ZEROSWARM

D5.2 - IEC-61499 Platform Extensions for Dynamic Setup and Reconfiguration of Agents

ZERO-enabling Smart networked control framework for Agile cyber physical production systems of systems

Project funded by Horizon Europe, Grant Agreement #101057083

Topic	HORIZON-CL4-2021-TWIN-TRANSITION-01-08
Project Title	ZERO-enabling Smart networked control framework for Agile cyber physical production systems of systems
Project Number	101057083
Project Acronym	Zero-SWARM
Contractual Delivery Date	M14
Actual Delivery Date	M15
Contributing WP	WP5
Project Start Date	01/06/2022
Project Duration	30 Months
Dissemination Level	Public
Editor	INNO
Contributors	all

Author List

Leading Author (Editor)			
Surname	Initials	Beneficiary Name	Contact email
Maccioni	RM	RWG	Raffaele.Maccioni@RWings.tech
Co-authors (in alphabetic order)			
Surname	Initials	Beneficiary Name	Contact email
Serra	DS	RWG	Domenico.Serra@Modelite.tech
Contributors (in alphabetic order)			
Surname	Initials	Beneficiary Name	Contact email
Fritz	AF	NX-SE	artur.fritz@se.com
Deshmukh	SF	NE-SE	shreya.deshmukh@se.com
Serra	SD	RWG	Domenico.Serra@modelite.tech
Lepore	ML	RWG	Mario.Lepore@modelite.tech
Liakh	TL	LTU	tatiana.liakh@ltu.se

Reviewers List

List of reviewers (in alphabetic order)			
Surname	Initials	Beneficiary Name	Contact email
Pagliarini	GP	RPK	GianPietro.Pagliarini@promachbuilt.com
Mendez	RM	AIM	roi.mendez@aimen.es
Khodashenas	PS	HWE	Pouria.khodashenas@huawei.com
Wangwei	WW	HWE	wangwei505@huawei.com
Mpatziakas	AM	CERTH	ampatziakas@iti.gr

Document History

Document History			
Version	Date	Author	Remarks
00.00	15/02/2023	R Maccioni	Table of Content
00.00- 00.14	01/03/2023 12/09/2023	R Maccioni	working revision with contributions by Authors and Coauthors
00.15	16/09/2023	G.P. Pagliarini	Revision by reviewers
00.16 / 00.17	27/09/2023	R Maccioni	Revision after comments from reviewers
00.18	04/10/2023	R Maccioni	Comments Incorporated

DISCLAIMER OF WARRANTIES

This document has been prepared by Zero-SWARM project partners as an account of work carried out within the framework of the contract no 101057083.

Neither Project Coordinator, nor any signatory party of Zero-SWARM Project Consortium Agreement, nor any person acting on behalf of any of them:

- makes any warranty or representation whatsoever, express or implied,
    - with respect to the use of any information, apparatus, method, process, or similar item disclosed in this document, including merchantability and fitness for a particular purpose, or
    - that such use does not infringe on or interfere with privately owned rights, including any party's intellectual property, or
- that this document is suitable to any particular user's circumstance; or
- assumes responsibility for any damages or other liability whatsoever (including any consequential damages, even if Project Coordinator or any representative of a signatory party of the Zero-SWARM Project Consortium Agreement, has been advised of the possibility of such damages) resulting from your selection or use of this document or any information, apparatus, method, process, or similar item disclosed in this document.

Zero-SWARM has received funding from the European Union’s Horizon Europe research and innovation programme under grant agreement No 101057083. The content of this deliverable does not reflect the official opinion of the European Union. Responsibility for the information and views expressed in the deliverable lies entirely with the author(s).

Executive Summary

Considering the IEC 61499 framework, this deliverable outlines the critical concepts and considerations in developing a dynamic setup and reconfiguration system for agents, including software agents, edge devices, robots, etc.

The final goal is to address a novel, robust way to deliver auto-recognition, setup, and dynamic reconfiguration of agents in a cloud-to-edge architecture, including distributed intelligent agents (software & hardware). We provide practical evidence considering the Zero-SWARM use cases.

Moreover, the context of this deliverable involves leveraging new cloud-to-edge technologies and 5G connectivity to enable secure auto-recognition of agents within a dynamic environment. Considering the goal to extend the IEC 61499, the key points discussed are as follows:

Dynamic Device Recognition: to dynamically identify and recognize devices as they are added or removed from the setup. Achieving this requires a flexible device discovery mechanism that continuously scans the environment, detects changes, and updates the system's knowledge of available agents.

Dynamic Environment Recognition: To ensure coherent reconfiguration of logic (software agents) and devices (hybrid software-physical agents), the system should dynamically recognize and identify the context or status of the operating environment. For instance, changes in the layout of a workspace can impact the operations of robots or other agents. This recognition is essential for making informed near and real-time decisions about reconfigurations.

Real-time Reconfiguration: Reconfiguration actions need to occur in real-time to adapt to dynamic changes effectively. The system's reconfiguration engine should be designed for quick adjustments to control logic, function block connections, and resource allocation based on the updated configuration.

Dynamic Resource Allocation: As the setup evolves, the system should dynamically allocate resources such as computing power, communication bandwidth, or physical space to support the new configuration. This may involve reallocating resources from existing devices or provisioning additional resources to accommodate changes efficiently.

Event-driven Adaptation: Events and notifications are pivotal in dynamic setups. Devices should continuously communicate their status, availability, and changes to the system through event-driven mechanisms (see Deliverables 4.2 and 4.4 for the architecture).

Highlighting the innovative elements of the research project in the context of dynamic setups and reconfiguration of agents, especially when compared to the state-of-the-art:

1. Autonomous Device Recognition: The system autonomously identifies devices as they join or leave the network, a departure from manual configurations.
2. Real-time Reconfiguration: A real-time engine quickly adjusts control logic, resource allocation, and connections in response to dynamic changes.
3. Dynamic Resource Allocation: Resources are dynamically allocated, breaking away from static or rigid allocation methods.
4. Event-driven Adaptation: Real-time responsiveness through event-driven mechanisms enhances adaptability, crucial in dynamic environments.
5. 5G as Enabler: Leveraging 5G for high-speed, low-latency communication, revolutionizing real-time data exchange and decision-making. In addition, Cellular communication, in this case 5G

includes functionalities such as the Unified data management (UDM) which manages network user data in a single, centralized element. It can be paired with the user data repository (UDR) which stores the user data such as customer profile information, customer authentication information, and encryption keys for the information. UDM resides on the control plane of cellular communication. Next generation UDM in cellular communication should be able to support dynamic setup and reconfiguration platforms with Digital ID - Describing Digital Person/Robotics/Human. It involves developing a comprehensive digital identity (Digital ID) system that accurately represents and distinguishes between digital entities such as digital persons, robots, and humans.

6. Semantic Understanding: Incorporating semantic data for intelligent, context-aware reconfiguration decisions.
7. Autonomous Adaptation in Complexity: The system excels in complex, dynamic environments, going beyond traditional control systems' capabilities.
8. Security in Agent Identification: Ensuring secure agent recognition and identification in a growing network of connected devices.

ZEROSWARM

Table of Contents

List of Figures

List of Acronyms

Acronym	Description
5G	Fifth-Generation Wireless Communications
AE	Alarm And Events
AFoF	AALTO Factory of the Future
AMR	Autonomous Guided Vehicles
AI	Artificial Intelligence
AIC	Automotive Intelligence Centre
AIIC	AALTO Industrial Internet Campus
AR	Augmented Reality
CAT	Composite Automation Type
CPSoS	Cyber-Physical System of Systems
CUC	Centralized User Configuration
DA	Data Access
DFA	Demonstration Factory Aachen (DFA)
DLFi	Distributed Learning Framework
DSS	Dynamic Spectrum Allocation
E2E	End-To-End
eMBB	Enhanced Mobile Broadband
gPTP	Generalized Precision Time Protocol
HDA	Historical Data Access
ICT	Information Communication Technologies
IDS	International Data Spaces
IICF	Industrial Internet Connectivity Framework
IIoT	Industrial Internet Of Things
IIRA	Industrial Internet Reference Architecture

IMC	Intelligent Mechatronic Components
ITU	International Telecommunication Union
LBO	And Local Breakout
MAS	Multi-Agent Systems
MES	Manufacturing Execution Systems
mIoT	Massive Internet of Things
mMTC	Massive Machine Type Communication
MR	Mixed Reality
NASA	National Aeronautics and Space Administration NASA
NASA	Telematic Data Collector
NPN	Non-Public Networks
OLE	Object Linking and Embedding
OPC	Open Platform Communication
OPC-UA	OPC Unified Architecture
PLC	Programmable Logic Controllers
PTZ	Production Technology Center (PTZ)
ROS	Robot Operating System
SLAM	Simultaneous Localization and Mapping
SNPNs	Stand-Alone Npns
SOAP	Simple Object Access Protocol
SoS	Systems Of Systems
TSN	Time Sensitive Networking
UA	Unified Architecture
UACP	Unified Architecture Connection Protocol
UES	User Equipment
UPF	User Plane Function
uRLLC	Ultra-Reliable Low Latency Communication
VR	Virtual Reality
AMR	Automatic Mobile Robots
AGV	Automatic Guided Vehicles

1 Introduction

1.1 Purpose of the document

The purpose of the document is to provide a map of various dynamic set-up and reconfiguration needs that could be managed in different applications by the Zero-SWARM framework and define how they could be managed effectively by extending the IEC-61499.

The multiple aspects involved in dynamic set-up and reconfiguration of agents can be summarized in:

Dynamic Device Recognition: The system should be capable of dynamically recognizing and identifying devices as they are added or removed from the setup. This requires a flexible device discovery mechanism that can continuously scan the environment and detect changes.

Dynamic Environment Recognition: The system should be capable of dynamically recognizing and identifying the status of the context, where robots and humans operate to coherently reconfigure logics and devices. For example, in the case of a fleet of AMR, changes on the layout impact their operations.

Real-time Reconfiguration: Reconfiguration actions need to be performed in real-time to accommodate the dynamic changes. The reconfiguration engine should be designed to quickly adapt the control logic, update connections between function blocks, and allocate resources based on the new configuration.

