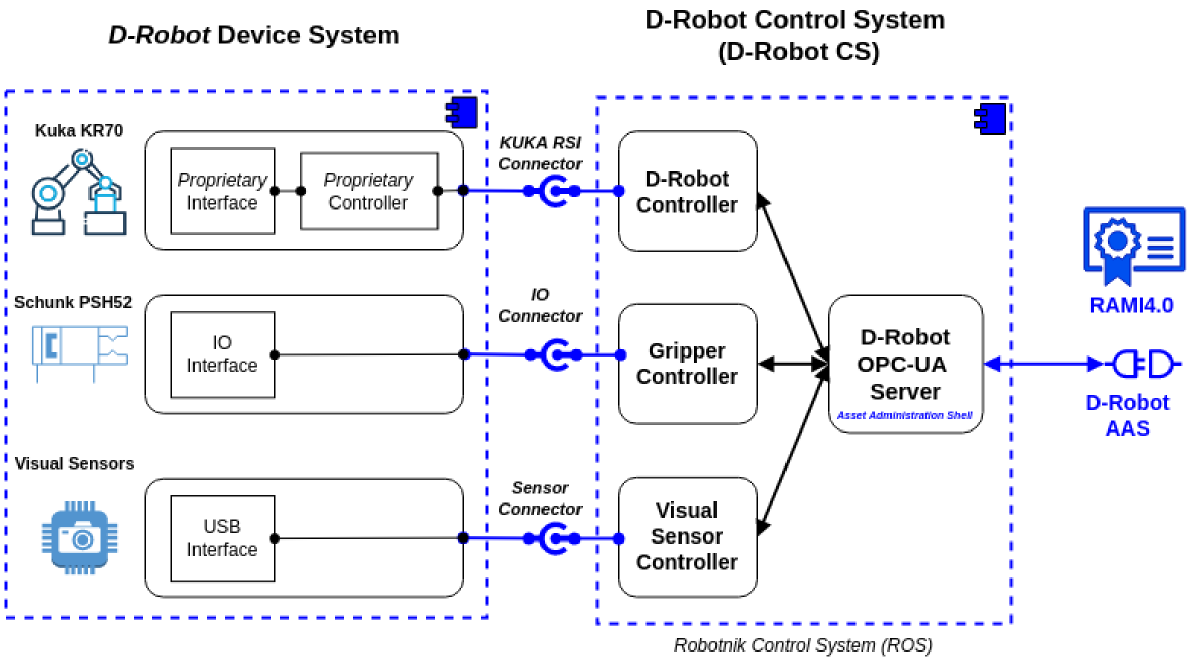

Figure 8: Stacking/De-stacking Robot (D-Robot) – Crates' handling cell – integration architecture

3.1.2 Manipulation Robot – Insects' handling cell

Core tasks during the insect rearing processes involve the manipulation of the insects (e.g. picking, placing, sorting), feeding the insects (e.g. adding feed to insect crates), the monitoring of the insects (e.g. visual monitoring of growth), and material handling for manipulating the insects' environments (e.g. adding/removing support structures into/from crates). To fulfil these tasks, CoRoSect has developed a versatile manipulation robot, the M-Robot.

The M-Robot manipulates the content of (de-stacked) crates. Crates are transported to and from the M-Robot by the CoRoSect Staking/De-stacking Robots (D-Robot) so that a M-Robot and a D-Robot form an integrated robot cell. Material such as feeding devices and support structures can be transported to and from the M-Robot through an autonomous guided vehicle (AGV).

The M-Robot integration architecture is shown Figure 9 and follows the general approach proposed for CoRoSect Shop Floor devices. Components of the architecture include:

- **M-Robot Device System**, composed of the KUKA LBR iiwa 14, a collaborative robot arm with a payload of 14kg, developed for safe human-robot collaboration. This robot is controlled through the KUKA sunrise controller that allows precise control of the robot's joints and provides access to the robot's torque sensors. To operate autonomously and safely, the KUKA robot arm is equipped with

  - Custom-made **End-Effectors** for insect and material handling. End-Effectors include e.g. a compliant gripper that is actuated by pneumatic system.
  - a **Visual Inspection Sensor**. This high-resolution camera is used for visual inspections of insects for automatic quality control. The Visual Inspection Sensor is guided by the M-Robot for inspecting areas within the crate that are difficult to reach or occluded.
  - **External Visual Sensor(s)** for safe human-robot interaction and visual servoing. The External Visual Sensors are used to detect and avoid collisions with obstacles and humans in the operational space of the robot. Movement trajectories of the robot arm are automatically calculated and adjusted based on the visual feedback provided by the External Visual Sensors.
- **M-Robot Control System (M-Robot CS)** which manages the cell status (status of crate serviced by the robot, state of robot, state of end effectors, safety signals, emergency stop, ...) and the different available M-Robot operations. It is composed of:
  - the **M-Robot Controller**, which has been implemented using ROS [13] and MoveIt! [14]. MoveIt! is an open-source tool that allows robot trajectory path planning taking into account the robot environment model generated and updated with the help of the External Visual Sensors. This model that contains information on the placement and structure of crates is used for collision avoidance. A custom manipulation software which computes and executes collision free trajectories to manipulated insects and material has been developed. Trajectory commands are sent by the M-Robot controller to the KUKA arm through the KUKA interface.
  - the **End-Effector Controller**, which interfaces to the End-Effectors mounted on the KUKA robot arm and its extensions. The End-Effector Controller monitors and controls the state of the End-Effectors such as controlling the opening and closing of grippers.
  - the **Visual Inspection Controller** controls and obtains images from the Visual Inspection Sensor. Given two consecutive image frames that depict the surface of a crate in which insects of various lifecycle stages are lying on, a small-object detection (semantic segmentation) system is developed, that provides high-level semantic predictions to characterize the detected population of insects. Automatically extracted information such as anomaly detection and insect counting will be exploited to enhance the quality management of the insect farms.
  - the **External Sensor Controller** controls and obtains data from the External Visual Sensor(s). Sensor data, such as RGB and depth images, is being used for the automatic guidance and obstacle avoidance of the robot arm as well as for enhancing the safe human-robot interaction. For this the External Sensor Controller extracts semantic information from the obtained images and generates a model of the robot's workspace including the position and content of the crates.
  - **M-Robot OPC-UA Server**: OPC-UA is the communication protocol selected to integrate the M-Robot, according to RAMI4.0 recommendations and CoRoSect's targets. The M-Robot OPC-UA Server implements and exposes the M-Robot AAS defined in D9.1. It also supports the OPC-UA protocol implementation on the Server side. The M-robot provides one AAS for the whole Control System, as the internal components (End Effectors and Sensors) are managed internally and therefore not interfaced. The M-Robot OPC-UA server exposes to the IMS the different command actions available in the M-Robot and updates the status of the M-Robot dynamically.

Further details on the M-Robot Device and Control Systems and their implementation within the CoRoSect System are described in D9.2 Section 3.2.2.

