Agents in the Computing Continuum: the MLSysOps Perspective

Marco Loaiza, Claudio Savaglio, Raffaele Gravina
University of Calabria, Italy
[m.loaiza, csavaglio, r.gravina]@dimes.unical.it

Dimitris Chatzopoulos
University College Dublin, Ireland
dimitris.chatzopoulos@ucd.ie

Spyros Lalis
University of Thessaly, Greece
lalis@uth.gr

Abstract

Multi-agent systems (MASs) have gained considerable attention in the field of distributed computing due to their ability to provide technical interoperability, resource sharing and flexible coordination. Consequently, MAS are well-suited to address the challenges posed by the distributed and heterogeneous nodes within the device-edge-cloud continuum, including orchestration and standardization, optimal resource allocation, micro service placement policies, security and privacy. The objective of this study is to introduce the MLSysOps project, which aims at the autonomous management of the entire continuum tackling some of the challenges mentioned before. MLSysOps utilizes a hierarchical agent-based AI architecture to interface with the underlying resource management and application deployment/orchestration mechanisms. A comparative analysis is conducted between the existing related work and the proposed framework, highlighting the peculiarities and advantages of our approach.

Categories and Subject Descriptors

H.4 [**Operating Systems**]: Miscellaneous; I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multi-agent systems*

Keywords

MLSysOps, multi agent systems, computing continuum, system management, device edge cloud computing

1 Introduction

Over the past decade, cloud computing has emerged as the dominant paradigm in the computing field. Its scalability, on-demand resource availability, cost-efficiency, and flexibility have positioned it as a driving force across various industries. Cloud computing fuels digital transformation and empowers businesses to innovate at a rapid pace. Moreover, the emergence of edge computing has extended intelligent decision-making to distributed environments, revolutionizing data processing and enabling real-time insights at the network's edge. The integration of Internet of Things (IoT) devices, edge technologies, and cloud computing has given rise to the concept of the Device-Edge-Cloud (DEC) continuum. However, the emergence of this promising paradigm has presented significant challenges in managing heterogeneous and distributed resources, making human management entirely unrealistic. To achieve dynamic and flexible management of systems and applications with minimal user involvement, rule-based policies have been employed, offering some improvement in management. Nevertheless, these policies come with several limitations due to the intrinsic heterogeneity, dynamism and context-dependence of continuum systems. In light of these challenges, Artificial Intelligence (AI) emerges as a promising alternative, leveraging MAS to innovate the management of distributed systems.

This article introduces a Platform-as-a-Service (PaaS) framework that provides automated and adaptive deployment and orchestration of distributed applications, being developed in the context of the EU funded project named MLSysOps [11]. These applications consist of interacting container-based components with specific resource and quality-of-service (QoS) requirements. To highlight the strengths and benefits of our AI-driven approach, we compare MLSysOps with recent works that also utilize MASs for resources management and orchestration in the continuum.

The remainder of the paper is structured as follows: Section **2** describes the background, providing an introduction to the key features of the technologies and paradigm involved. Section **3** presents the motivations and discusses related work, summarizing recent papers exploiting agent-based solutions in the DEC continuum. Section **4** describes the MLSysOps project, presenting its vision and objectives of the proposed multi-agent framework in subsection **4.1**, and conducting a comparison between MLSysOps and the aforementioned surveyed articles in subsection **4.2**. Final remarks conclude the paper.

2 Background

This section introduces key concepts related to the MLSysOps project: SysOps, Computing Continuum, and software agents.

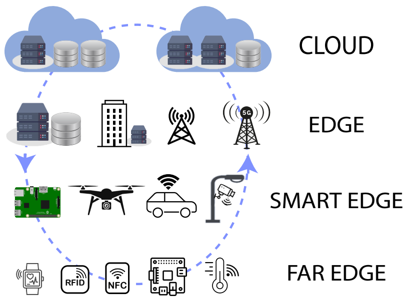


Figure 1: Device-Edge-Cloud continuum.