Dynamic Resource Allocation: As the setup changes, the system should dynamically allocate resources such as computing power, communication bandwidth, or physical space to support the new configuration. This may involve reallocating resources from existing devices or provisioning additional resources to handle the changes.

Event-driven Adaptation: Events and notifications play a crucial role in dynamic setups. Devices should continuously communicate their status, availability, and changes to the system through event-driven mechanisms. The system should be designed to receive, process, and respond to these events in real-time, triggering the necessary reconfiguration actions.

By extending IEC 61499 with these capabilities, the system can automatically recognize and accommodate new devices, such as an added AMR, by dynamically reconfiguring the control logic and resources. This allows for efficient and flexible management of the manufacturing system, enabling seamless integration of new devices and improved adaptability to changing production requirements.

**Note: The submission of this deliverable was delayed by < 1 month, because of additional synchronisation needs with WP4 and other WP5 tasks.*

2 Structure of the document

The document is structured as follows:

- **Chapter 1** is an introduction to the whole document, describes its scope and purpose, as well as its structure;
- **Chapter 2** From Use-Case Requirements Analysis to the Functions Design

- MAP of dynamic setups and reconfiguration real contexts and practical needs Definition of the dynamic setup and reconfiguration requirements based on use cases and additional possible real and frequent applications.
- Classification of the general aspects versus more specific / use case related aspects.
- Identification of situations where 5G is a significant enabler vs situation where 5G is a nice to have opportunity

- **Chapter 3** Reconfiguration functionality
  - Self-declaration functionality and Services needed for AMR and Semantics
  - HMI and IEC61499 Application
  - Reconfiguration with Digital Twin
  - ROS2-IEC61499 Interface
  - Reconfigurability impact on AASn
- **Chapter 4** Reconfiguration functionality
  - Object-Design of IEC 61499 application

3 From **Use-Cases Requirements & real-world Analysis** to **Functions Design**

3.1 MAP of dynamic setups and reconfiguration from real contexts

There are several possible auto-reconfiguration functionalities that automated handling devices, production machines, Automated Mobile Robots, or even drones may have. For examples:

- **Self-diagnosis and repair**: These devices can detect faults or errors in their systems and automatically reconfigure themselves to resolve the issue. This can include resetting the device or replacing faulty components.
- **Adaptive operation**: These devices can adjust their operations based on changing conditions or production requirements. For example, a machine may adjust its speed or tooling based on the materials being processed.
- **Automatic tool changing**: Production machines can automatically change out tooling or end effectors based on the requirements of the production process. This can reduce downtime and increase efficiency.
- **Autonomous navigation**: Automated Mobile Robots and drones can autonomously navigate to different locations and adjust their paths based on obstacles or changes in the environment.
- **Load sensing and adjustment**: Handling devices can sense the weight or size of the items they are handling and adjust their grip or operation accordingly. This can prevent damage to products or equipment.
- **Predictive maintenance**: These devices can use sensors and data analytics to predict when maintenance will be needed and automatically reconfigure the resources (e.g., schedule maintenance activities).
- **Multi-tasking**: These devices can perform multiple tasks or functions without requiring manual intervention. For example, a production machine may be able to automatically switch between different production runs without requiring operator input.

- **Energy optimization**: These devices can automatically adjust their energy usage to optimize efficiency and reduce energy costs. This can include turning off or reducing power to idle equipment, optimizing motor control, and managing heating and cooling systems.
- **Intelligent material handling**: These devices can use sensors and algorithms to optimize the movement and handling of materials. For example, a robot's arm may be able to identify the best grip or orientation for a particular item based on its shape and weight.
- **Configuration and Behaviour update by Machine Learning**: These devices can use machine learning algorithms to learn from past operations and adjust their configuration or behaviour accordingly. This can include predicting maintenance needs, optimizing production processes, and improving efficiency over time.

The specific functionalities will depend on the device and its application.

3.2 IEC 61499 Extensions for Dynamic Setup & Reconfiguration

IEC 61499, as a standard for distributed control systems, specifically focusing on the industrial automation domain, defines a model for the design and implementation of function blocks, which can represent different elements of control logic.

The standard allows for modularity, reusability, and distribution of control logic in industrial applications. While IEC 61499 does not explicitly include auto-discovery and reconfiguration mechanisms, these functionalities can be extended. Following the key elements to achieve such extension.

Auto-discovery:

Auto-discovery allows devices or components within a distributed control system to automatically detect each other and establish communication without requiring manual configuration.

This feature can be beneficial when adding new devices or when existing devices need to connect or reconnect without manual intervention.

To implement auto-discovery within an IEC 61499 system, you can use one of the following methods:

a. Zero-configuration networking (Zeroconf): Zeroconf protocols like mDNS (Multicast DNS) and DNS-SD (DNS Service Discovery) enable devices to announce their presence and services on the network. Implementing Zeroconf protocols in the communication infrastructure of IEC 61499 devices allows them to automatically discover each other and their capabilities.

b. Service-oriented middleware: it is possible to introduce a service-oriented middleware layer in the IEC 61499 system. This middleware would handle device registration, service discovery, and message routing, making it easier for devices to find and communicate with each other.

Reconfiguration:

Reconfiguration in IEC 61499 refers to the ability to modify the behaviour of a distributed control system at runtime. This can involve changing the configuration of function blocks and the connections between them or even adding or removing function blocks on the fly to adapt to changing requirements or fault tolerance.

Reconfiguration includes the capability to update configuration systems data and parameters, such as reconfiguring the AMR fleet.

To enable reconfiguration in an IEC 61499 system, we should consider the following elements:

a) Dynamic binding and instantiation: Implement mechanisms that allow function blocks to be dynamically instantiated and bound to each other during runtime. This enables the system to adapt to changes in the environment or requirements.
b) Remote parameterization: Allow function blocks to receive configuration or parameterization updates from a centralized source or a configuration manager. This way, you can modify the behaviour of function blocks without stopping the entire system.
c) Redundancy and fault tolerance: By introducing redundant function blocks or devices, the system can switch to alternate components in case of failure, ensuring continuous operation and reconfiguration in response to faults.
d) Runtime state monitoring: Use monitoring mechanisms to track the state of function blocks and the overall system. When certain conditions are met or errors are detected, the system can trigger reconfiguration processes.

3.3 5G and beyond as an enabler for Dynamic Reconfiguration

The rapid advancement in networking and communication technologies, particularly cellular communication (5G and beyond), presents significant opportunities for the development of Industry 4.0 and smart manufacturing. 5G and next generations of mobile communication offer ultra-low latency, high bandwidth, and massive device connectivity, enabling real-time communication and control of distributed industrial systems. The IEC-61499 standard, a widely adopted standard for designing distributed control and automation systems, provides a framework for managing the re-configurability of heterogeneous devices in industrial systems. This section discusses the potential of integrating cellular networking within the IEC-61499-based industrial systems, enabling the realization of highly flexible, adaptable, and efficient manufacturing processes.
New generations of mobile network offer several enhancements over previous generations, including Ultra-low latency, High bandwidth, and massive device connectivity. In fact, since 5G, mobile network provides latencies as low as 1 ms, allowing real-time communication between devices, control systems, and other system components. With peak data rates of up to 20 Gbps, it can accommodate the transfer of large volumes of data, facilitating efficient communication between a vast number of devices. Furthermore, mobile networks can support up to one million devices per square kilometre, enabling seamless connectivity for a multitude of industrial devices and sensors. These capabilities make it an ideal candidate for the implementation of industrial systems that demand high performance, reliability, and flexibility.

The IEC-61499 standard provides a framework for designing distributed control and automation systems, comprising:

- Function blocks: Encapsulates control algorithms, data processing, and communication interfaces.
- Applications: Represents a collection of interconnected function blocks that describe a specific control task.
- Devices: Hosts function block applications and manages the execution of control tasks.
- System: Comprises a set of devices that are interconnected through communication networks.

Integrating mobile networking capabilities into IEC-61499-based industrial systems can result in several benefits. Mobile's ultra-low latency and high bandwidth can facilitate seamless communication between devices, allowing for rapid reconfiguration of the system. This can enable the dynamic addition or removal of devices and function block applications, as well as real-time adaptation to changes in the manufacturing process. The massive device connectivity offered by mobile networks enables IEC-61499-based systems to accommodate a large number of devices. This enhances the scalability of the system, allowing it to cope with growing demands and adapt to various types of manufacturing processes. Its low latency enables real-time data processing and decision-making, allowing industrial systems to react promptly to changes in the environment. This can result in improved operational efficiency, reduced waste, and better overall system performance. The integration of mobile networks into IEC-61499-based systems can provide improved security features, such as authentication and encryption mechanisms. This can help protect sensitive data and prevent unauthorized access to control systems, ensuring the safety and integrity of industrial processes.

The integration of mobile networking within IEC-61499-based industrial systems can revolutionize the way manufacturing processes are managed, offering enhanced re-configurability, scalability, real-time data processing, and decision-making capabilities. By leveraging the benefits of mobile technology, industrial systems can become smarter, more adaptable, and efficient, contributing to the realization of Industry 4.0 and the advancement of smart manufacturing. 5G technology also provides several capabilities that support the re-configurability of the dynamic network topology in a system utilizing the IEC 61499 standard.

One key feature is network slicing, which allows the creation of virtual networks within a single physical infrastructure. Each network slice can have its own specific network topology configuration, enabling dynamic creation and modification of network slices to accommodate changes in the dynamic network topology of the IEC 61499-based system. The low latency and high bandwidth offered by mobile networks are essential for maintaining a dynamic network topology. The low latency ensures quick response times for network reconfiguration commands, while high bandwidth allows efficient exchange of topology information between devices, sensors, and control units. Another important aspect is the massive machine type communication (mMTC) capability allows a large number of devices to be connected simultaneously, supporting seamless integration of diverse devices into the IEC 61499-based system. As the dynamic network topology evolves and new devices are added or removed, mobile's mMTC ensures continuous connectivity and communication.

5G's edge computing capabilities bring computation and data processing closer to the network edge, reducing latency and enabling faster decision-making during network topology reconfiguration. By utilizing edge computing resources, the IEC 61499-based system can perform local computations, analyse network topology changes, and dynamically adapt the network configuration in real-time. Another important feature is software-defined networking (SDN), a key principle of networks. SDN enables programmable control of network resources, allowing for dynamic reconfiguration of network elements, including the dynamic network topology. By leveraging SDN, the IEC 61499-based system can use mobile's programmable network infrastructure to modify network connections, establish new links, or remove existing ones as the dynamic network topology changes.