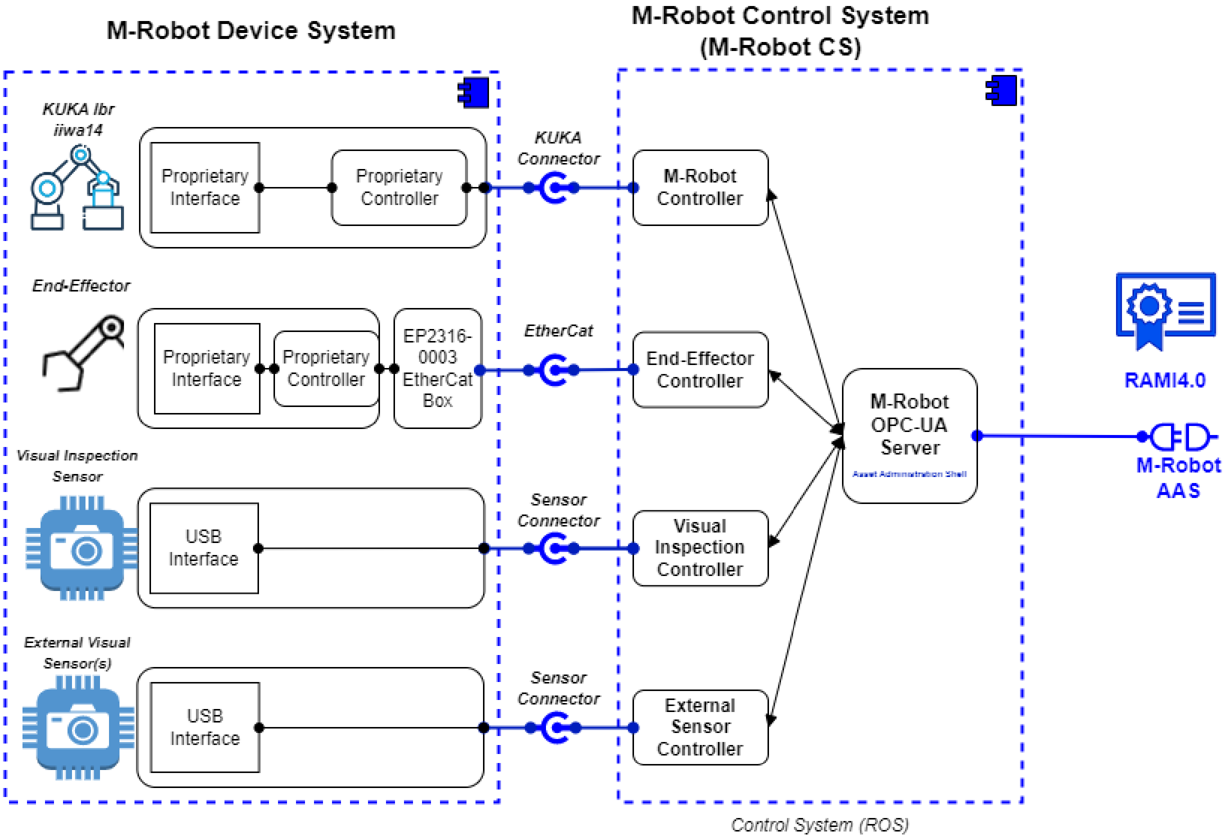

Figure 9: Manipulation Robot (M-Robot) – Insects' handling cell – integration architecture

3.1.3 Intelligent Crates (I-Crates)

Intelligent crates (I-Crates) are conventional crates for insect rearing enhanced with state-of-the-art and novel sensing and data communication technologies. Measured sensor data is crucial for continuous monitoring of the insect rearing process and for enabling the use of digitalized integrated robotics for efficient and smart insect farming process. The development of I-Crates has been implemented at task 7.1.

I-Crates contain embedded environmental sensors and a processing and control node which together form so called Intelligent Integrated Sensors.

Environmental sensor that are implemented in I-Crates are:

- Temperature
- Humidity (RH)
- Soil (substrate) moisture
- Carbon dioxide (CO2)
- Ammonia (NH3)
- pH

The details of the sensors will be presented at D7.1 (Sensing solution to support insect rearing process automation) on M30.

Intelligent Integrated Sensors can obtain and transfer sensor data to the I-Crate dedicated gateway device operating as a cell controller system (I-Crate controller). I-Crates intelligent integrated sensors created data is mainly read through the I-Crate controller. In certain cases, such as to be able to implement cost-efficient pH measuring, a different type of controller can be used. The I-Crate controller consist of a gateway and included (MQTT) client software and it 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 any intelligent integrated sensor (IIS) in the I-Crates
- Sending the data to IMS via its software (MQTT) client
- Receiving and executing instructions from IMS (as an option in certain cases)

The I-Crate integration architecture is show in Figure 10.

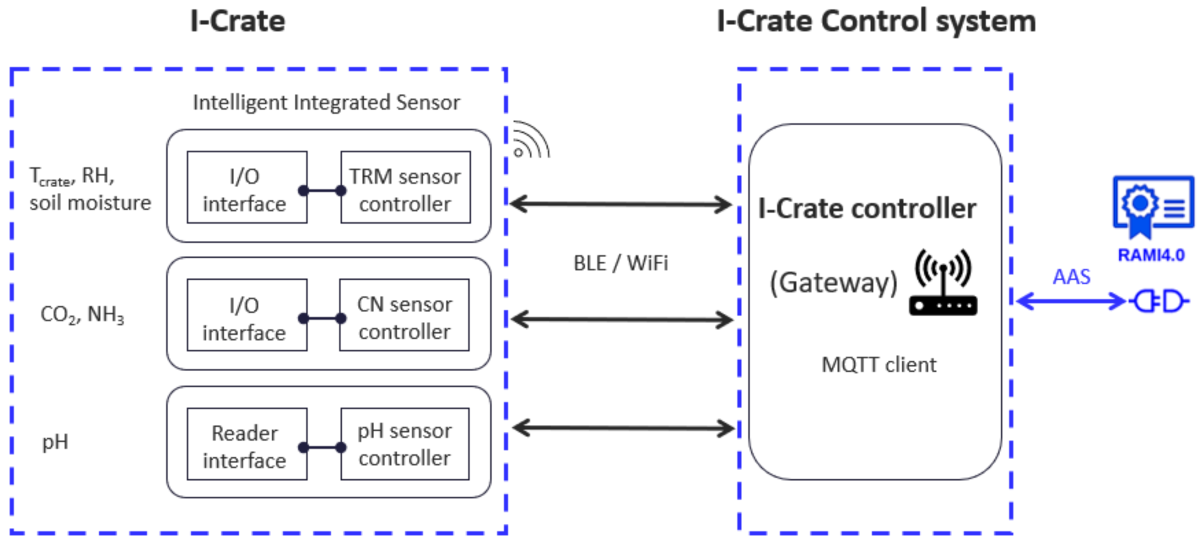

Figure 10: I-Crate integration architecture

MQTT has been selected as the communication protocol to integrate the I-Crate to the CoRoSect's Control System. The I-Crate gateway (and/or dedicated server) implements the I-Crate's AAS, which is defined in D9.1 and guarantees the RAMI4.0 compliance. To make the system as easy to operate as possible, the I-Crate is completely covered by a single AAS managed by the CoRoSect Control System. All sensors' data are aggregated by the controller and the data sent via MQTT according to the i-Crates AAS (RAMI4.0 compliant).

3.1.4 Automated Guided Vehicle (AGV)

The Automated Guided Vehicle (AGV) (Figure 11) is responsible for transporting the crates from and to the different robot cells on a robust and safe way. The vehicle is accommodated with all safety components to comply with the European safety regulations for AGVs like the EN-ISO 3691-4:2020[3].

The AGV receives driving commands from the Route Manager (RM) and Actions from the Shopfloor manager (SFM). A driving command consists out of a number of coordinates in the layout, the vehicle must follow. These coordinates are chosen in such a way that the vehicle is not colliding with any obstacles like mobile or stationary machinery, products, humans and building-related structures. This ensures a minimum stop time of the AGV and increases the feeling of safety for the operators.

As soon as the vehicle arrives at the end position, the Shopfloor manager is sending an action. An action is a programmed task, the vehicle can fulfil. The following actions are defined:

- **Pick**: the AGV picks a (stack of) crates from a predefined location

---

[3] https://www.iso.org/standard/70660.html

- **Drop**: the AGV drops a (stack of) crates on a predefined location
- **Park**: the AGV is on a parking location. If required, the AGV is allowed to connect to an automatic battery charging station

If a pick or drop is foreseen inside a robot cell, the vehicle interfaces with the robot control system (RCS) to ensure a safe entrance and exit. The aim of this interface is to send the robot to a safe position and keep it there until the AGV exists the cell.

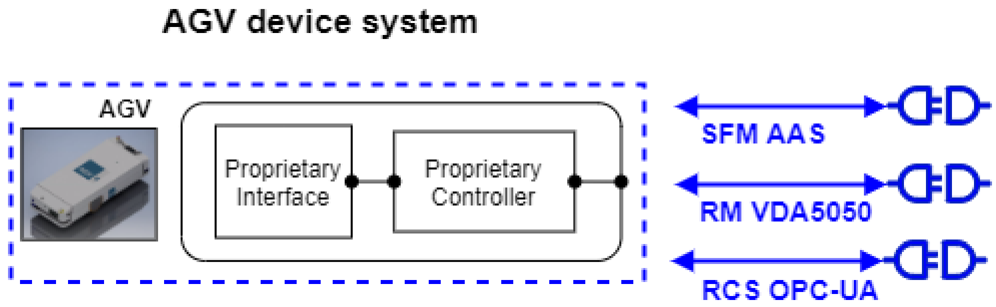

Figure 11: AGV integration architecture

3.2 Manufacturing Execution System (MES) components

The CoRoSect's Manufacturing Execution System (MES) covers the IT (Information Technologies) layer, addressing the data management and technologies' integration, supporting the intelligent management and control of the shop floor layer. It also provides the homogeneous interfaces for the Enterprise Resource Planning (ERP) factory system to collect shop floor information and interact with it. The components developed by the Human-Robot collaboration environment (in WP8) and in the Robot actions' control and planning (in WP6) layer are also served by the MES.