2.1 SysOps

IT systems encompass the combination of hardware, software, networks, and data to meet the information technology requirements of an organization. These systems are specifically designed to efficiently store, process, transmit, and manage data and information. Systems Operations (Sysops) is the discipline of managing and upholding the operational aspects of an IT system or infrastructure. It involves supervising the configuration, deployment, and monitoring of various components and services within a system to ensure its seamless operation and optimal performance. In most cases, these tasks are automatically carried-out through static rule-based policies or manually handled based on human expertise. Sysops professionals, indeed, are responsible for tasks such as provisioning systems, installing and configuring hardware and software elements, managing user accounts and permissions, monitoring system health and performance, implementing security measures, executing backups and disaster recovery procedures, and addressing technical issues or incidents that may arise.

In light of the substantial growth in such ecosystem, various approaches leveraging AI-related solutions have been devised to advance the current approach and efficiently address the Sysops and its inherent complexities. Among these, MLSysOps is proposed as a prominent candidate.

2.2 Computing Continuum

The DEC continuum is a computing paradigm that integrates processing, storage and network resources across a multi-layer hierarchy composed of IoT devices, edge clusters, and cloud data centers. Also known as *Computing Continuum*, its goal is to create a fluid and flexible ecosystem where distributed resources and services can be dynamically combined to support data-driven applications [2]. Needless to say that Sysops in DEC continuum involves further complexity.

This continuum, indeed, encompasses diverse layers of collaborating computing infrastructure needed to be managed for supporting a wide range of applications. Each layer utilizes unique technologies and architectures tailored to meet the specific demands of enterprise applications. Through collaboration, these layers form a seamless continuum, enabling efficient data processing from IoT devices to powerful systems in the cloud as can be seen in Figure1. At the top of the continuum is cloud computing, which provides a model for convenient, on-demand remote access to a shared pool of configurable computing resources. It offers a highly scalable and flexible infrastructure for storing, processing, and analyzing data using one or multiple cloud platforms. Despite the widespread adoption of this computing paradigm, it has certain limitations when it comes to supporting real-time, low-latency applications, as well as concerns about security, privacy, and data protection. To address these limitations, the proposed solution lies in edge computing, which is a distributed computing paradigm that focuses on processing data and running applications closer to where the data is generated. Instead of sending all data to a remote centralized data center in the cloud, this approach emphasizes on local data processing, resulting in reduced latency and data traffic over wide area networks and the Internet.

Edge computing can also be divided into different types of layers, depending on the capabilities of the devices and the functions they perform in data-driven applications. The term “Edge” refers to the data center located between the cloud data center and the data generators, which consist of local server nodes on-premises. These nodes are tightly coupled or clustered, ensuring fast network connectivity among them. Edge data centers are physically closer to the nodes that produce data. Examples include server clusters in buildings, such as hospitals, or special purpose infrastructure like 5G stations, which are also considered edge data centers. On the other hand, smart edge nodes are powerful nodes at the edge capable of mobility or static deployment. They have proper operating systems and internet connectivity. Examples of smart edge nodes include those that utilize single-board computers such as Raspberry Pi, Coral Edge, Beagle-Board, and others. Far edge nodes, in contrast, are resource-constrained nodes that often lack of a proper operating system and Internet connectivity. These devices communicate wirelessly with an edge node that acts as a gateway for them. Far edge nodes can be organized into tightly coupled groups, similar to a server node cluster, but with significantly less computing and communication resources. They generally consist of wearable devices, smart sensors, or smart devices that use micro-controller or systems-on-chips.

The DEC continuum paradigm envision the exploitation of the strengths of cloud computing, edge computing, and IoT devices to create an ecosystem where data-driven applications can thrive. By combining their strengths, this continuum enables efficient and scalable data processing, ultimately enhancing the capabilities and user experience of modern applications.

