A Multi-Agent Distributed Architecture for Autonomous, Resilient and Reliable Industrial CPS

Jorge Alves*, Pedro Sousa†, Tiago Cruz‡, and Jérôme Mendes§

*University of Coimbra, Institute of Systems and Robotics, ARISE, Pólo II, PT-3030-290 Coimbra, Portugal

† Oncontrol Technologies, Rua Cidade Poitiers, nº 155 – 1º Andar, PT-3000-108 Coimbra, Portugal

‡ University of Coimbra, CISUC, Department of Informatics Engineering Pólo II, PT-3030-290 Coimbra, Portugal

§ University of Coimbra, CEMMPRE, ARISE, Department of Mechanical Engineering, Pólo II, PT-3030-290 Coimbra, Portugal

jorge.alves@isr.uc.pt, pedro.sousa@oncontrol-tech.com, tjcruz@dei.uc.pt, jerome.mendes@uc.pt

***Abstract*—This paper presents a multi-agent architectural proposal that defines the granularity and interactions among elements to achieve fully distributed and autonomous orchestration of the system. The architecture reduces hierarchical dependence and completely avoids a single centralized global controller. This ongoing research demonstrates the adoption of modern distributed management methodologies in real industrial systems and highlights how the proposed architecture increases flexibility, resilience, and autonomy, thereby improving overall operational outcomes.**

***Index Terms*—Multi-Agent, Distributed Architecture, Industrial Cyber-Physical Systems.**

I. Introduction

Current industrial Cyber-Physical Systems (CPS) largely rely on rigid, centralized architectures derived from the traditional ISA-95 model [1]. They offer significant limitations when facing the increasing complexity, scalability demands, and dynamic requirements of modern industrial environments. While digitalization via Internet of Things (IoT) and Internet of Services (IoS) has improved connectivity and monitoring [2], architectures are only now evolving toward distributed paradigms where independent components coordinate without a single point of control to enhance flexibility and reliability [1], [2]. However, many existing connected solutions, such as distributed control systems, edge computing, multi-agent and service-oriented architectures, still utilize self-contained controllers and maintain hierarchical dependencies for centralized decision-making [1].

This paper proposes a modular, flexible, and standards-compliant distributed CPS architecture in development aligned with edge computing principles that integrates with existing industrial infrastructures and enables the autonomous execution of service-oriented industrial processes without centralized control. The architecture supports distributed management through mechanisms that provide autonomy and resilience to the system, thereby increasing its computational intelligence.

II. Architecture Proposal

The proposed architecture aligns with edge computing models while remaining compatible with traditional automation models, such as the ISA-95 pyramid. This ensures that the transition from the current high-hierarchical dependency and centralized architecture to a distributed architecture can occur progressively. The architecture explicitly adopts a service-oriented design and integrates a multi-agent system for decentralized management of industrial CPS, see Fig. 1. The architecture utilizes plug-and-play software instances that serve as digital interfaces and abstractions for individual physical or virtual resources, enabling seamless monitoring and control across the distributed system. Designed for portability, these instances support both ARM and x86 architectures, ensuring compatibility with embedded field devices in industrial environments. This modular approach facilitates a progressive transition to edge computing and digitization by allowing deployment on existing infrastructure, such as field and server nodes, with minimal restructuring.

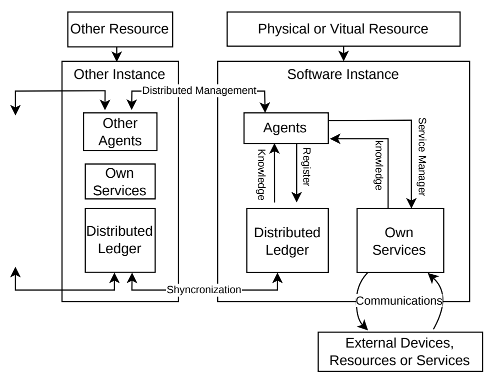

Fig. 1. Elements of distributed architecture.

Each software instance is composed of three core elements: Agents, Services, and Distributed Ledger. The agents constitute the agent space, responsible for automated and decentralized decision-making, and a service space, which provides

Jorge Alves is supported by Fundação para a Ciência e a Tecnologia (FCT) by the grant ref. 2023.00998.BDANA. Work partially sponsored by national funds through FCT under projects UID/00285/2025 (DOI: 10.54499/UID/00285/2025) and LA/P/0112/2020 (DOI: 10.54499/LA/P/0112/2020), and co-financed by the European Regional Development Fund through Innovation and Digital Transition Thematic Programme (COMPETE 2030) of Portugal 2030, Project WOOSU (COMPETE2030-FEDER-01463200).

modular functional capabilities for the industrial process. Co-operative agent interactions are backed by a blockchain-based distributed ledger that ensures data immutability, transparency, and trust through consensus mechanisms.