As presented in D2.3 and expanded by WP4 (currently in D4.1 and D4.2 and in M30 in D4.3), the CoRoSect MES (Figure 12) is composed by:

- The **Information Management System** (IMS), which collects, stores and distributes the information from the Shop Floor components, working also as a homogeneous interface to interact with the CoRoSect integrated devices.
- The **Shop Floor Manager** (SFM) controls the overall processes execution, commanding each of the involved Shop Floor components by correlating the information at the IMS
- The **Decision Support System** (DSS) that supports the SFM by analysing the past and current information from the SFM to alter the processes flows for the sake of efficiency.

On top of the MES, CoRoSect will also develop a User Interface to enable the interaction with the SFM and the DSS, as well as implement some specific dashboards to monitor the Shop Floor status.

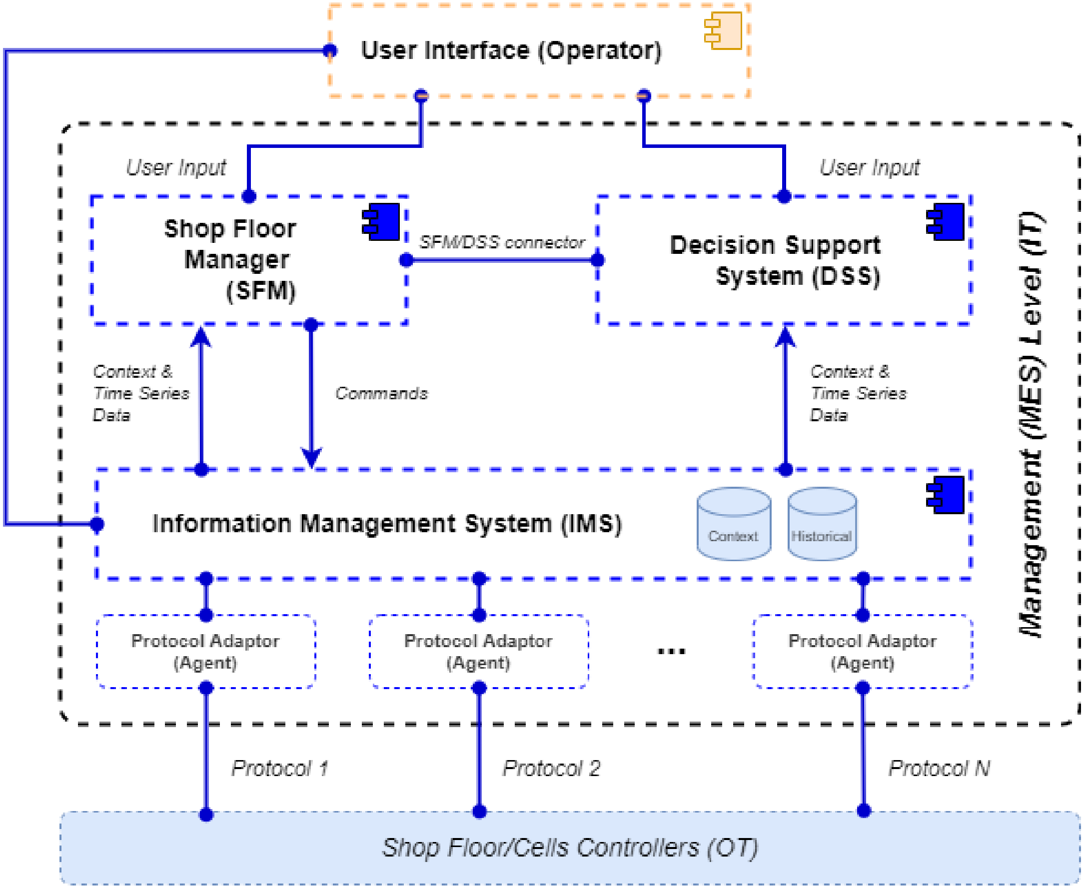

Figure 12: Updated MES logical view

3.2.1 Information Management System (IMS)

The Information Management System (IMS) in the CoRoSect architecture is the MES building block that:

1) Concentrates and distributes all the information collected from the Shop Floor level (and from any other information system involved in the manufacturing process) according to a commonly agreed information model
2) Supports the digitalization (Digital Twin) of each of the Shop Floor components, enabling a homogeneous communication interface to implement the commands flow between the shop floor manager and the shop floor level

All the specifications related to the data management, components digitalization and commands interfaces have been introduced in D4.1 and D4.2 and are to be detailed in D4.3 (M30). This section presents the internal architecture of the developed IMS, represented in the (Figure 13).

At the southbound, the IMS presents an entry point to connect any Shop Floor device (or data source) according to the communication protocol supported by the device. This entry point will be used to gather information from the device (synchronously or asynchronously) and to send requests (commands) to it. The IMS should provide a different entry point per each different supported protocol and, depending in the way these entry points are provided, also a different entry point per each connected device. These are known as **CoRoSect Agents**, and the IMS should have, at least, as many different agents as protocols supported.

The IMS follows the SoA paradigm and so it's northbound is composed by the service REST APIs that enable the interoperability according to RAMI4.0 indications. In this sense, the IMS implements:

- API REST endpoint for context (shop floor current status information) synchronous datasets query/retrieve (**synchronous Q&R service**).
- API REST endpoint to get subscribed to shop floor events, supporting asynchronous data retrieve (**Publish/Subscribe service**).
- API REST endpoint for specific time series historical data retrieve (**historical data access service**)
- AP REST endpoint supporting direct SQL queries to access historical data (**historical data access service**)

The core of the CoRoSect's IMS is the **Broker** building block. This is the component that implements the data gathering and data distribution. It is divided into a) the Context Management module, which receives the data from the shop floor through the protocol agents and implements the synchronous Q&R service; and b) the Publish/Subscribe module, connected to the Context Management module to implement the asynchronous pub/sub service. All the context information, composed by the Digital Twins models and the last information reported by the represented shop floor components are stored in the Context Info database.

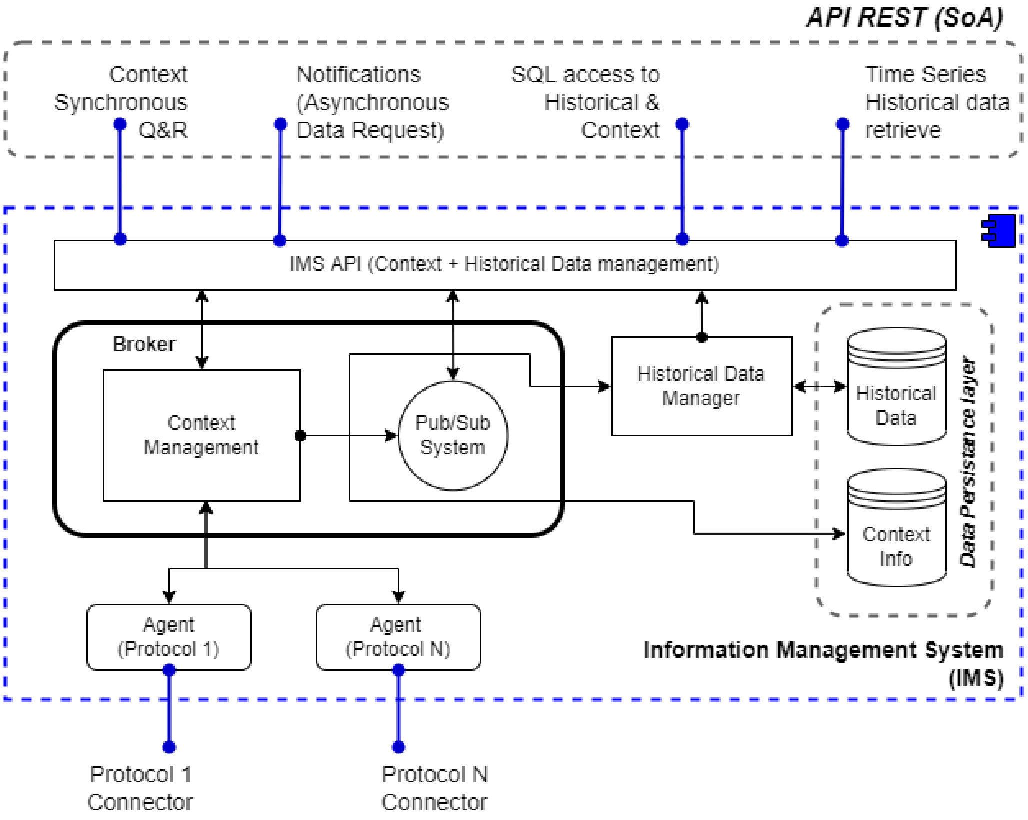

Figure 13: Updated CoRoSect Information Management System's logical view

The Historical Data Manager is, in turn, connected to the broker to collect and store all data sent by the shop floor level. This is organised and stored, according to the CoRoSect information model, in the Historical Data base. This module supports both historical access interfaces, the Time series retrieval, and the SQL queries API.