2.3 Software Agents

There is no universally accepted definition of an agent, but it can be described as an autonomous software entity interacting with other systems and the environment to achieve specific goals [17]. Indeed, agents possess the ability to perceive their surroundings, make decisions, and take actions in pursuit of their objectives. They exhibit characteristics such as autonomy, rationality, responsiveness, mobility, reactivity, proactivity, and social abilities [18, 19]. Agents are capable of encapsulating complex functionalities and abstracting heterogeneous resources. They act as facilitators for interoperability and support the development of complex, cooperative, and adaptive distributed systems.

The Foundation for Intelligent Physical Agents (FIPA) [5, 15] is an international organization that focuses on promoting and standardizing the use of intelligent agent technology. FIPA provides a platform for researchers, developers, and practitioners to collaborate, share knowledge, and establish standards for agent-based systems. The organization develops specifications, standards, and methodologies to ensure interoperability and compatibility among different agent platforms. FIPA provides the FIPA-ACL, a standard format and protocol for agents to communicate and exchange information. This capability enables the creation of MAS consisting of interconnected agents that interact with each other to cooperate and solve common tasks through a unified environment, making them suitable for distributed systems. MAS have roles and interaction rules that regulate the relationship between entities or entities with the environment.

In the context of continuum computing, there are challenges in effectively processing and managing the vast amount of data generated by IoT devices, making intelligent real-time decisions, and seamlessly integrating with data center resources (at the edge or in the cloud) for additional computational capabilities and storage. Traditional centralized approaches may face limitations in terms of latency, bandwidth, and scalability when dealing with these challenges. This is where MAS come into play in the continuum. Rooted in AI and distributed computing, MAS provide a framework for orchestrating and coordinating intelligent behavior among autonomous entities. In the DEC continuum, agents can be deployed across different nodes of the system, such as IoT devices, edge servers, and multiple cloud servers. By leveraging the collective intelligence and cooperation of these agents, MAS enable adaptive and scalable architectures that effectively handle the challenges of the continuum. Agents can collaborate and share information to collectively analyze and process data at the edge, enabling localized real-time decision-making. They can also interact with agents deployed in the cloud to offload computation-intensive tasks or access additional resources when needed.

3 Motivations and Related Work

Due to the vast number of devices that comprise edge cloud computing, there are several identified challenges stemming from the scale, heterogeneity, high dynamism, and intrinsic local properties/variability of the continuum. The most common challenges encountered in surveys related to edge-cloud computing [9, 4, 7, 1] consistently include concerns such as optimal resource allocation, security and privacy, orchestration and standardization, and micro service placement policies. In recent years, some of these challenges have been addressed through approaches that prioritize the use of agents for service orchestration and resource management in the DEC continuum.