A. Agent Space

The agent space can be defined as the space where agents will compose cooperative control for service management in the distributed system. In the proposed holonic architecture

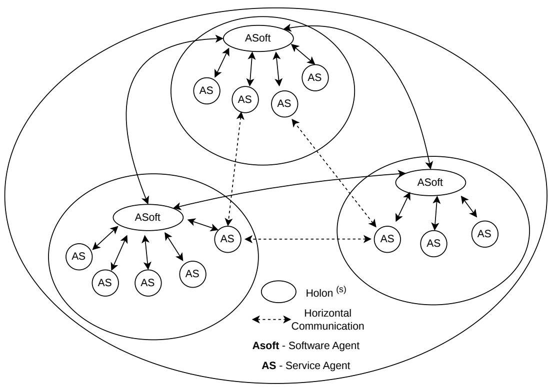

Fig. 2. Proposed holonic architecture of agent space.

(Fig. 2), the agent space relies on distributed autonomous management by dividing responsibilities between Software Agents (ASoft) and Service Agents (AS). The Software Agent, acting as a Mediator Holon, is responsible for high-level decentralized orchestration of services, managing resource allocation, security, and system health using knowledge gathered from telemetry and distributed ledgers. Complementing this, Service Agents provide granular modularity by executing the actual industrial control and managing information specific to functional packages within an instance. While these Service Agents primarily interact with the Mediator Holon to report local status and trigger reconfigurations of instance services, they also maintain peer-to-peer communication channels for rapid security and failure verifications, ensuring a resilient system without reliance on a centralized controller. According to the IEEE Std 2660.1™-2020 standard [3], the software agent is defined as on-device tightly coupled, and the service agents may also act as tightly coupled hybrid.

In this setup, interactions between software agents and service agents facilitate micro-level operations, while high-level distributed orchestration is driven by the software agent.

B. Distributed Architecture

Figure 3 presents an overview of the architecture in a single layer in a distributed environment, illustrating the relationships among software instances, agents, services, and devices with industrial devices or resources. Building upon the established agent roles, these instances can also be deployed across a multi-layer edge-to-cloud environment, utilizing the plant's existing Ethernet infrastructure. This deployment spans heterogeneous cyber-physical resources, ranging from constrained field devices to high-performance cloud servers. Consequently,

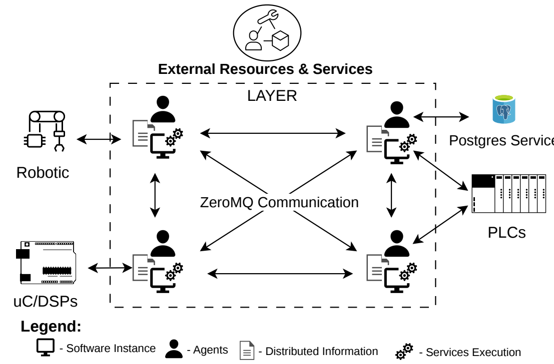

Fig. 3. Distributed & heterarchical multi-agent architecture at one layer.

the system establishes a hierarchical and heterarchical architecture: it is inherently hierarchical regarding hardware capabilities - with tight physical integration at the device/edge layers and high-level coordination in the cloud - but strictly heterarchical in communication. Because agents communicate directly peer-to-peer across layers without a central coordinator, the architecture maximizes robustness, scalability, and fault tolerance.

III. Conclusion and Final Analysis

The direct communication among agents drives the convergence of IoT and edge computing architectures. By pushing execution, data processing, and decision-making closer to the physical processes, the networking and application layers become seamlessly distributed across the edge paradigm. This evolution unifies computational capabilities across the system, blurring traditional layer boundaries. For instance, device-level units can now perform edge-processing tasks, albeit within their specific hardware constraints. Consequently, this unification further minimizes hierarchical dependencies, enabling dynamic service reallocation and reinforcing the system's overall adaptability and autonomous management. The distributed management is achieved via mechanisms for dynamic service allocation, failure detection and recovery, alongside trust and reputation assessment. Furthermore, leveraging computational intelligence, consensus algorithms, and distributed ledger technologies is crucial to enhancing the system's resilience and autonomy. In conclusion, these capabilities are critical for modern industrial environments, driving process optimization and ensuring reliable operational outcomes.

References

[1] J. Alves, P. Sousa, T. Cruz, J. Mendes, A review of architecture features for distributed and resilient industrial cyber–physical systems, Journal of Manufacturing Systems 82 (2025) 1069–1090. doi:10.1016/j.jmsy.2025.07.012.

[2] B. Dafflon, N. Moalla, Y. Ouzrout, The challenges, approaches, and used techniques of cps for manufacturing in industry 4.0: A literature review, The International Journal of Advanced Manufacturing Technology 113 (2021) 2395–2412.

[3] IEEE Recommended Practice for Industrial Agents: Integration of Software Agents and Low-Level Automation Functions, IEEE Standard IEEE Std 2660.1-2020, IEEE, New York, NY, USA (sep 2020).