3.2.2 Shop Floor Manager (SFM)

The SFM (Figure 14) is mainly responsible for the whole production operations management, with the help of the DSS. The IEC/EN 62264[4] is taken partially into account for the necessary service decomposition (see picture below) and overall underlying architecture. A more detailed description can be found also in D4.2.

Functionalities

- **Detailed Scheduling** (DS): A human operator triggers operations requests, and a production schedule will be added.
- **Execution Management** (EM): The EM executes the received job list from the DSS and sends necessary operational commands to the IMS to forward them. Work alerts will be handled appropriately, either to inform the operator or to stop the production line immediately.
- **Data Collection** (DC): Production information or work alert events during the production process can be saved (historical data).
- **Tracking** (T): Ongoing feedback responses to the end users are given based on the current production process.
- **Asset Administration Shell** (AAS): SFM & DSS are represented by appropriate AAS to the outside world.

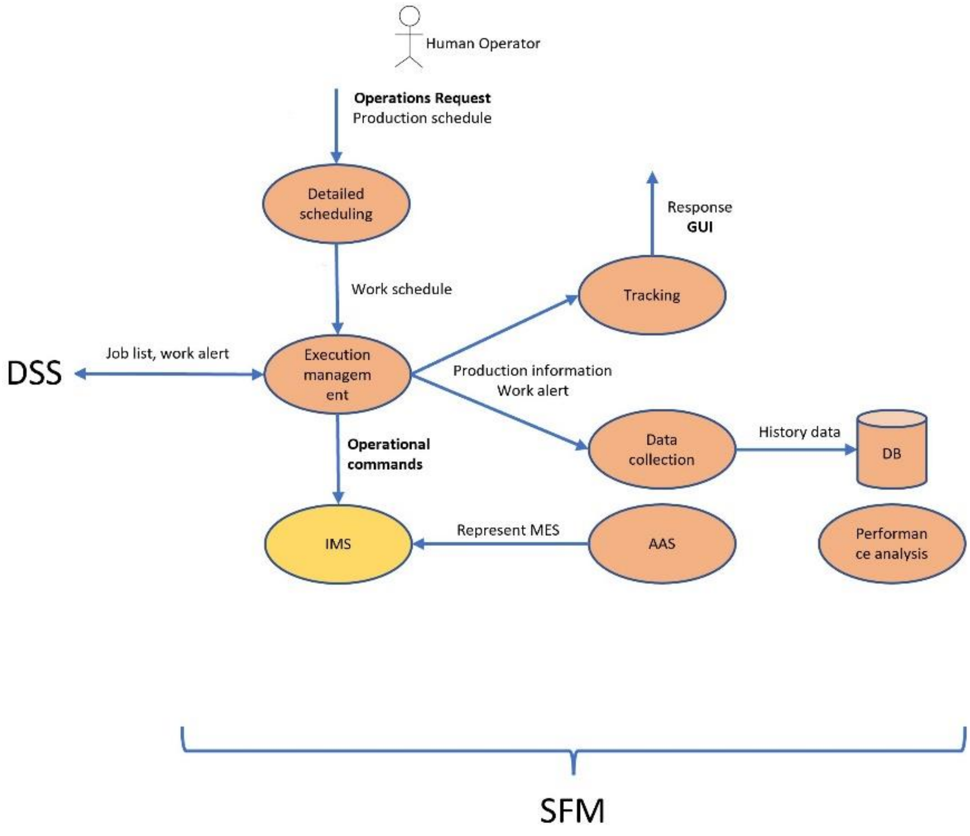

Figure 14: Structure of the Services within the Shop Floor Management (from D4.2)

---

[4] https://www.iso.org/standard/57308.html

3.2.3 Decision Support System (DSS)

The DSS (Figure 15) is supporting the SFM in managing the whole production process. It follows a combined solution of a classical approach, and the IEC/EN 62264 is taken partially into account for the necessary service decomposition (see picture below) and overall underlying architecture. A more detailed description can be found also in D4.2.

Functionalities

- **Model Management** (MM): BPMN configuration files will be loaded. These configuration files include the production definitions and rules. In this case, a work plan is defined with associated tasks to assets, where pre- and postconditions should be met accordingly during the production process.
- **Dispatching** (D): According to the loaded BPMN configuration file, the next tasks to be executed will be determined. After the preconditions are checked, the final job list will be sent for execution to the EM of the SFM. Preconditions are checked based on currently provided data of the related shop floor component AASs retrieved from the IMS. Warehouse information about palettes and crates is considered when the D-Robot is involved.
- **Device Shadows or Digital Twins:** (DT)**:** DTs hold an actual copy of the AAS of the corresponding shop floor component for easier access to the required data, which should avoid continuously asking the IMS.
- **Resource Management** (RM)**:** Includes mainly the provided warehouse for getting actual information about palettes and their stacked crates. This information is necessary especially for D-Robot to stack/destack crates from the palettes correctly.
- **Knowledge Management** (KM): KM includes all the rules which are necessary to make a decision based on the current problem.
- **Decision Maker** (DM): The decision maker is responsible for controlling the ongoing production process. It informs the EM of the SFM when a new job list is available for execution. On the other hand, it can advise the SFM how to react to an occurred problem accordingly.
- **Data Collection** (DC): Incoming production data during the production process can be saved (historical data). Operational responses are forwarded to the dispatching service. AAS data from the shop floor components are retrieved when necessary.
- **Tracking** (T): Optional ongoing feedback responses to the end users are given based on the current production process.

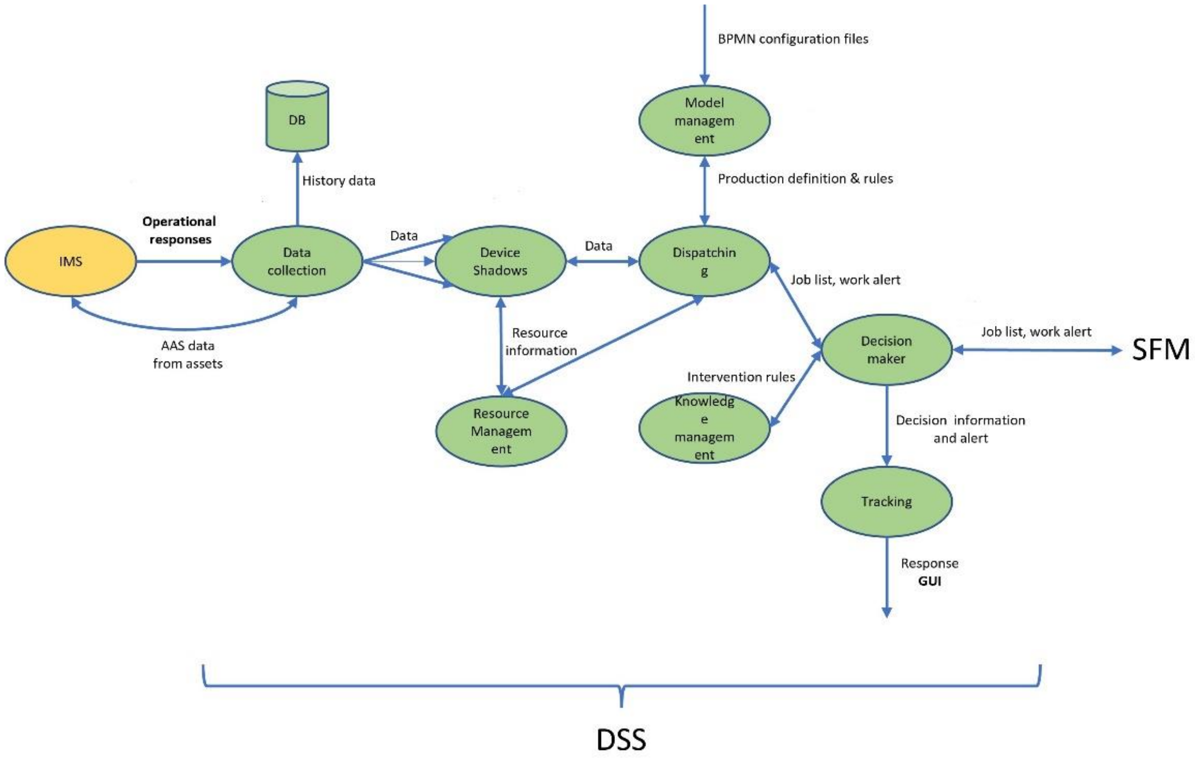

Figure 15: Structure of the Services within the Decision Support System (from D4.2)