These capabilities collectively enable the IEC 61499-based system to achieve enhanced re-configurability of the dynamic network topology. The system can dynamically adapt its network connections, establish new communication links, and adjust resource allocation based on changing system requirements or environmental conditions. This ensures seamless connectivity, optimized communication performance, and efficient data exchange within the dynamically changing network topology.

Looking into the future, we are witnessing the emergence of new concepts under future generation of cellular connectivity. The concept of "physical network and digital network convergence" refers to the integration of traditional physical infrastructure (like communication networks and devices) with modern digital networks (such as the internet and cloud services). This integration creates a seamless environment where both physical and digital systems can communicate and interact effectively. This convergence has the potential to bring about significant advancements in various domains, especially with regards to the integration of artificial intelligence (AI) technology.

A breakdown of how this convergence enables different types of communication and interactions is presented below:

- AI Everywhere: By merging physical and digital networks, AI systems can be seamlessly integrated into various aspects of our lives. This means that AI technologies, such as virtual assistants, chatbots, and predictive algorithms, can be present and operational in both physical environments (like smart workplaces) and digital platforms (like websites and apps).
- Communication between Human and Digital Person: The convergence allows for improved communication between humans and digital entities that simulate human-like interactions. For instance, virtual avatars, chatbots, and other AI-driven characters can interact with humans in a more natural and meaningful way. This creates opportunities for enhanced customer service, personalized assistance, and even companionship.
- Communication between Human and Robotics: The integration of physical and digital networks enables smoother communication between humans and robots. This is especially relevant in scenarios where humans work alongside robots, such as manufacturing, healthcare, and logistics. The convergence ensures that human instructions, feedback, and commands can be seamlessly transmitted to robots, enhancing collaboration and efficiency.
- Communication between Robotic Systems: The convergence also facilitates communication between different robotic systems. Robots can share information, coordinate actions, and work together more effectively. This has applications in fields like swarm robotics, where multiple robots collaborate to achieve complex tasks.
- Entrance and Content for Intelligent Agents: The converged network serves as a gateway for intelligent agents, which are AI systems capable of performing tasks autonomously. These agents require access to data, information, and instructions to function effectively. The converged network provides a platform for these agents to access the necessary resources, enabling them to perform tasks ranging from answering queries to controlling robotic systems.
- Content Delivery: The convergence also enables efficient content delivery across various platforms. For instance, an AI assistant can seamlessly provide information and services to users, regardless of whether they are interacting through a physical device or a digital interface.

The convergence of physical and digital networks lays the groundwork for a world in which AI becomes an integral part of every facet of our lives. This convergence facilitates seamless communication between AI-powered entities and humans, enabling them to work together and collaborate with other AI systems. This interconnected network serves as a channel for intelligent agents, granting them access to the information necessary for tasks and valuable service provision. This convergence has the potential to transform how we engage with technology and how technology interacts within itself. Please note that this vision is still under construction and Zero-SWARM is in a perfect time to influence the future of connectivity industry aligned with the needs of IEC-61499.

The mobile core network establishes the technological underpinning that empowers the smooth amalgamation of physical and digital networks for AI-driven interactions. Its attributes such as ultra-high speed, low latency, extensive connectivity, edge computing, and support for immersive experiences play a pivotal role in realizing seamless communication between humans, digital entities, and robotic systems. This convergence ushers in a new era of interactivity, where AI agents comprehend and respond to human intentions, collaborate with fellow robots, and engage in unprecedented interactions. To bring the envisioned solution to fruition, here we include some suggestions for future generation of cellular networks to support better IEC-61499 platform extensions for dynamic setup and reconfiguration of agents:

Device-Edge-Cloud Integrated Architecture： Devices have the limitation in computing resources, cost, and batteries. Network can support large models that cannot be deployed in devices. Even for the models which can be deployed in devices, there is trade-off between inference speed and accuracy. The trade-off can be broken by leveraging edge and cloud resources. For this purpose, we may study several topics, including resource/task management and scheduling mechanism, distributed AI framework, data collection and transmission, privacy preserving in the device-edge-cloud integrated architecture.

Digital ID - Describing Digital Person/Robotics/Human: It involves developing a comprehensive digital identity (Digital ID) system that accurately represents and distinguishes between digital entities such as digital persons, robots, and humans. Including:

- Digital Identity for Digital Persons: Creating a system that assigns unique identifiers and attributes to digital entities that simulate human-like interactions, such as AI-powered chatbots and virtual avatars. This helps in recognizing and interacting with these entities effectively.
- Digital Identity for Robotics: Developing a framework that establishes digital identities for robotic systems, enabling seamless communication and authentication between robots and network services. This is crucial for secure coordination and control of robots.
- Digital Identity for Humans: Designing a robust digital identity system that represents human users in the digital realm, ensuring privacy, security, and accurate representation in online interactions.

New Protocol between Device and Network: It involves designing and developing new communication protocols that improve the interaction between devices and networks, enhancing efficiency and capabilities.

- Efficient Data Transmission: Researching protocols that optimize data transmission between devices and networks, considering factors such as latency, data volume, and energy consumption.
- Quality of Service Enhancement: Developing protocols that prioritize certain types of data traffic, ensuring that critical data (such as real-time video streams or robotic control signals) receives the necessary network resources and low latency.
- Support for Emerging Technologies: Designing protocols that can accommodate emerging technologies like IoT devices, edge computing, and AI agents, ensuring seamless communication and collaboration across diverse platforms.
- Security and Privacy: Researching protocols that enhance security and privacy during data transmission, including encryption, authentication, and data integrity measures to protect sensitive information.

Zero-SWARM under WP3 will go deeper on these items and will provide solutions to WP5 where these extra cellular capabilities could support dynamic setup an reconfiguration of agents using IEC-61499.

3.4 Core steps to extend IEC 61499 for Dynamic Setup & Reconfiguration of Agents, also considering the CPSoS

IEC 61499 standard defines the architecture and engineering of distributed control systems in industrial automation. It provides a framework for designing and implementing control systems using function blocks, which are modular components that encapsulate control logic.

To extend IEC 61499 for managing automatic reconfiguration in manufacturing contexts, including the automatic recognition of devices like, for example a new robot or a new automatic guided vehicles (AMRs) added to a fleet of AMRs, is crucial to implement the following steps:

Dynamic Device Discovery: Implement a mechanism to automatically discover and recognize new devices in the system. This can be achieved through various methods such as network scanning, device broadcasting, or using specific protocols for device identification.

Device Profile Definition: Define device profiles that describe the capabilities, interfaces, and behaviour of different devices. These profiles should be standardized and adhere to a common format so that the system can interpret and utilize them during reconfiguration.

Reconfiguration Engine: Develop a reconfiguration engine that can handle the dynamic addition or removal of devices. This engine should be able to identify the changes in the system configuration, update the control logic accordingly, and deploy the necessary resources to accommodate the new devices.

Event and Notification Handling: Establish a mechanism for devices to communicate their presence and changes to the system. Devices should be able to send events or notifications when they are added, removed, or experience any changes in their status. The system should be capable of receiving and processing these events to trigger the appropriate reconfiguration actions.

Function Block Management: Modify the existing function block management mechanisms of IEC 61499 to handle reconfiguration. This includes updating the function block instances, their connections, and their parameters based on the changes in the device configuration. The reconfiguration engine can utilize this information to adapt the control logic accordingly.

Runtime Environment Adaptation: Enhance the runtime environment to support automatic reconfiguration. This involves providing the necessary tools and interfaces for managing device discovery, device profiles, reconfiguration actions, and event handling.

Scalable Architecture: The architecture of the control system should be designed to scale and handle the dynamic nature of the setup. This may involve distributed control, modular function blocks, and a flexible infrastructure that can accommodate the addition or removal of devices without disrupting the overall system operation.

By considering these factors, IEC 61499 can be extended to effectively manage dynamic setups. The automatic recognition of devices, real-time reconfiguration, event-driven adaptation, and scalability of the architecture can help create a robust and flexible control system capable of handling the constant changes and reconfiguration requirements in dynamic manufacturing environments. About the architecture see deliverables 4.2 and 4.4.

3.5 Auto-Recognition and Re-configurability data & service in the Reepack use case

Reepack Use-Case - SN1/2 Trials - is based on a Compact Packaging Line (SN1) and ReeNEXT Platform I4.0 Compliant (SN2) with refers to a typical Industrial Production Line for Packaging of Ready-Meals.

Figure 1 Reepack Production Line Example

Specifically, ReeNEXT Platform is characterized by a Main User Web-Interface for Data Monitoring and Management, including additional Modules for Production Planning, Predictive and Predictable Maintenance, Cost Analysis and others.

Thanks to Production Planning ReeNEXT Module, the Production Manager can Create and Supervise each Internal Production Order, assigning:

1. The Production Line to be used according to the Type of required Packaging and the respective Availabilities, considering that in the same Factory there could be different Packaging Line based on same or different Machine Types;
2. The Priorities, when the Order starts and finishes according to Machine Production Capacity and Products to be packed;
3. The Packaging Materials (Trays and Film) and the Food-Product to be prepared in advance by other Departments (Warehouse and Kitchen);
4. The Machine Recipe to be executed by the Machine Operator because all Machine Set-Ups are

properly defined to obtain the requested Packed Product.

Depending by the Customer's Reference/Brand and Ready-Meal Recipe the production line re-configuration, the Logistics of Packaging Materials (in terms of Trays and Film) and Food-Product (in terms of possible different Ingredients to be filled into the Trays) can present Different Levels of Complexity and Tasks. The reconfiguration of the production line has direct implication on logistics operations, such as the re-filling of specific items.

In addition, in case of Scheduled or Unpredictable Maintenance Activities, New Tasks can increase the List, and Complexity, of previous Tasks. For example, sometimes is needed to reschedule the Priorities because Spare Parts becomes urgently necessary for the Maintenance Manager.

The Logistics of Packaging Materials and Food Product, Spare Parts as well, is traditionally in charge to Human Operators, in Zero-SWARM - SN1/2 Trials the AMR Device will be integrated into the Packaging Production Line.