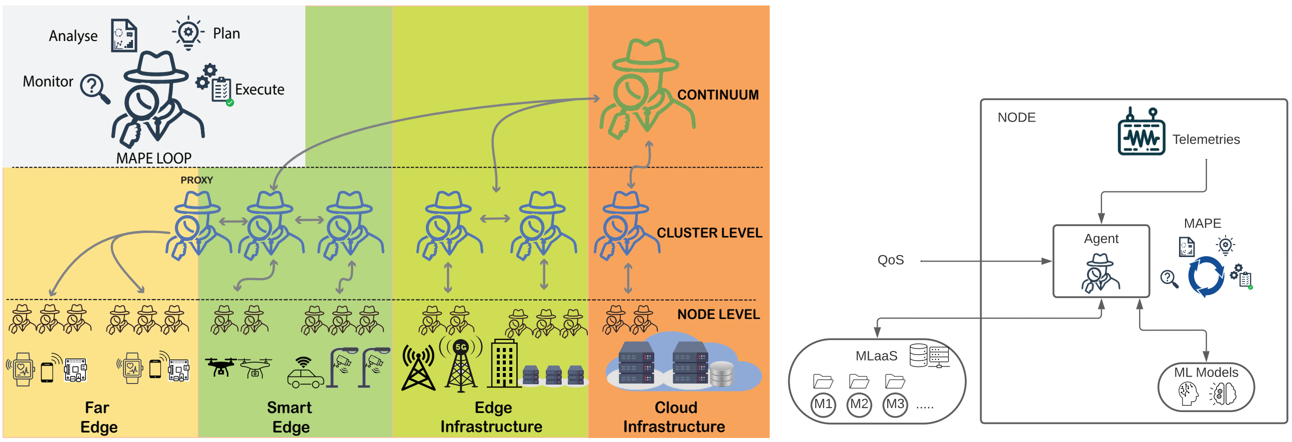
For example, Chima et al. [3] introduced a novel approach to scheduling edge-native applications on 5G networks by considering contextual information. It addresses the limitations of existing scheduling techniques for Industrial IoT (IIoT) applications. The proposed solution adopts a client-server model, where a central monitor functions as the client and requests real-time data from monitor agents acting as servers. These agents are responsible for monitoring individual edge devices within the distributed system and reporting their runtime data to the edge-cloud network management system. By incorporating contextual information and utilizing the CoAP protocol for communication, the authors introduce a framework that addresses the limitations of existing scheduling techniques and enables efficient data gathering and scheduling of edge-native applications. The agents in the system appear to be implemented in Python, utilizing the "async" library to facilitate communication among themselves through asynchronous function facilities.

In their work, Morkevicius et al. [12] proposed a two-stage multi-objective optimization method for dynamic service orchestration in fog computing. Their method focuses on optimizing service placement among available fog nodes while considering QoS requirements and minimizing resource usage. They introduce orchestrating agents, specialized services located in each fog node, to communicate with a central orchestrator. These agents monitor local hardware and software environments, collecting data to share with the cloud-based orchestrator. Based on the orchestrator's decisions, the agents initiate the necessary actions on the services. The architecture consists of three main components: the fog orchestrator, fog node acting as a fog orchestrator agent (FOA), and end devices. The FOAs are responsible for managing local resources within specific fog nodes and play a crucial role in local resource management. The proposed method is implemented in simulation using Matlab.

Fu et al. [6] proposed Nautilus, a run-time system designed for deploying user-facing services based on micro services in the edge-to-cloud continuum. Their work focuses on using reinforcement learning-based resource management agents to optimize micro service deployment. These agents observe resource usage on each node, such as CPU cores, memory capacity, and network bandwidth, and take actions to optimize CPU usage, memory quota, and network bandwidth for each micro service. The agents employ deep reinforcement learning and consider monetary costs based on cloud service pricing models for cost optimization while maintaining QoS and throughput targets.

Masip et al. [10] proposed a hierarchical architecture for resource management in the cloud continuum. Their architecture includes agents, cloud agents, and micro agents that handle devices with different capacities. The proposed agent consists of a Platform Manager, an Agent Controller, APIs, security, and Data Management blocks. These agents play a crucial role in resource management at the edge and fog levels, while also facilitating communication with the cloud. Through collaborative efforts, they ensure coordinated and efficient monitoring and management of the entire resource set, spanning from the edge to the cloud. The authors have implemented their own agents and made them available as downloadable docker containers on the project web page.

EPOS Fog, a decentralized multi-agent system designed for load-balancing in fog computing, is presented by Nezami et al. [13]. This system aims to distribute the workload efficiently and reduce the costs associated with executing services in IoT service placement. EPOS Fog utilizes a self-



(a) MLSysOps Infrastructure

(b) MLSysOps agent operation

Figure 2: MLSysOps target infrastructure, the MAS hierarchical organization and the ML-driven agent-based approach.

organized tree topology with decentralized agents in a multi-agent system implemented in java. The agents generate and select placement plans to optimize edge utilization and service execution cost. Through collective decision-making and adaptive choices, the agents learn and optimize their objectives. The global service placement plan is achieved by aggregating the selected plans from each agent.