3.3 Robotics' Actions planning and control

Within the CoRoSect's architecture, the Robotics' Actions Planning and Control conforms one of innovative layers of the whole system, developing the complementary modules that assist the Shop Floor components on their performance. These novel modules deploy additional data sources (like visualization cameras), extracts shop floor context information from the IMS and combines AI technologies and ML/DL algorithms to adapt the manufacturing process and enhance its efficiency. CoRoSect's WP6 (Robotic action planning and control) develops the software components and WP5 (AI-enabled perception methods) supports the corresponding AI powered embedded algorithms.

3.3.1 Handling cell's system controllers

CoRoSect System develops two cells: the Crates' Handling cell, centred in the D-Robot and attached devices (section 3.1.1) and the Insects' Handling cell, with the M-Robot as its core (section 3.1.2). Unlike the controllers provided at the Shop Floor Level, which concentrates all the functionalities and capabilities of each cell components, exposing the whole combination as a single interface and enable integrations, the handling cell system controllers here developed monitor and orchestrate all their components, to adapt their performance according to efficiency criteria defined in WP6.

Figure 16 depicts the common structure for both handling cells controllers for planning and control. These are composed, in main lines, by:

- **Data gathering** module, which connects with the hardware cell controller to directly collect status and datasets of all the cell's components. It also connects with the CoRoSect's IMS to collect data from other cells or elements at the shop floor (or other relevant sources) that may impact on the cell's performance.
- **ML/DL module** implements the AI algorithms specifically designed and developed for efficiency performance improvement. It is fed by the data gathering module.

- **Loop control** module implements the control mechanisms for the cell. This is assisted by the ML/DL module and the data gathering module.
- The **Actions management** module connects with the cell's hardware controller to interact with the cell's components and control its performance.

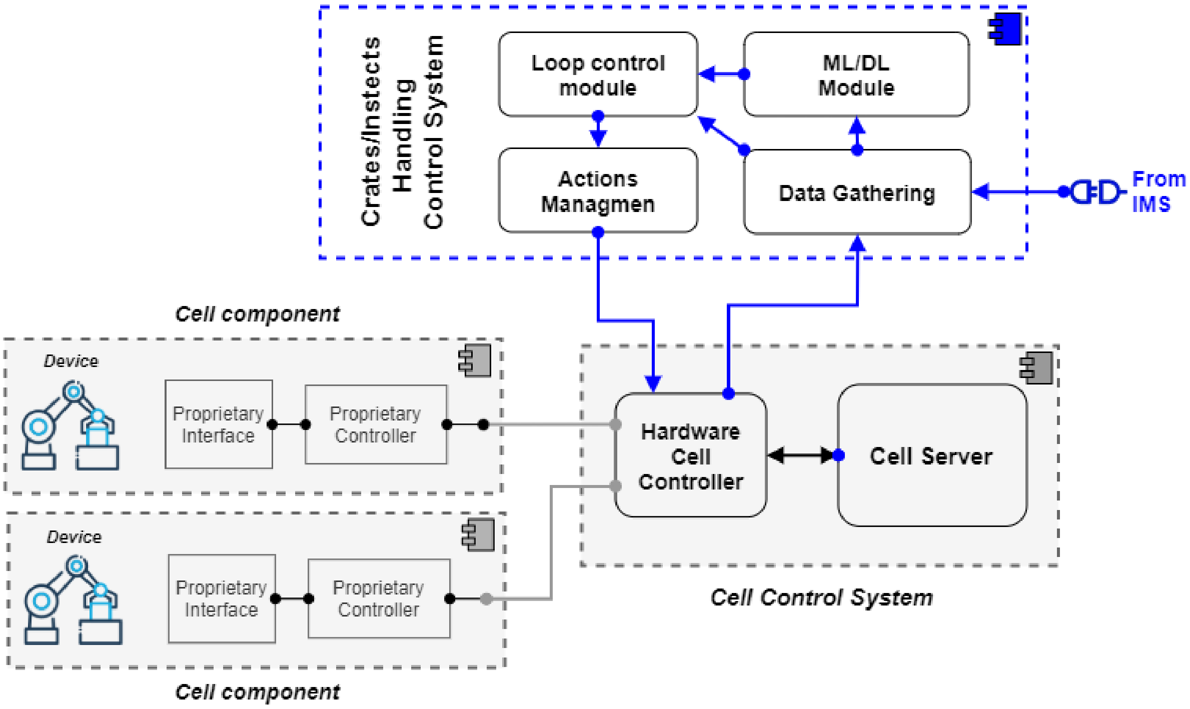

Figure 16: Common handling control system for planning & control (logical view)

3.3.2 Slam module

The AGV localisation is done by a SLAM [10] module (Figure 17). Two IDEC SE2L safety scanners are measuring the physical environment of the AGV. This raw scanner data is used, together with the drive encoders input, to calculate the position of the vehicle inside the layout. The coordinate system will be merged with the coordinates used by the route manager to ensure an accurate localisation. The complete description of the SLAM module can be found in D6.6 Localization, Mapping, and Navigation of AGVs

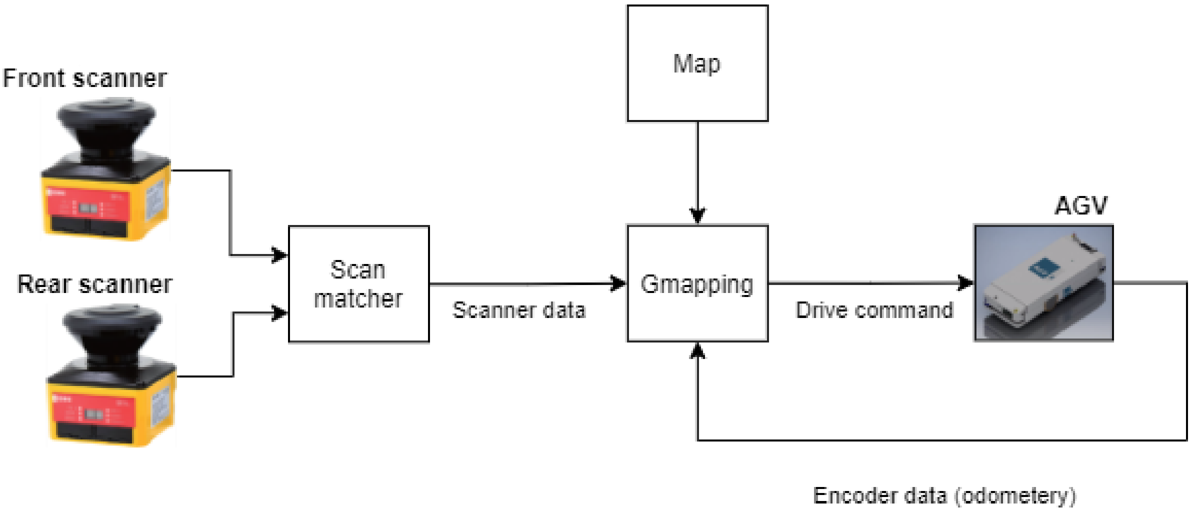

Figure 17: Simple diagram of the SLAM module

3.3.3 Obstacles detector

Obstacles detector component mission is the detection of any possible dynamic or static obstacle placed on the factory floor using monocular (IP) cameras set in fixed points. We detect any obstacle (person or object) present in the areas of pass of the robots and estimate their possible trajectory if they are moving.