AI Algorithms for AMR Jobs Scheduling and Re-Scheduling, according to the consequent Production Orders Re-configurability, will be carried out engaging a Single Physical ARM (1st Step) and a Fleet of ARMs (2nd Step - one Physical and Other Ones through the Virtual Commissioning Interface), so that contextually the Auto-Recognition and Re-configurability features will be tested as well.

A dedicated New ReeNEXT Module for AMR and Fleet Management will be implemented with an own User Interface for:

1. AMR Mission Declaration according to Production Orders Planning: according to Mission List the AI Algorithms will generate ARM Tasks and relative Priorities;
2. AMR Task Monitoring: the Production Manager could be constantly updated about the Task Execution and ARM Status;
3. Last but not least, Single or Fleet AMR Auto-Recognition: according to the List of ARM Devices intercepted into the Factory Network, when not allowed an Automatic Auto-Recognition Process and AMR Authorization, the Production Manager could authorize the use of Single or Fleet AMR.

4 Reconfiguration functionalities implementations aspects

4.1 Example of Re-configurability Functionalities applied to AMR Fleets

In the context of fleets of Automated Mobile Robots (AMRs), re-configurability offers several specific scenarios that are interconnected, allowing for enhanced operational efficiency and adaptability. The dynamic path planning capability enables AMRs to adjust their planned paths based on real-time information. By doing so, they can avoid obstacles or congestion and find alternative routes to reach their destinations efficiently. This flexibility is further complemented by load balancing, where re-configurability allows for the distribution of tasks and workload among the AMRs in the fleet. AMRs can be dynamically assigned tasks based on their proximity or current workload, optimizing the overall efficiency of the fleet. Moreover, re-configurability facilitates adaptive formation, empowering AMRs to dynamically adjust their formations to suit specific task requirements or operational conditions. By doing so, different formations can be created to optimize operations such as picking, storage, or transportation based on the size and shape of the items.

To support these dynamic operations, reconfigurable docking stations play a crucial role. They can adjust their configurations to accommodate different types or sizes of AMRs. AMRs can autonomously identify and connect to the appropriate docking station that matches their requirements, enabling seamless integration and facilitating charging or replenishment operations.
The system's ability to expand or reduce its size is also important. Re-configurability allows the fleet to easily adapt to changes by adjusting its communication and coordination protocols when new AMRs are added, or existing ones are removed. This ensures efficient integration of new vehicles and maintains operational stability when modifications are made.
Additionally, re-configurability plays a significant role in fault tolerance. When unexpected events such as AMR malfunctions or communication failures occur, the system can dynamically reconfigure the fleet's operation by redistributing tasks to other available AMRs. This ensures uninterrupted performance and maintains the overall efficiency of the fleet.

Finally, re-configurability enables the system to dynamically adjust tasks prioritization and reprioritization based on changing requirements or urgent requests. This ensures that the fleet can respond promptly to emerging higher-priority tasks or changing customer demands, optimizing its operation accordingly.

Overall, these interconnected scenarios of re-configurability empower AMR fleets to adapt, optimize, and improve their operations. By establishing links between different aspects, such as dynamic path planning, load balancing, adaptive formation, docking stations, system expansion/reduction, fault tolerance, and task prioritization, the system can achieve enhanced efficiency, flexibility, fault tolerance, and overall performance.

The advent of advanced technologies, such as mobile networks and the IEC-61499 standard, has significantly impacted the way industries manage and control their Automated Mobile Robot (AMR) fleets. These advancements have led to the development of a semantic meta-model that enables the support of Cyber-Physical Systems of Systems (CPSoS) and offers re-configurability, auto-discovery, and cyber security services.
The definition of a semantic meta-model is crucial for establishing an interoperable framework that allows seamless communication and collaboration between the heterogeneous components of an AMR fleet. This meta-model consists of a set of ontologies and rules for defining the relationships between the different entities in the CPSoS. By leveraging the IEC-61499 standard, which provides a common framework for distributed control systems, the semantic meta-model ensures that the overall system is modular, scalable, and adaptable to changes in the industrial environment.
Incorporating auto-discovery capabilities within the AMR fleet enables the system to automatically detect and integrate new AMRs as they are added to the fleet. This is particularly important for industries that require dynamic and flexible manufacturing processes. The use of mobile networks facilitates the auto-discovery process by providing reliable and high-speed communication channels. This enables the AMR fleet to quickly exchange information with other devices and systems, allowing for efficient coordination and real-time decision-making.
Re-configurability services are essential for maintaining the resilience and adaptability of the AMR fleet in the face of changing requirements and conditions. In an industrial environment, it is common for production lines and processes to undergo modifications, either due to the introduction of new products or changes in customer demand. The semantic meta-model, coupled with the IEC-61499 standard, provides the necessary tools for engineers to easily reconfigure the AMR fleet in response to these changes. This includes the ability to modify the control logic, adapt to new communication

protocols, and adjust the AMR's operational parameters, ensuring that the fleet can continue to function efficiently and meet the evolving needs of the industry.

As the AMR fleet becomes more interconnected and reliant on data exchange, ensuring the security of these systems becomes paramount. Cybersecurity measures must be integrated into the design and implementation of the semantic meta-model to safeguard the AMR fleet from potential threats. This includes securing communication channels, implementing access control mechanisms, and ensuring data integrity and confidentiality. The mobile network plays a vital role in this aspect, as it provides advanced security features and encryption techniques that can be leveraged to protect the AMR fleet from cyber-attacks and unauthorized access.

The integration of a semantic meta-model with the IEC-61499 standard and the use of mobile networks have paved the way for advanced re-configurability and auto-discovery capabilities in AMR fleets. Furthermore, these technologies have enabled the implementation of robust cybersecurity measures to protect the fleet from potential threats. As industries continue to evolve and adopt innovative manufacturing processes, the development of such agile and secure AMR fleet management systems will become increasingly important to maintain efficiency and productivity in the industrial context.

4.2 Auto-Recognition and Re-configurability data & services Extending IEC 61499

Considering AMRs Fleet, auto-discovery and reconfiguration capabilities enable the fleet to adapt dynamically to changing operational needs, optimize tasks allocation, and improve overall efficiency and flexibility.

Reconfiguration in the context of a fleet of AMRs can include both automatically adding new AMRs to the fleet (using auto-discovery) and on-the-fly loading of function blocks on individual AMRs to modify their behaviour during runtime.

Example of automatically adding a new AMR to the fleet:

When a new AMR joins the fleet, it needs to be integrated seamlessly into the existing system without manual intervention. This process involves automatically recognizing the new AMR, configuring its communication interfaces, and enabling it to participate in the collective tasks performed by the fleet. To achieve automatic addition, is needed to implement the auto-discovery mechanism we discussed earlier. When a new AMR comes online, it should announce its presence and capabilities using zero-configuration networking (Zeroconf) protocols or any other appropriate discovery mechanism.[1] The Fleet Management Application detects this announcement and automatically integrate the new AMR into the fleet, making it part of the distributed control system.

Example of adding on-the-fly function blocks:

Considering AMRs, function blocks represent specific control logic units or behaviours that define how the robot should navigate, avoid obstacles, perform specific tasks, etc. Reconfiguration, in this context, refers to the ability to modify or load new function blocks on the AMRs while they are in operation, allowing them to adapt to changing requirements or tasks.

[1] Please note that Zero Configuration Networking (zeroconf) protocols (https://www.ietf.org/proceedings/55/149.htm) are under the scope of IETF (Internet Engineering Task Force) and therefore they are at the moment out of the mobile networks scope (under 3GPP). However, similar concepts can be introduced into mobile networks to support automatically adding new AMRs to the fleet.

For example, suppose an AMR needs to switch from a "navigation" function block to a "delivery" function block to perform a specific task. In that case, this can be achieved through on-the-fly reconfiguration. Similarly, if an updated version of a function block is available, it can be loaded onto the AMRs without stopping the entire fleet.

The following schema shows the auto-discovery and reconfiguration block diagram allowing the AMR Fleets Management Application to identify, acknowledge and include in the fleet a new-appeared AMR.