Liutkevivcius et al. [8] propose a distributed agent-based orchestrator model for fog computing, which is implemented using the JADE middleware. In this model, software agents are deployed on each fog node. These agents, including the Synchronization Agent, Decision Making Agent, Resource Monitoring Agent, Request Processing Agent, and Deployment Agent, collaborate to synchronize clocks, make service placement decisions, monitor resources, process service requests, and deploy services. The agents communicate with each other to distribute the decision-making process among multiple orchestrator fog nodes, enhancing the scalability and resilience of the system.

Finally, Yu et al. [20] proposed an adaptive and efficient function delivery engine, named "FaaSDeliver", to optimize the cost of running functions in a heterogeneous computing continuum that includes the cloud, fog, and the edge. They use a monitor agent within the FaaS platform that reports the execution log to the online optimizer in FaaSDeliver framework that creates a cost-efficient function delivery policy for each function, including the FaaS platform selection and resource allocation.

In the following section we present the MLSysOps project by introducing its fundamental concepts and by comparing its key features (i.e., a hierarchical AI architecture based on MAS and proposes re trainable ML models to enhance autonomic system operation in the DEC continuum) with respect to these related works (precisely, see Table 1).

4 The MLSysOps Project

The MLSysOps project [11] aims to tackle the challenges of managing resources and application deployment on DEC continuum systems via machine learning. It is intended as a PaaS which enables the implementation, orchestration, execution, and adaptation of distributed applications composed of interacting components with specific resource requirements and considerations for QoS. Besides application management, MLSysOps also handles system infrastructure administration to achieve robust, reliable, energy-efficient, and environmentally friendly operation. It uses ML methods to support these functionalities and autonomously control the underlying deployment, orchestration and resource management mechanisms. A key aspect of MLSysOps is its ability to leverage the available system resources to support continuous and explainable ML models while executing the application on the system.

The main objective of MLSysOps is to design, implement, and evaluate a comprehensive AI-driven framework for end-to-end system management throughout the DEC continuum. It aims to tackle the challenges related to managing and optimizing performance, reliability, and energy efficiency in continuum systems. More specifically, the main objectives set by MLSysOps are:

1. Deliver an open AI-ready, agent-based framework for holistic, trustworthy, scalable, and adaptive system operation across the heterogeneous DEC continuum.
2. Develop an AI architecture supporting explainable, efficiently re-trainable ML models for end-to-end autonomic system operation in the DEC continuum.
3. Enable efficient, flexible, and isolated execution across the heterogeneous continuum.
4. Support green, resource-efficient, and trustworthy system operation, while satisfying application QoS/QoE requirements.
5. Execute realistic model training, validation, and evaluation.

Shifting our focus to the subject of this paper, in the following subsection, we present the various agents employed in the proposed framework, highlighting their functionalities and making a comparison between this implementation and the related work previously presented. We examine the de-

Table 1: Agent-based Sysops approaches in the computing continuum.