Obstacles detector is an independent software module and only communicates (interfaces) with the Route Manager (Figure 18). The obstacles detector sends all the obstacles that it detects and the projection of their trajectory for the next seconds using a JSON message via MQTT [16] interface to the Route Manager. Additionally, it sends heartbeat information periodically.

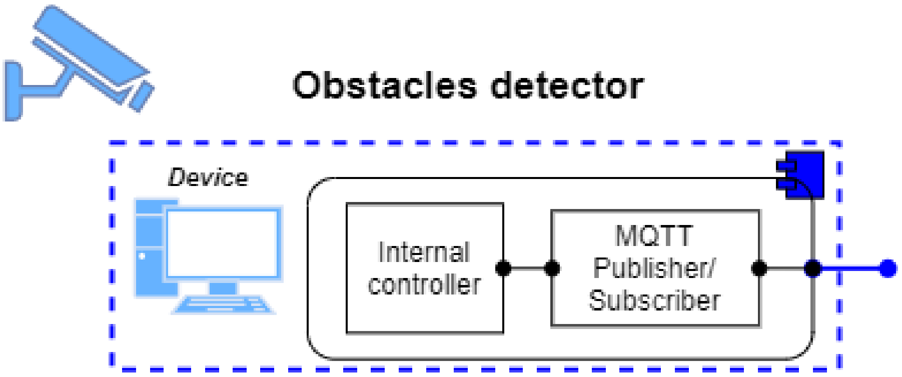

Figure 18: Obstacles detector process view

The system detects the possible obstacles using algorithms of Computer Vision, additionally it estimates their distance to the camera (via depth) and calculates the future trajectories. The controller sub-module interfaces with MQTT interface to communicate with the rest of components in a direct way.

A complete description of the Obstacles Detector can be found in D6.8 Safety concept for robotic systems (creation) due to M24

3.4 Human-Robot Collaboration (HRC) Environment

The Human-Robot collaboration framework includes all the elements operating at both, the shop floor level and on top of the MES, that consume information from the IMS and the OT level devices to implement services that enhance the performance of the manufacturing process while improve the safety in human and robot interactions. CoRoSect develops, at this level, two systems: i) and augmented reality simulator ton improve human and machine specific processes and actions learning; and ii) a Route manager, able to connect with the CoRoSect's objects detector and interact with AGVs to modify trajectories and avoid accidents or blockages.

3.4.1 Augmented Reality simulation – HoloLens System

Microsoft's HoloLens 2 (Figure 19) is a see-through-based augmented reality device. It utilizes augmented reality techniques to visualize 3d objects in enhancing the real world by digitalizing it, by using a variety of sensors like RGBD cameras. In addition, it provides the user with the ability to interact with those objects with different methods such as hand gestures or by using the gaze sensors.

In the context of CoRoSect, Microsoft's HoloLens 2 are used for an AR-based worker-robot communication system to provide the next robot's steps. The aforementioned trajectories of the robot are displayed in an augmented manner to the workers (Display Trajectories Module). Furthermore, notification messages will be displayed to advise workers of the current status of ongoing tasks and to inform them about the status of the HoloLens device (Display Message Module).

In addition, caution messages such as robot errors are also displayed to inform the worker when human intervention is necessary. An extensive explanation and description of the corresponding functionalities can be found on D9.1. The above functionalities are achieved by OPC-UA and the communication between HoloLens 2 and the corresponding server was achieved through ROS. A complete description of the communication protocols and the utilized tools can be found on D9.2 Finally, an AI-based Human Attention Detection will predict the concentration level of farmer workers while performing specific tasks. A complete description of the aforesaid human attention detection will be depicted extensively in D8.3 due to M30

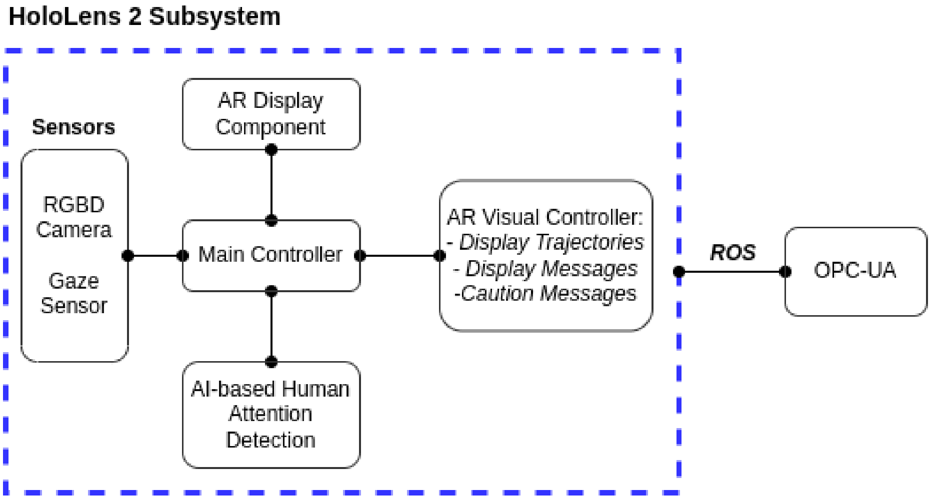

Figure 19: HoloLens integration architecture

3.4.2 Routes' Manager

Route Manager (RM) component deals with the calculation and managing of the routes of the robots in the insect farm floor. Its only mission is to calculate and follow (manage) these routes with two main directives:

- Avoid any possible collision of the robot with any static of moving obstacle (person, other robot or any other object)
- Keep the robot moving as much as possible avoiding it to stop if possible.

RM is an independent software module; it interfaces via JSON messages using Mosquitto[5] (MQTT) with the following components (see Figure 20):

- Obstacles detector: Obstacles detector sends all the obstacles that it detects using MQTT interface. This information is used to recalculate routes that collide with these obstacles avoiding them
- AGV: Route manager sends the "next" points for the route of the AGV. From its side, AGV sends periodically its position. This position information is used to track the position of the robots. All information is sent using MQTT interface
- SFM: Shoop Floor Manager orders the creation of a new route for a named AGV. Optionally it can cancel any running route. All information is sent using MQTT interface

Additionally, it sends heartbeat information periodically.

---

[5] https://mosquitto.org/

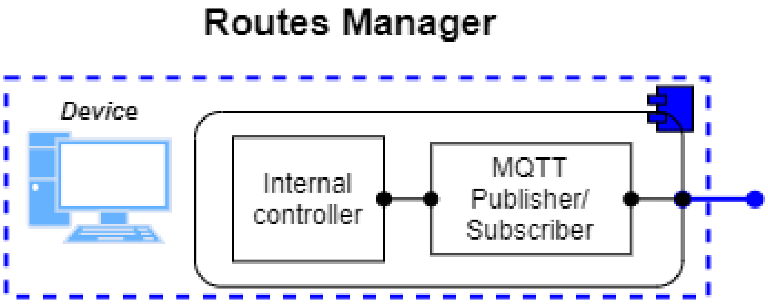

Figure 20: Routes Manager process view

The system processes internally the routes and its tracking (management). The controller sub-module interfaces with MQTT interface to communicate with the rest of components in a direct way.

The CoRoSect´s Route Manager is developed within WP6 and is fully described in D6.8 Safety concept for robotic systems (creation) due to M24.

3.5 Security layer

A common Industry 4.0 scenario (like CoRoSect) is driven by interconnections between robots and information systems, mechatronics, and novel infrastructures from the Internet of Things (IoT) world. These elements can generate, share, and consume big amounts of datasets which manage the manufacturing processes. Security (and cybersecurity) on these data movements that guarantees the Confidentiality, Integrity, and Availability (the CIA[6] triad, pillars for information security) is mandatory to ensure a reliable operation at all levels of the CoRoSect's architecture.