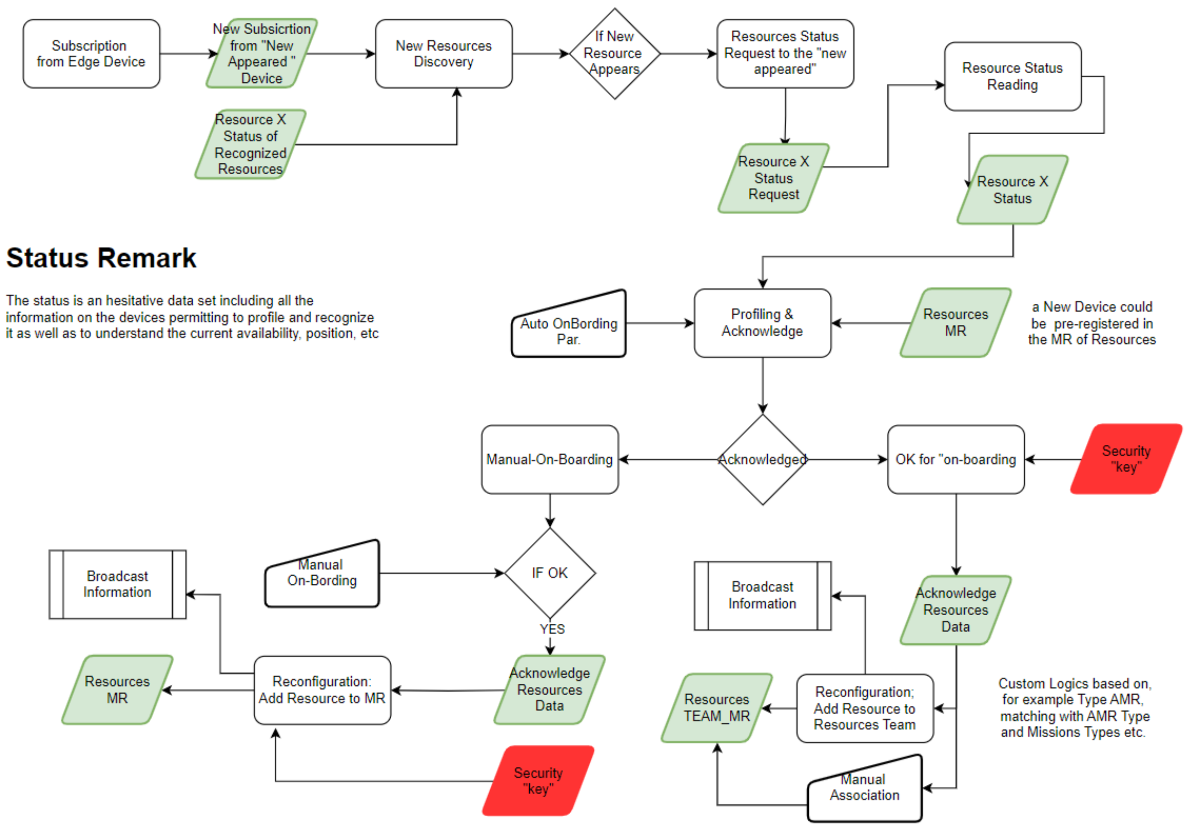

Figure 2 Status Remark

The AMR, to be recognized, must subscribe to the system, for example sending the message via MQTT. A specific service of the Fleet Management Application sends a message to the novel devices asking for its status. The status is a wide set of information, including those needed for the recognition and the profiling, and those permitted to understand the current status, such as the position, the velocity etc. The status is read by a service. At this point the Profiling and Acknowledge Service, evaluates if the new device is already pre-registered on the Master Record. If yes the new Devices is on-boarded, the Master Record updated and the AMR added to the proper fleet by a service or by the input of an operator.

If the new device is not pre-registered in the MR, the on-boarding could occur by operator or automatically based on configuration parameters which establish different behaviour based on the specific application. The On-Boarding and the Fleet information are broadcasted for other services and applications needing such information. The following UML sequence diagram explains the data exchange between the different involved actors.

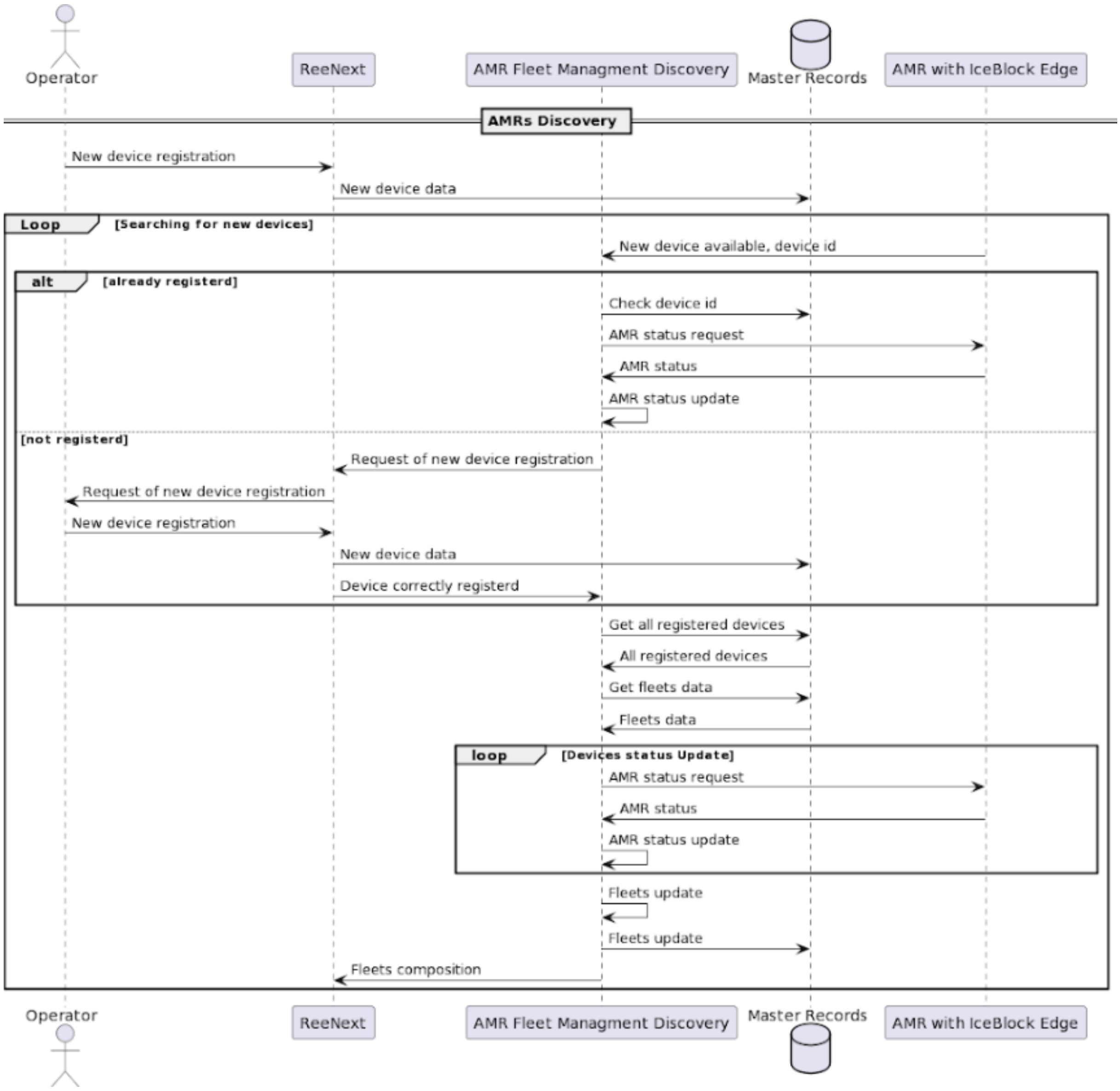

Figure 3 Sequence diagram AMRs Discovery

4.3 Re-configurability Semantics

Re-configurability semantics in the context of a processing station refers to the systematic and well-defined way in which changes in the configuration of mechatronic components, such as Drill components, are managed and understood within the system. It encompasses the rules, processes, and capabilities that govern how components can be added, removed, or relocated, and how these changes are recorded and understood within the processing station's semantic model, typically stored in a Graph Database (Graph DB).

Here are key aspects of re-configurability semantics in a processing station:

1. Dynamic Configuration: Re-configurability implies that the processing station can adapt to changing requirements or scenarios. Components like Drill agents can be added or relocated on the fly to meet specific manufacturing needs. This dynamic configuration capability allows for versatility in the production process.

2. Capability and Skill Assignment: When a new Drill component is introduced or an existing one is

reconfigured, the system assigns specific capabilities and skills to the component. These capabilities define what the Drill can do (e.g., drilling depth) and the associated skills outline how these capabilities are invoked and used.

3. Graph Database Integration: The re-configurability semantics are often integrated into a Graph Database (GraphDB). The GraphDB stores the ontology and the semantic models, allowing for efficient querying, reasoning, and constraint-checking. This integration ensures that the processing station can make informed decisions based on its current configuration.

4. Querying, Reasoning and Constraint-Checking: The semantic model and the associated re-configurability semantics enable sophisticated querying and reasoning capabilities. Operators or automated systems can query the GraphDB to understand the current state of the processing station, check compatibility between components, or assess the impact of a reconfiguration. Re-configurability semantics also support constraint-checking. This means that before a reconfiguration is executed, the system can validate whether the proposed changes comply with predefined constraints or rules. For example, it can check if the new Drill component is compatible with the workpiece dimensions.

4.4 Implications of the Re-configurability versus Digital Twin

In a system based on the IEC 61499 standard and utilizing digital twins, reconfiguration can be achieved through a well-defined process that establishes connections between various steps.

The system is represented using the IEC 61499 standard, providing a modelling framework for distributed control systems. Components, functionalities, and interconnections are modelled using function blocks, event connections, and data connections.

Digital twins, which mirror the behaviour, state, and attributes of physical components, are integrated into the system. These digital twins are connected to their physical counterparts, facilitating real-time data exchange and synchronization.

Continuous monitoring by digital twins enables data collection regarding the state of physical components, operational parameters, performance metrics, and environmental conditions. Based on predefined conditions, a reconfiguration trigger is defined. This trigger may be related to changing system requirements, detected anomalies, or predefined events. Once triggered, the reconfiguration process begins.

Data collected from digital twins is analysed to assess the current system state and determine the necessary reconfiguration actions. The analysis involves comparing the current state with predefined rules, performance thresholds, or optimization objectives. Using the analysis results, a reconfiguration plan is generated. This plan outlines the required changes to the system, including modifying interconnections, updating control logic, adjusting parameters, or adding/removing components.

The reconfiguration plan is executed dynamically, with system components and digital twins collaborating to implement the planned changes. This involves updating the behaviour of function blocks, establishing new connections, reassigning tasks, or modifying the configuration of digital twins. After reconfiguration, the system's behaviour is validated and tested to ensure it meets the desired objectives. Simulation, controlled environment tests, or phased introductions of changes may be conducted to assess their impact.

The digital twins continue to monitor the reconfigured system, collecting data and providing feedback on its performance. This feedback loop enables continuous adaptation and further reconfigurations, if necessary.

By establishing links between modelling, digital twins, monitoring, trigger-based reconfiguration, analysis, plan generation, dynamic execution, validation, and continuous monitoring, the system achieves a cohesive and interconnected reconfiguration process. This integration allows for efficient and effective adaptation, optimization, and performance improvement based on real-time data and evolving system requirements.

4.5 CPSizer Layer for Re-configurability

Central Node Trial 2 aims to implement the production of a fuel cell in IEC 61499 with function blocks as a feasibility study. Based on the functional and technical requirements from D5.1 CN T2 intends to integrate a ROS2-IEC61499 interface. The software models that are shown in Figure 1 are described in the following.