Year	Paper Title	Agent Based Task and mobility	Continuum	Approach to Sysops	Use cases	Agent Technology
2020	Context-Aware Kubernetes Scheduler for Edge-native Applications on 5G [3]	Monitor Fixed	Edge	Scheduler algorithm	Industrial IoT Car 3D model	Python
2021	Method for Dynamic Service Orchestration in Fog Computing [12]	Monitor, Execute (manage) Fixed	Fog	Multi-objective optimization	Simulation with Matlab	Matlab
2021	Adaptive Resource Efficient Micro service Deployment in Cloud-Edge Continuum [6]	Monitor, Analyze (prediction) Fixed can be mobile	Edge to Cloud	Machine Learning based approach	Social Network, Media Service, Hotel Reservation and Ticket train	not specified
2021	Managing the Cloud Continuum: Lessons Learnt from a Real Fog-to-Cloud Deployment [10]	Monitor, Execute Fixed	Fog to cloud	Scheduler algorithm executed by agents	Smart cities, Smart Fog-Hub Service, Smart boat service management	Java
2021	Decentralized Edge-to-Cloud Load Balancing: Service Placement for the Internet of Things [13]	Monitor, Plan, Execute Fixed	Edge fog cloud	Optimization approach	Health monitoring systems	Docker
2022	Distributed Agent-Based Orchestrator Model for Fog Computing [8]	Monitor, Analyze (Decision Making), Execute Fixed	Fog	Policy based approach	Person-oriented applications, smart home and environments monitoring	JADE
2023	FaaSDeliver: Cost-efficient and QoS-aware Function Delivery in Computing Continuum [20]	Monitor Fixed	Edge-fog-cloud	Machine Learning based approach	Healthcare	not specified
2023	MLSysOps Framework (this work)	Monitor Analyse Plan Execute Fixed and mobile	DEC	Re-trainable Machine Learning based approach	Smart Cities and Smart Agriculture	SPADE

sign and implementation aspects of the MLSysOps agent-based framework, exploring the distinct roles and capabilities of each agent, as well as the proposed infrastructure and agent operations, as depicted in Figures 2a and 2b respectively. Our aim is to provide a comprehensive understanding of the structure, functionality, and unique benefits of the agent-based framework in contrast to the existing literature.

4.1 Proposed agent framework

As stated in the project objectives, MLSysOps introduces a distributed MAS to address the various challenges in system operations tasks within the DEC continuum. In particular, each agent has the capability to perform the MAPE (Monitor-Analyze-Plan-Execute) service [16] through each layer of the continuum to provide its core functionality. In other words, each agent has the ability to: **(i) Monitor** the collected data that will be used for both system monitoring and ML model re training; **(ii) Analyse** the data, supervise model retraining, detect model drifting, execute non-ML benchmark algorithms, collaborate with the Plan service for executing ML models, and performs conflict resolution among concurrent models; **(iii) Plan**, in addition to executing the models and in synergy with Execute service, the AI-ready resource provisioning, application deployment and orchestration, and security and trust management mechanisms; and **(iv) Execute** the plans generated by the Plan service.

The MLSysOps project uses a multi-layered approach to enable interoperability at varying levels of abstraction, considering both syntax and semantics of requirement specifications, events notification, and telemetry metrics in the heterogeneous DEC environment. As shown in Figure 2a, the layers allow to categorize the resources based on their infrastructure capabilities in four main layers: far edge, smart edge, edge data centers, and cloud data centers. Different types of agents are assigned the responsibility of AI-driven control and ML model training across different levels of the hierarchy, each with varying granularity of telemetry data and requirements. As depicted in the same figure, Node agents are represented in black, Cluster agents in blue, and the Continuum agent in green.

Node agents: These agents are responsible for the management of ML model training and evolution processes at the node level.

Cluster agents: These agents are responsible for AI-driven control and ML model training for entire groups of nodes, such as nodes that are part of the same edge infrastructure in a given geographical area, or server nodes that are part of the same cloud-edge data center.

Continuum agent: This singleton agent facilitates the interaction between cluster agents to coordinate higher-level AI-driven control and ML training purposes within a layer (e.g., Cluster agents of different cloud data centers) or between layers (e.g., Cluster agents of different edge infrastructures and Cluster agents of cloud data centers).

It is worth noting that the node agents are executed directly in the respective nodes of the system except for those devices where the agent resides at the proxy level, taking into account that the devices located in the far edge layer do not have the capabilities to execute the agent due to the resource or communication limitations. In contrast, Cluster agents and the Continuum agent can be hosted anywhere, given their ability to interact between layers and available system resources.