As introduced in D2.3, the CoRoSect's security layer here proposed is focused on the information managed all along its different architecture's layers and the CIA triad objectives. Specifically, the CoRoSect's Security layer is designed to introduce the information security management in the pilots to the carried out, remarking mainly the data access control and the cryptography mechanisms to protect the information, but allowing the expansion of the security measurements to reinforce different aspects in a wider and functional extended scenario. In this sense, within this architecture we will implement:

- **Access Control** policies, to protect both, the access to the information and the data uploaded into the MES. These access control mechanisms will guarantee that only registered and allowed users can read, modify, deleted, update or alter the compiled information. Here "user" means either, a human operator accessing, reading and managing system's information, or a digital component (device, data source, controller or server) that updates, uploads or modifies any record in the system. This is done according to three steps:
    - **Identification**: all the CoRoSect's users (either human or digital operators or data sources) must be registered, so any data request (including data query, update, upload, modification, etc.) must be assigned to a user id, which travels attached to the request.
    - **Authentication**: the user id of the request is verified (authenticated) by the security layer to ensure the user id is not impersonated. This will be done, within CoRoSect, by using a password and/or a token.
    - **Authorization**: once the user is authenticated, the system checks if this user has the proper rights to execute the request (mainly if it can read or modify the requested resource).

---

[6] https://panmore.com/the-cia-triad-confidentiality-integrity-availability

- **Cryptography** mechanisms to protect the data while it is in transit. Data sent from a source to the MES and vice-versa are encrypted. This protects the data links against unauthorized or accidental disclosures.

Figure 21 illustrates the Security Layer architecture for CoRoSect system. On its simplest description, it is composed by 4 main components to implement the CIA triad according to the project's targets:

- **PEP** (Policy Enforcement Point) proxy: are agents configured per each API endpoint of the architecture that acts as a barrier, **intercepting** each request addressed to its endpoint to apply the configured access policies, relying on the PDP. We have one PEP proxy instance per each endpoint to be protected.
- **PDP** (Policy Decision Point): is where the request intercepted by the PEP is **evaluated** according to a set of authorisation policies defined in close collaboration with the IdM.
- The **IdM** (Identity Manager): **registers** and **manages** all the system's **users** set, and the different **roles** created to implement the CoRoSect's access control. A security role is composed by a set of permissions (access to specific applications, resources, and operations) usually defined according to a management process role. On this first architecture proposal, CoRoSect will use a RBAC (Role-Based Access Control) mechanism to define its authorisation policies: each user will have a specific role assigned and will be able to operate within CoRoSect's architecture according to its role.
- Finally, each of the exposed endpoints (both, external and internal to the architecture) will use **encrypted channels**.

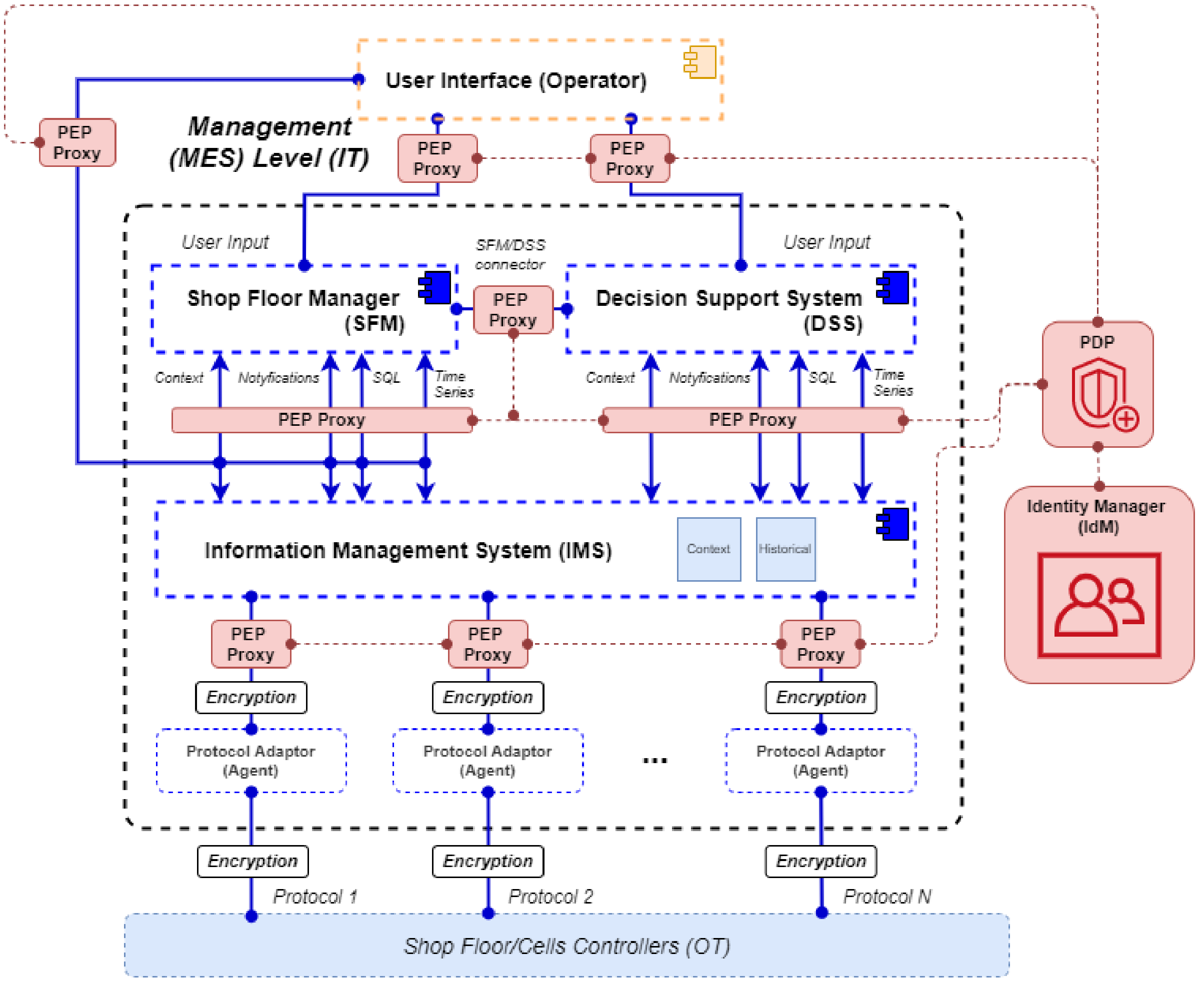

Figure 21: CoRoSect's security layer proposal

3.6 Scalability and Replicability

The CoRoSect's Advanced Architecture has been designed and defined on top of the requirements and the scenarios collected and proposed in WP2 to support the CoRoSect's pilots and test the project's objectives. Also, this proposed architecture is envisioned to cover a wider smart manufacturing scenario which means more integrated shop floor components, growing data volumes and new functionalities and exploitation services consuming resources. This is understood as scalability. On the other hand, the architecture should provide the system with the ability to be easily duplicated and deployed in or adapted to another similar manufacturing scenario. This is replicability. In main lines, CoRoSect System covers:

- **Scaling-up in size**: the implementation of the system supports a larger area than the one used in pilots, covering the integration of new compliant devices and information systems (larger Shop Floor - OT layer), a wider network infrastructure, and a larger number of system's users (larger IT layer).
- **Scaling-up in density**: the system is designed to support a larger number of shared datasets and data consumers. This means growing storage, data requests, functionalities etc.
- **Insects' farms replication**: the system is to be deployed in five different insect farms' scenarios, with different layouts and insects' rearing processes.
- **International replication**: the system is to be deployed in five different European countries, supporting all legal European constraints in terms of insects' rearing.

The way of evaluating the CoRoSect's system scalability and replicability can be done according to these dimensions [17]

- **Regulations**' dimension addresses the different regulatory frameworks affecting the actual system deployment. This would affect mainly the international replication, but also the insects' farm replication and the scaling-up in size. CoRoSect pilots' have been already analysed for 5 different EU countries regulation frameworks in terms of insect's rearing.
- **Economic** dimension analyses the costs of scaling (or replicating) the system and if the cost-benefit ratios or internal rates of return makes this viable or attractive to the market. This aspect may constitute a major barrier or driver, so it is worth to be considered.
- **Stakeholders' acceptance** reflects if the current set of involved end users, regulators, and tech providers are ready to embrace a scaled-up version of the system. CoRoSect involves stakeholders from the very beginning on its system's designing process.
- **System's Use Cases** deep analysis must be done to identify barriers, constraints and drivers that affect the scalability and replicability. The CoRoSect's use cases have been analysed in parallel with the architecture definition, and this architecture has been designed to cover the uses' cases requirements.
- **Standardization** level, reflected in the data-models and communication protocols used to build the system. The more standards used at this level, the better replicability and scalability factors will be achieved. CoRoSect relies on RAMI4.0 and on a Service oriented Architecture (SoA) what provides a high standardised framework.
- **Technical components** required to replicate the system may also impact on its' scalability. All technical CoRoSect providers are involved in pilots, so technical layer is guaranteed for the system's replicability.