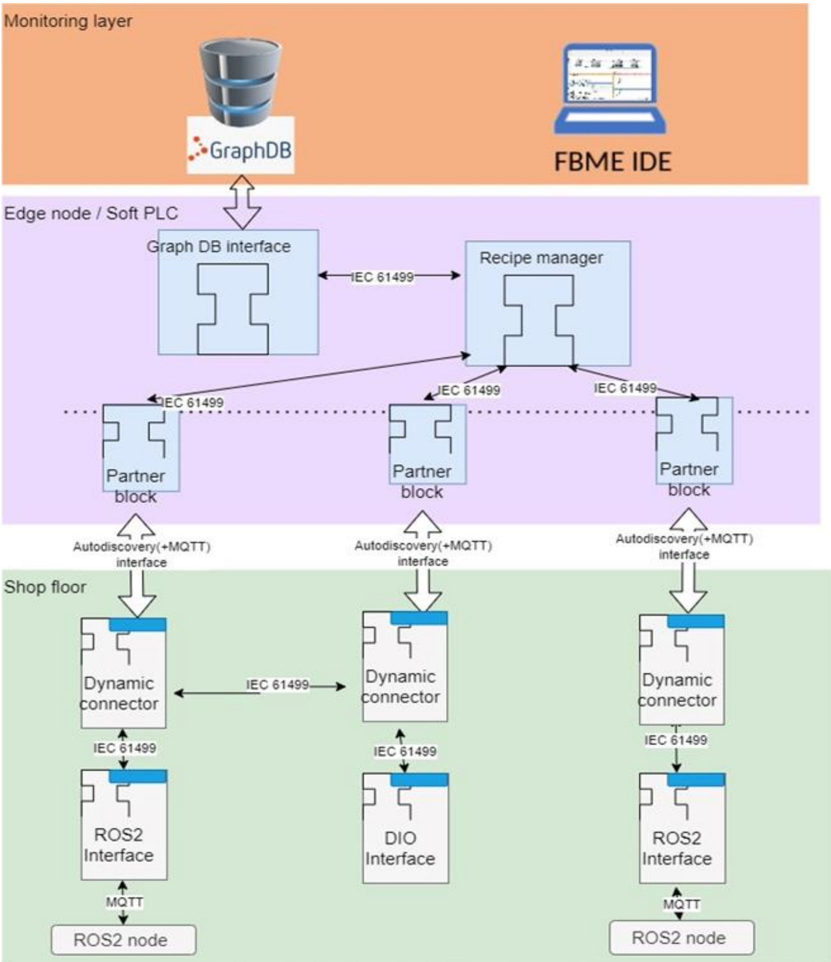

Figure 4 Central Node Trial 2 Data Exchange Architecture

Shop floor is a software layer which provides functionality to connect flexible reconfigurable agents to IEC 61499 control system. IEC 61466 **ROS2 interface** manages connection from IEC 61499 to ROS2 devises via MQTT channel. **DIO interface** manages connection to input and output digital signals of device. The **Dynamic connector** interface controls the connection to the Edge layer and transmits data to **ROS2 interface** and **DIO interface**.

Edge/ Soft PLC layer is a software layer which orchestrates the system's high-level control logic. **IEC 61499 Recipe manager** contains high-level control logic and interact with agents via **IEC 61499 Partner block**. **IEC 61499 Partner block** communicates with **Dynamic connection** block on a particular agent. Also through this interaction the information about the skills of a particular agent is transmitted. **Recipe manager** transmits the received data about agent skills to the **IEC 61499 GraphDB interface.**

In **Monitoring layer**, cloud database GraphDB stores data about reconfiguration of the system and agent skills. **FBME IDE** receives updates from GraphDB and displays the system changes.

4.5.1 ROS2-MQTT-Interface

ROS2 is well established in robotics for modular and distributed real-time applications and runs in Windows and Linux systems. IEC 61499 runtimes offer a mapping to MQTT. Thus, MQTT can be used as middleware between ROS2 and IEC61499.

Both MQTT and ROS2 implement the publish-subscribe messaging pattern which is beneficial for a dynamic reconfiguration because the sender (publisher) and receiver (subscriber) of messages do not need to directly know each other, but instead communicate via predefined topics where they either publish on or subscribe to. A specific topic defines the structure of the messages that contain the actual information.

Previous work of Aalto [Ref 4.5.1.1] offers an ROS1-IEC61499 interface. Since ROS1 does not support real-time communication which ROS2 does [Ref 4.5.1.2], CN Trial 2 intends to upgrade the existing interface to ROS2. Further CN Trial 2 aims to additionally offer a programming interface by IEC61499 function blocks for a robotic arm and not only an AGV. The real-time control services for the AGV and the robotic arm are implemented in ROS2 and will be accessed from IEC61499 via the ROS2-IEC61499 interface.

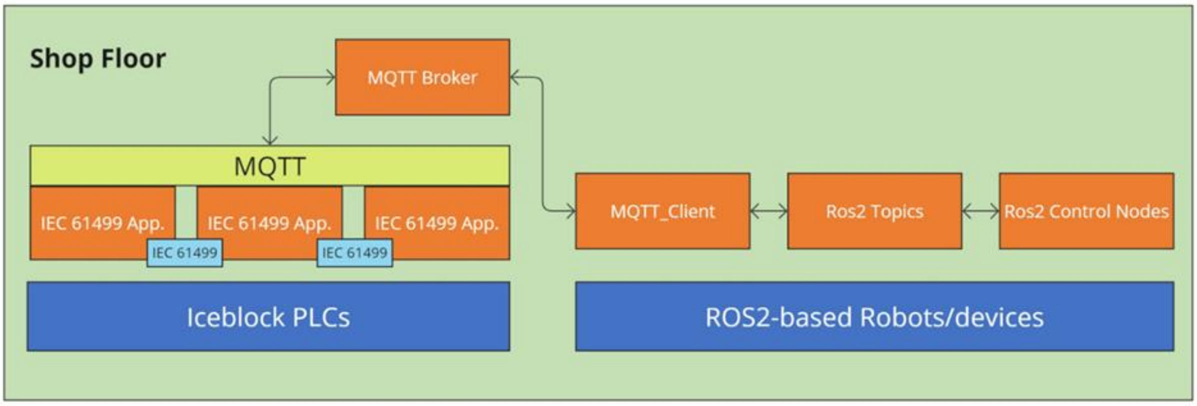

Figure 5 Shop Flow

The interface is achieved via a three-step process involving an IEC 61499 application, MQTT broker, and an MQTT-ROS bridge (MQTT_Client). In the initial phase, the IEC 61499 application, comprising Function Blocks (FBs) with MQTT client functionalities, publishes data messages to an MQTT broker.

Subsequently, the MQTT broker, positioned as the central communication hub, receives, filters, and directs these messages from the IEC 61499 application, enabling a connection between the distributed control system and ROS2 via MQTT.

In the final stage, the MQTT-ROS bridge acts as the crucial link, subscribing to broker topics, converting incoming MQTT messages to ROS2 messages, and publishing them to ROS2 topics. The data flows from ROS2 control nodes to IEC 61499 control applications is based on the same architecture and logic as above introduction. This methodology achieves an efficient ROS2-IEC61499 communication interface, facilitating seamless data flow between the systems.

4.5.2 Dynamic Connector & Partner Block

The idea behind the dynamic connector and partner blocks is to have a smooth integration of additional operators such as Automated Mobile Robots (AMRs) into the production scenario running on the IEC 61499 standard. The goal is that the additional agent/operator can be synchronized or added as a dynamic functionality to existing parts of sections of the machine. In summary, the system's operation would be such that when an agent (AMR in our case) is ready and available for operation, it would send out a broadcast signal which would be received by an Agent Manager (AM). The agent manager is a software entity, a function block in the system that would take care of the detection and onboarding of such agents. Once the agent manager receives information about the available AMR, the agent manager would cross-check this information with the system orchestrator.

Based on the response, the agent manager would link the AMR to the respective dynamic partner block suggested by the orchestrator. After this onboarding process is completed, the AMR and the dynamic block would communicate directly, that is, without the involvement of the agent manager or system orchestrator.

Figure (4.5.2.1) below shows the system overview of the proposed dynamic connector and partner blocks. The full setup consists of three different stages which will be explained in the subsequent subsections.

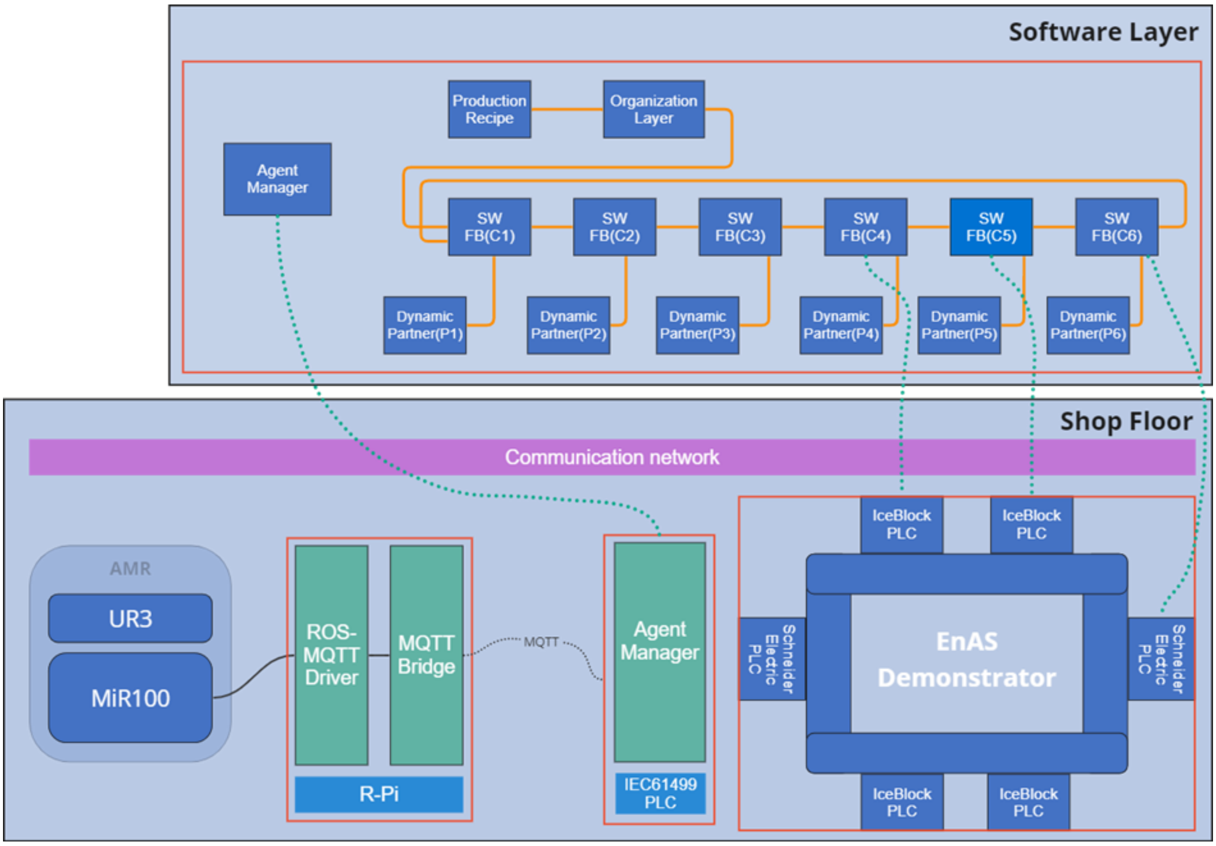

Figure 6 System overview of the proposed dynamic connector and partner blocks

Technical setup

In Figure (4.5.2.1.), the shopfloor section shows the EnAS demonstrator which is a production line consisting of numerous conveyors to which the AMRs can be connected. Subsequently, each of these conveyors has an individual IEC 61499 FB in the software layer. Connected to each of these conveyor FBs is a dynamic partner block.