By incorporating these functionalities, MLSysOps enhances its capabilities in monitoring, analyse, planning, and executing tasks within the continuum, promoting efficient and effective system management and orchestration. The expectation is to achieve scalable and incrementally refined AI-driven control across the heterogeneous DEC continuum.

4.2 Comparison with related works

The comparison between the proposed MLSysOps and related works, discussed in Section **3**, is presented in Table 1. The analysis focuses on the role of agents within the system, where it is observed that agents primarily serve a monitoring function. Agents in the discussed studies are adept at monitoring, collecting, and sharing information within the ecosystem usually being fixed on the nodes they are working on. Some studies also highlight agents ability to perform diverse tasks such as system management, orchestration, and applying ML or optimization techniques to improve system performance. The MLSysOps framework encompasses these tasks on each agent within the MAPE loop, continually assessing sysops and enhancing energy efficiency by distributing workloads across available resources in the continuum.

Unlike certain approaches that concentrate on specific layers of the continuum, MLSysOps provides an end-to-end solution including far edge devices, ensuring seamless execution of micro services throughout the entire DEC continuum. The approach towards sysops is another point of comparison, addressing whether system management is handled through human intervention, policy-based automation, or ML techniques for continuous monitoring and maintenance. MLSysOps stands out by utilizing explainable ML approaches, aiming to understand the decision-making process and provide explanations for system operations within the extensive ecosystem. It strives to manage the continuum extensively without relying on human intervention, by employing automatic re trainable ML models according to current telemetries and expected performance (see Figure 2b). Notably, ML models are stored outside the MLSysOps agent and can be invoked as microservices: such a decoupling provides higher degree of flexibility (agents are not monolith but lightweight entities free to use always updated/customized models) and it is another relevant distinctive feature of our project. For the implementation of MLSysOps agents, we are considering a MAS platform that is based on Python, such as SPADE [14]. The use of Python opens up the possibility of leveraging widely employed machine learning libraries like Scikit-learn, TensorFlow, and PyTorch. Additionally, compliance with FIPA standards in the framework permits communication among various heterogeneous MAS, thereby enhancing their interoperability and integration. The adherence to such a standard is key given that agents could be implemented in different languages/platforms depending on the resource of their host environment.

Overall, MLSysOps offers a comprehensive framework that utilizes MAS, monitoring, automation, and optimization techniques to impulse an AI-driven control across the heterogeneous DEC continuum. The effectiveness and versatility of the MLSysOps framework will be evaluated through both simulated use cases and real testbeds in the DEC continuum in the domain of Smart Cities and Smart Agriculture.

5 Conclusions

At the state-of-the-art there exist a number of studies implementing agents or MAS to enhance resource management and application deployment in the DEC continuum. However, many of these works still depend on policies or automation algorithms, which prove to be inefficient in the face of a large, heterogeneous ecosystem that undergoes dynamic changes. This inefficiency is particularly evident when considering system infrastructure changes due to the intrinsic local properties and variability of the DEC Continuum.

To overcome these limitations, this paper has presented an approach that utilizes ML techniques implemented through MAS to enhance resource management and application deployment in the DEC continuum. Specifically, the proposed MLSysOps framework is based on the MAPE loop model and continuously retrains ML models to adapt and enhance functionalities in response to system changes. In the next three years the project will implement the proposed infrastructure and will carry out numerous tests using a combination of research testbeds and real world applications to showcase and validate the discussed functionalities of the proposed framework.

Acknowledgments

The research leading to this work was carried out under the Italian MIUR, PRIN 2017 Project "Fluidware" (CUP H24I17000070001), and under the "MLSysOps" Project (Grant Agreement 101092912) funded by the European Community's Horizon Europe Programme. For the purpose of open access, the authors have applied a CC BY public copyright license to any Author Accepted Manuscript version arising from this submission.