According to the standardization level of the proposed architecture and the involvement of the stakeholders (end users, farms, etc.) and the technical components providers, it can be derived that the CoRoSect Architecture, and so, the CoRoSect System proposed has been designed to achieve the scalability and replicability parameters that makes this a reliable solution for insect's rearing which can be also extended to other related manufacturing processes.

4 Conclusions

Industry 4.0 converges cutting-edge industrial mechatronics and cyber-systems with novel IoT infrastructures to create integrated smart manufacturing environments with new services that enhance productivity, efficiency, safety and reduce carbon footprint. This is also known as the Industrial Internet of Things (IIoT). It's Reference Architecture Model (RAMI4.0) structures in a common set of layers the I4.0, breaking down complex processes into easy-to-grasp packages with well-identified functionalities, to make easier the integration and cooperation of heterogenous sub-systems.

With the full compliance with this RAMI4.0 in mind (Figure 22) and during the first two years of the project, CoRoSect has depicted its own Advanced System Architecture, building on top of the Initial System Architecture proposed in D2.3 (M12). We've revisited the proposed integration approaches according to the development of the software controllers (WP5, WP6, WP7, WP8) and the system interfaces (WP4 and WP9) and assisted these to guarantee an I4.0 interoperable environment that support the project's pilots deployment and performace (WP10) while addressing the functional and technical requirments from WP2.

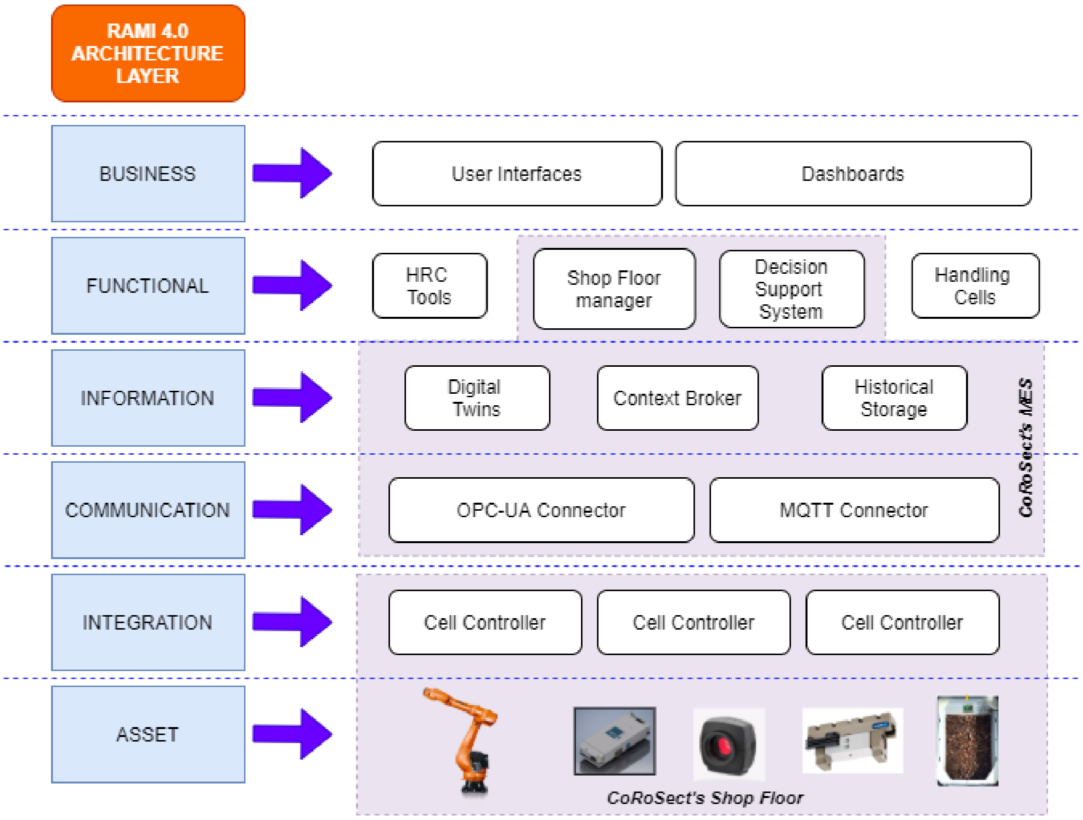

Figure 22: RAMI4.0 layer's correspondence with CoRoSect's Advanced System Architecture

This CoRoSect's System Advanced Architecture

- Defines the integration schemas for all the CoRoSect components, to be I4.0 compliance, and provides a common template to integrate new systems (Robots, Mechatronics, IoT infrastructures, data sources or data sinks) and so expand the whole system capabilities.
- Composes a versatile architecture, capable of supporting CoRoSect's scenarios variations and being adapted to other similar manufacturing environments.

- Has been used to build and instantiate the first release of the CoRoSect's system which, in turn, validates the architecture itself on its first stage (supporting the integration of all components)

Next steps, within WP9 and WP10, will validate the performance of the CoRoSect's System build and deployed according to this Advanced Architecture. This will also validate the full architecture from the point of view of processes executions and addressed functionalities, confirming this as an option for Smart Manufacturing environments. During these validation steps, some corrections or modifications may appear, that will be reflected in the corresponding WP9 deliverables.

.

5 References

[1] CoRoSect, “D2.3- Initial System architecture,” European Union’s Horizon 2020. G.A. No 101016953, 2021.

[2] CoRoSect, “D2.4- Advanced System Architecture,” European Union’s Horizon 2020. G.A. No 101016953, DEC-2022.

[3] CoRoSect, “D4.2- Data analytics to obtain the prediction models,” European Union’s Horizon 2020. G.A. No 101016953, JUN-2022.

[4] CoRoSect, “D6.1- Documentation of control for handling of crates,” European Union’s Horizon 2020. G.A. No 101016953, 2021/2022.

[5] CoRoSect, “D6.2- Documentation of control for insect handling,” European Union’s Horizon 2020. G.A. No 101016953, 2021/2022.

[6] CoRoSect, “D6.4- Safety concept and control in robotic systems,” European Union’s Horizon 2020. G.A. No 101016953, DEC 2021/DEC 2022.

[7] CoRoSect, “D7.2- Report on and documentation of robot cell for handling crates,” European Union’s Horizon 2020. G.A. No 101016953, DEC 2021/DEC 2022.

[8] CoRoSect, “D9.1- Integration plan and definition of the interfaces,” European Union’s Horizon 2020. G.A. No 101016953.

[9] CoRoSect, “D9.2- Integrated CoRoSect Platform - Release I,” European Union’s Horizon 2020. G.A. No 101016953, DEC 2022.

[10] Mathworks.com, “SLAM (Simultaneous Localization and Mapping),” 2022. [Online]. Available: https://uk.mathworks.com/discovery/slam.html. [Accessed 21 12 2022].

[11] CoRoSect, “D10.2- Pilot preparation and planning,” European Union’s Horizon 2020. G.A. No 101016953, 2022/2023.

[12] “ISO/IEC 7498-1:1994 Information technology — Open Systems Interconnection — Basic Reference Model: The Basic Model,” June 1999. [Online]. Available: https://www.iso.org/standard/20269.html. [Accessed 14 December 2022].

[13] Open Robotics, “ROS - Robot Operating System,” Open Robotics, 2021. [Online]. Available: https://www.ros.org/. [Accessed 20 12 2022].

[14] PickNik Robotics, “MoveIt - Moving robots into the future,” 2022. [Online]. Available: https://moveit.ros.org/. [Accessed 20 12 2022].

[15] OPC Foundation, “OPC Unified Architecture (UA),” OPC Foundation, 2022. [Online]. Available: https://opcfoundation.org/about/opc-technologies/opc-ua/. [Accessed 20 12 2022].

[16] "MQTT: The Standard for IoT Messaging," OASIS, [Online]. Available: https://mqtt.org/.

[17] R. C. P. F. Andrea Rodriguez-Calvo, "Scalability and replicability analysis of large-scale smart grid implementations: Approaches and proposals in Europe,," *Renewable & sustainable energy reviews,* vol. 93, pp. 1-15, 2018.

CoRoSect

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