The job of the dynamic partner block is to communicate with an external agent/operator and have the capability to perform reconfiguration of this agent on-the-fly. For example, if the AMR is linked to dynamic partner (P2), then the operations of the AMR will be performed at conveyor 2 of the machine. In this case when the conveyor 2 does not need further assistance by the AMR, it can be linked to any other conveyor with the help of the agent manager and the dynamic partner block. Using this mechanism, it is possible to include multiple different AMRs into the production scenario.

The system uses MQTT as the communication mechanism between the dynamic blocks and the AMRs. The AMRs in this use case (MiR100) are powered by ROS, and the ROS topics can be published or subscribed to via ROS-MQTT interface.

Agent announcement and detection

When the AMR is initiated and is ready to be integrated into the system, it publishes a message on the general MQTT topic 'NewAgent'. The initial message can contain whatever information is desired by the users, in this use case with the initial message, we send the name, network address and capability of the AMR.

The agent manager subscribes to this topic and processes the received information. Once the agent manager confirms that the AMR is a new agent and is not already integrated into the system, it passes the processed information to the system orchestrator and asks if the system orchestrator would like to integrate the AMR into the system. If the system orchestrator decides to add the AMR to the system, it will respond to the agent manager along with the position and details of the dynamic connector block.

Agent addition

The agent manager would then confirm to the AMR about the addition into the system, following which the AMR would send detailed information about the various parameters it would need for successful operations. These parameters can vary depending on the type of AMR and user needs. In this use case we consider two parameters which are the desired coordinates for the AMR, and mission status updated by the AMR. The topics in our case are MQTT topics to which the agent, dynamic connector block, and the AMR can subscribe and publish based on the need.

Upon receiving the detailed information, the agent manager directs this information to the dynamic connector block indicated by the system orchestrator. The dynamic partner would subscribe and publish to the topics requested by AMR. Upon successful integration of the AMR to the dynamic connector block, the agent manager would send a confirmation to the AMR and the system orchestrator, following which the role of the agent manager in this case would be done.

Agent reconfiguration

In the case where the system orchestrator needs the AMR to be reconfigured to a different location, the system orchestrator would send a message to the agent manager, which would then reassign the AMR to the desired dynamic connector block. During this process, the agent manager would ensure safety and ensure the AMR is only reassigned when it is ideal and does not have any pending tasks.

4.5.3 GraphDB Interface

The Graph DB interface will help to integrate re-configurability semantics into a Graph Database (GraphDB). This interface acts as a vital link, connecting the GraphDB with the IEC 61499 application through a dedicated function block. Its primary purpose is to facilitate efficient communication and data exchange between the IEC 61499 application and the GraphDB, offering a range of essential functionalities.

The Graph DB interface will be designed as a Service function block that utilizes REST APIs to establish a connection with the GraphDB, providing a standardized and accessible method for communication. It offers the ability to execute various operations, including querying, reasoning, and constraint-checking. This integration helps for the processing station's adaptability, as it allows components like Drill agents to be added or relocated on the fly to meet evolving manufacturing requirements.

4.5.4 Recipe-Manager Interface

The Recipe-Manager sequences the production process can be programmed with function blocks in IEC 61499. It calls the control services for the hardware at the appropriate time points based on hardware states and sensor data. The low-level, real-time controllers for the hardware run as ROS2-services and their technical details are hidden from the function block programmer. Instead, the Recipe-Manager receives only data via the ROS2-IEC61499 interface that is required for the sequencing of the production process. Production specific parameters that the ROS2-servcies require for control are read from a configuration file by the Recipe-Manager as “ingredients” (hence the name) and are also passed via the ROS2-IEC61499 interface. These can be:

- process parameters:
    - map positions
    - object position,
    - object IDs,
    - predefined production programs on other machines,
- hardware parameters:
    - user frame,
    - tool centre point.

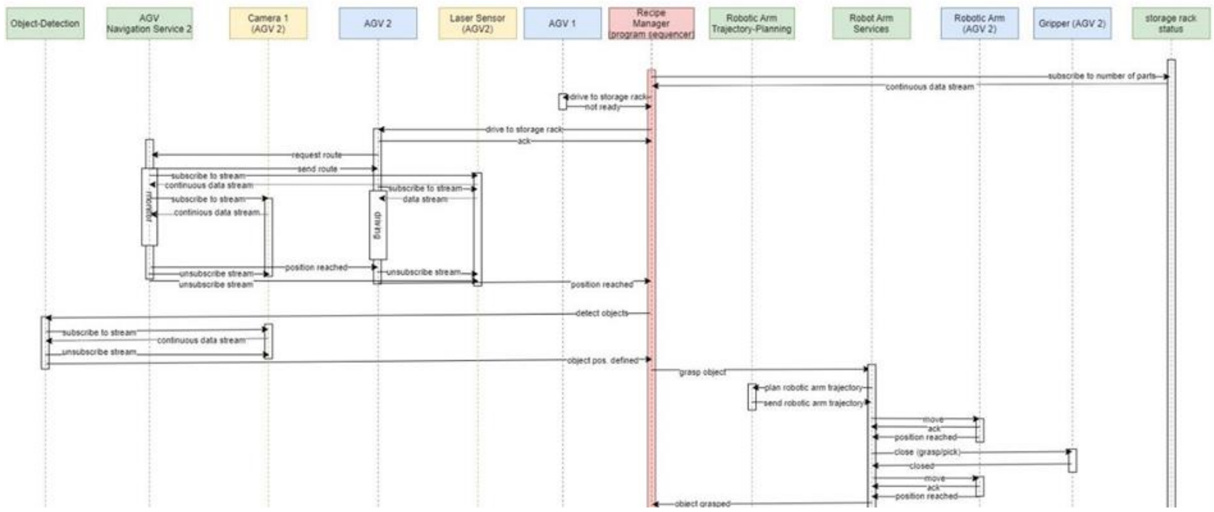

Figure 7 Exemplary communication sequence between Recipe-Manager (red), Actuators (blue), sensors (yellow) and control services (green)

The sequence diagram in Figure 4.5.4.1 shows the exemplary actions of the Recipe-Manager. The communication from and toward the Recipe-Manager (in red) passes the ROS2-IEC61499 interface, while the hardware (blue), the control services (green) or the sensors (yellow) communicate directly in ROS2. At the beginning of the use case for the production of a fuel cell, the Recipe-Manager subscribes to the charging level of a storage rack where the parts (e.g. bipolar plates) are stored either by human workers or mobile manipulators (AGV + robotic arm). Once the Recipe-Manager detects an available part, it requests an AGV to drive to the storage rack until one AGV acknowledges. Once the AGV has reached the storage rack, it informs the Recipe-Manager which then starts the object detection service. After the object position had been communicated to the Recipe-Manger, it passes the information toward the robotic arm service to perform a pick and place operation.

4.5.5 FBME IDE to GraphDB-Interface

FBME (Function Blocks Modelling Environment) is an integrated development environment (IDE) for IEC 61499 systems. It is an open-source, modular IDE based on IntelliJ and the Meta Programming System (MPS). FBME was designed to support modularity, cross-platform compatibility, and expandability. FBME provides a platform for seamless integration of development tools for IEC 61499 and research projects. Also, FBME provides an SDK for developing IEC 61499 instruments and tools. Instruments developed with this SDK can be seamlessly integrated into FBME as plugins.

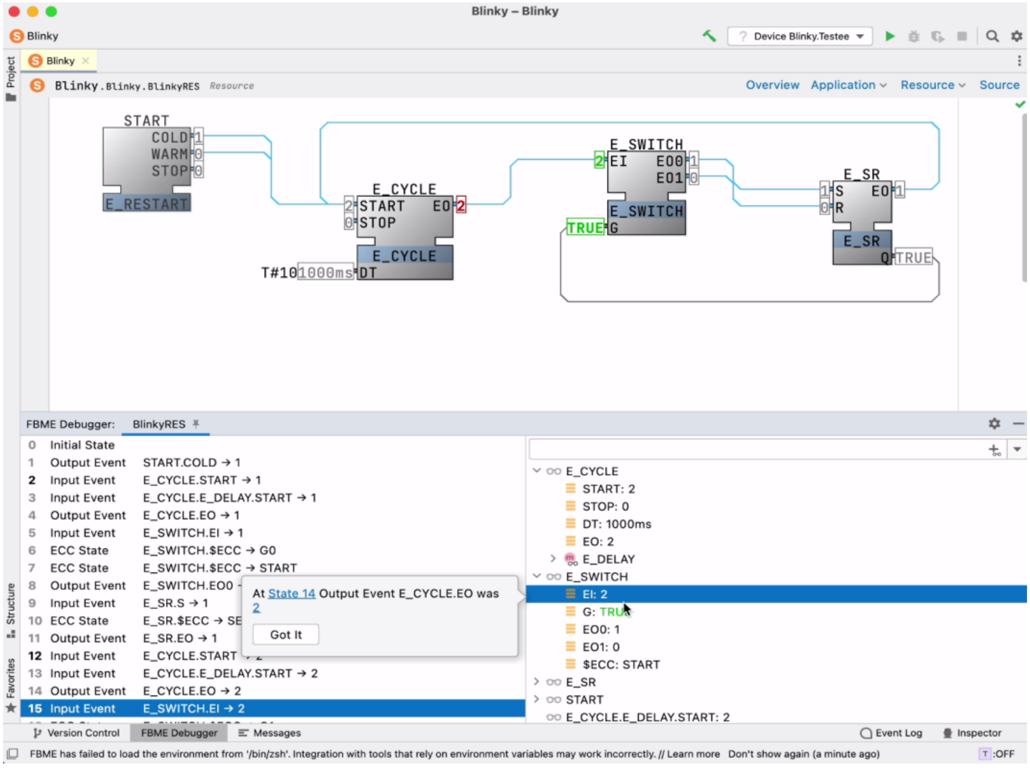

Figure 8 FBME IDE GUI

In this task we plan to develop a plugin for FBME IDE that will display information about changes in the state of the control system. The plugin will display the system reconfiguration, changes in its topology when new agents are connected or disconnected from the industrial system. The plugin will receive information about agents from the GraphDB interface.