6 References

[1] V. Barbuto, C. Savaglio, M. Chen, and G. Fortino. Disclosing edge intelligence: A systematic meta-survey. *Big Data and Cognitive Computing*, 7(1):44, 2023.

[2] P. Beckman, J. Dongarra, N. Ferrier, G. Fox, T. Moore, D. Reed, and M. Beck. *Harnessing the Computing Continuum for Programming Our World*. John Wiley & Sons, Ltd, New York, 2020.

[3] M. Chima Ogbuachi, A. Reale, P. Suskovics, and B. Kovács. Context-aware kubernetes scheduler for edge-native applications on 5g. *J. of communications software and systems*, 16(1):85–94, 2020.

[4] B. Costa, J. Bachiega Jr, L. R. de Carvalho, and A. P. Araujo. Orchestration in fog computing: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 55(2):1–34, 2022.

[5] FIPA Foundation. http://www.fipa.org (last accessed July 2023).

[6] K. Fu, W. Zhang, Q. Chen, D. Zeng, and M. Guo. Adaptive resource efficient microservice deployment in cloud-edge continuum. *IEEE Trans. on Parallel and Distributed Systems*, 33(8):1825–1840, 2021.

[7] T. Kim, S. Oh, I. Cha, S. Lee, H. Ko, S. Pack, and J. Han. Holistic orchestration for edge-native applications: A review. In *2022 13th Intl. Conf. on Information and Communication Technology Convergence (ICTC)*, pages 771–774. IEEE, 2022.

[8] A. Liutkevičius, N. Morkevičius, A. Venčkauskas, and J. Toldinas. Distributed agent-based orchestrator model for fog computing. *Sensors*, 22(15):5894, 2022.

[9] R. Mahmud, K. Ramamohanarao, and R. Buyya. Application management in fog computing environments: A taxonomy, review and future directions. *ACM Computing Surveys (CSUR)*, 53(4):1–43, 2020.

[10] X. Masip-Bruin, E. Marín-Tordera, and e. a. Sánchez-López. Managing the cloud continuum: Lessons learnt from a real fog-to-cloud deployment. *Sensors*, 21(9):2974, 2021.

[11] MLSysOps Project. https://mlsysops.eu/, (accessed July 2023).

[12] N. Morkevicius, A. Venčkauskas, N. Šatkauskas, and J. Toldinas. Method for dynamic service orchestration in fog computing. *Electronics*, 10(15):1796, 2021.

[13] Z. Nezami, K. Zamanifar, K. Djemame, and E. Pournaras. Decentralized edge-to-cloud load balancing: Service placement for the internet of things. *IEEE Access*, 9:64983–65000, 2021.

[14] J. Palanca, A. Terrasa, V. Julian, and C. Carrascosa. Spade 3: Supporting the new generation of multi-agent systems. *IEEE Access*, 8:182537–182549, 2020.

[15] S. Poslad. Specifying protocols for multi-agent systems interaction. *ACM Trans. Auton. Adapt. Syst.*, 2(4):15–es, nov 2007.

[16] C. Savaglio and G. Fortino. Autonomic and cognitive architectures for the internet of things. In *8th Int. Conf. on Internet and Distributed Computing Systems (IDCS 2015)*, pages 39–47. Springer, 2015.

[17] C. Savaglio, M. Ganzha, M. Paprzycki, C. Bădică, M. Ivanović, and G. Fortino. Agent-based internet of things: State-of-the-art and research challenges. *Future Generation Computer Systems*, 102:1038–1053, 2020.

[18] Y. Shoham and K. Leyton-Brown. *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press, 2008.

[19] M. Wooldridge. *An introduction to multiagent systems*. John Wiley and Sons, 2009.

[20] G. Yu, P. Chen, Z. Zheng, J. Zhang, X. Li, and Z. He. Faasdeliver: Cost-efficient and qos-aware function delivery in computing continuum. *IEEE Trans. on Services Computing*, 2023.