4.5.6 Cybersecurity for Re-configurability

Cybersecurity for re-configurability in processing stations using blockchain is a critical and innovative approach to safeguarding the integrity, confidentiality, and availability of sensitive data and operations within such industrial environments. This concept leverages blockchain technology, known for its decentralized, immutable, and transparent nature, to address various cybersecurity challenges associated with reconfigurable processing stations. Here are key aspects of this approach:

1. Immutable Audit Trail: Blockchain creates an immutable ledger of all transactions and changes made to the processing station's configuration. Every addition, removal, or relocation of components, as well as any updates to the semantic model, is recorded as a tamper-proof entry. This audit trail enhances accountability and traceability, ensuring that any unauthorized or malicious changes are easily detectable.

2. Secure Identity Management: Blockchain offers robust identity management mechanisms. Each user or component accessing the processing station is assigned a unique digital identity stored on the blockchain. This ensures that only authorised entities can make changes to the system's configuration or access sensitive information, reducing the risk of unauthorised access.

3. Smart Contracts for Access Control: Smart contracts, self-executing agreements with predefined rules, can be employed to manage access control within the processing station. These contracts enforce permissions and access rights, ensuring that only authorised users or components can initiate

configuration changes or interact with the system.

4.6 Re-configurability impact on AAS

Flexibility and reconfiguration and responsiveness of production lines need to be increased implementing Plug and Produce (PnP) concept, which requires the digitalization of the resources/devices. The PnP concept can be accomplished by implementing standards that contain machine-readable self-descriptions and by encapsulating the different assets as Industry4.0 components using Asset Administration Shell (AAS).

In particular, AAS could be used either to implement Digital Twins itself or support some of its key functionalities, for instance, physical object encapsulation, data gathering and communication, interoperability across the solutions of different suppliers, reconfiguration, and intelligent decision support.

Digital Twin and Asset Administration Shell Concepts and Application in the Industrial Internet and Industrie 4.0, Industrial Internet Consortium, [Ref. 4.6.1].

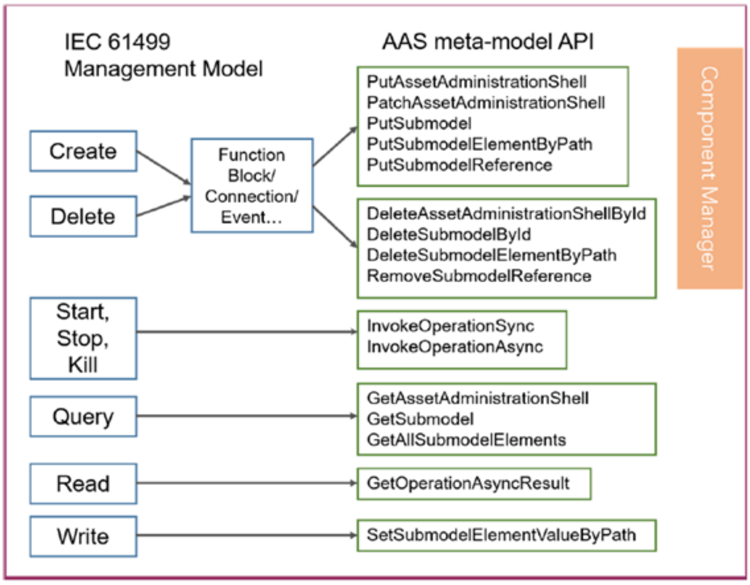

Figure 9 Management instructions mapping model from [Ref. 4.6.4]

The asset's information is described in the AAS as Submodels, which describes technical functionality exposed by the Asset Administration or respective assets. Below the Submodel level, there are the Submodels elements for storing specific data related to the Submodel. Also, the meta model of AAS allows reference properties and AASs. This encapsulation and hierarchical capabilities are described in IEC 612360 and the Industrial Digital Twin Association (IDTA) supports the AAS developments by publishing official templates for the Submodels (e.g., TechnicalData, OPC UA Server, etc.). The use of these templates will be possible to achieve interoperability between suppliers, thus allowing the reconfiguration of production lines thanks to following the same model and metamodel.

There are three types of AAS: passive, active and proactive. The first one is considered a static digital representation (i.e., a file), but the active and proactive AASs are able to update their properties, can be implemented as agents, and could help transform the current rigid automation pyramid to a flexible automation network where automation components with AAs can perform autonomously [Ref 4.6.2], like the features of the function blocks of IEC 61499.

Regarding the combination of AAS and IEC 61499, the ICE 61499 uses the function blocks to encapsulate algorithms that can be seen as independent module and has unified structure, which is highly similar to AAS Submodels. Moreover, both AAS and function blocks have features of encapsulation and flexibility. These similarities make possible to integrate IEC 61499 into the meta-model of AAS, and some work has been done including static and dynamic information [Ref 4.6.3].

5 Conclusions

This deliverable stands out from the state of the art by combining dynamic recognition, real-time reconfiguration, event-driven adaptation, 5G integration, semantic understanding, and security to create a holistic and highly adaptive system for managing agents in complex and dynamic setup in the background context of the IEC 61499.

These innovations collectively enable more efficient and autonomous operations in various applications, particularly manufacturing and intra-logistics with application in Packaging Production Processes typically characterized by Environments with Critical Conditions of work (e.g. Low Temperature, Humidity, Wet Floors, confined and restricted Working Spaces) and, for these reasons, mainly based on Human Operators effort.

In conclusion, the innovation and advancements are:

Dynamic and Autonomous Recognition:

<u>Innovation</u>: The deliverable introduces a framework capable of autonomously recognizing and identifying devices and agents as they are added or removed from the setup.

<u>Advancement</u>: Traditional setups often rely on manual configurations or predefined rules. In Zero-SWARM platform, autonomous recognition is a significant leap toward fully adaptive systems.

Real-time Reconfiguration Engine:

<u>Innovation</u>: The development of a real-time reconfiguration engine is a novel approach.

<u>Advancement</u>: The reconfiguration engine can quickly adapt control logic, update connections between function blocks, and allocate resources based on the dynamically changing configuration. This agility is a notable departure from static or slow-reacting systems.

Dynamic Resource Allocation:

<u>Innovation</u>: Zero-SWARM focus include dynamic resource allocation as an integral part of reconfiguration.

<u>Advancement</u>: Traditional setups often allocate resources statically or with limited adaptability. This innovation enables efficient resource allocation to support the evolving configuration.

Event-driven Adaptation:

<u>Innovation:</u> The emphasis on event-driven mechanisms for communication and adaptation is forward-looking.

Advancement: This approach allows for real-time responsiveness to changing conditions, which is crucial in dynamic environments. Traditional systems may lack this level of adaptability.

5G Connectivity as an Enabler:

Innovation: Leveraging 5G technology as a key enabler for real-time communication and coordination among agents.

Advancement: The use of 5G, with its high-speed, low-latency capabilities, represents a significant advancement in facilitating rapid data exchange and decision-making, especially in scenarios like robot controllers for production lines or dynamic intra-logistic layouts.

Semantic Understanding for Reconfiguration:

Innovation: Incorporating semantic data for improved understanding and decision-making.

Advancement: By considering the meaning and context of data, the system can make more intelligent and context-aware reconfiguration decisions, setting it apart from rule-based or data-driven approaches without semantic understanding.

Autonomous Adaptation in Complex Environments:

Innovation: The ability to autonomously adapt in complex and dynamic environments is a key innovation.

Advancement: Traditional systems often struggle to handle highly dynamic situations. This project's focus on complexity sets it apart from simpler, more static control systems.

Security and Agent Identification:

Innovation: Developing secure mechanisms for agent recognition and identification.

Advancement: Ensuring auto-recognition security is paramount, especially as the number of connected devices and agents grows. This project addresses this concern proactively.

References

[Ref 4.5.1.1] Myllyviita, M. (2022). Multi-platform integration approach for IEC61499 and Robot Operating System.

[Ref 4.5.1.2] ROS2 Real-Time-Programming. (2023). https://docs.ros.org/en/iron/Tutorials/Demos/Real-Time-Programming.html.

[Ref. 4.6.1]. Norfolk, MA, USA, Sep. 2020. [Online]

https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/Digital-Twin-and-Asset-Administration-ShellConcepts.pdf?_blob=publicationFile&v=7

[Ref 4.6.2] YE, Xun, et al. An industry 4.0 asset administration shell-enabled digital solution for robot-based manufacturing systems. IEEE Access, 2021, vol. 9, p. 154448-154459.

[Ref 4.6.3] B. Lv, Y. Zhang and W. Dai, "Integrated Asset Management Model based on IEC 61499 and Administration Shell," IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society, Toronto, ON, Canada, 2021, pp. 1-6, Doi: 10.1109/IECON48115.2021.9589597.

[Ref 4.6.4] B. Lv, Y. Zhang and W. Dai, "Integrated Asset Management Model based on IEC 61499 and Administration Shell," IECON 2021 – 47th Annual Conference of the IEEE Industrial Electronics Society, Toronto, ON, Canada, 2021, pp. 1-6, Doi: 10.1109/IECON48115.2021.9589597