

Received 4 April 2024, accepted 9 May 2024, date of publication 29 May 2024, date of current version 13 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3406861



A Framework for Cognitive, Decentralized Container Orchestration

RUTE C. SOFIA¹, (Senior Member, IEEE), JOSH SALOMON²,
SIMONE FERLIN-REITER², (Member, IEEE), LUIS GARCÉS-ERICE³, (Member, IEEE),
PETER URBANETZ³, HARALD MUELLER⁴, RIZKALLAH TOUMA⁵, ALEJANDRO ESPINOSA⁵,
LUIS M. CONTRERAS⁶, VASILEIOS THEODOROU⁷, NIKOS PSAROMANOLAKIS⁷,
LEFTERIS MAMATAS⁸, (Member, IEEE), VASSILIS TSAOUSSIDIS⁹,
XIAOMING FU¹⁰, (Fellow, IEEE), TINGTING YUAN¹⁰, ALBERTO DEL RIO¹¹,
DAVID JIMÉNEZ¹¹, (Member, IEEE), ANDRIES STAM¹², EFTERPI PARASKEVOULAKOU¹³,
PANAGIOTIS KARAMOLEGKOS¹³, VITOR VIEIRA¹⁴, JOSEP MARTRAT¹⁵,
IGNACIO MARISCAL PRUSIEL¹⁵, DORINE MATZAKOU¹⁶, JOHN SOLDATOS¹⁶,
DAVID REMON¹⁷, AND MARCO JAHN¹⁷

¹fortiss GmbH, 80805 Munich, Germany

²RedHat, 4350109 Ra'anana, Israel

³IBM Research Europe, 8803 Zürich, Switzerland

⁴Siemens AG, 80333 Munich, Germany

⁵i2CAT Foundation, 08034 Barcelona, Spain

⁶Telefónica, 28050 Madrid, Spain

⁷Intracom Telecom, 19002 Attica, Greece

⁸Athena Research Center, University of Macedonia, 546 36 Thessaloniki, Greece

⁹Athena Research Center, Democritus University of Thrace, 671 00 Xanthi, Greece

¹⁰Institute of Computer Science, University of Göttingen, 37073 Göttingen, Germany

¹¹Electrónica Física, Ingeniería Eléctrica y Física Aplicada, Universidad Politécnica de Madrid, 28040 Madrid, Spain

¹²Almende, 3013 AK Rotterdam, The Netherlands

¹³Data & Cloud Research Group, University of Piraeus Research Centre, 18534 Piraeus, Greece

¹⁴INOVA+, 4450-309 Matosinhos, Portugal

¹⁵ATOS, 28037 Madrid, Spain

¹⁶Netcompany-Intrasoft, Luxembourg City 1253, Luxembourg

¹⁷Eclipse Foundation, 64295 Darmstadt, Germany

Corresponding author: Rute C. Sofia (sofia@fortiss.org)

This work was supported in part by European Commission in the Context of the Horizon Europe Cognitive Decentralised Edge-Cloud Orchestration (CODECO) Project 101092696; and in part by the Swiss State Secretariat for Education, Research and Innovation (SERI) under Contract 23.00028.

ABSTRACT This perspective paper introduces a novel framework for container orchestration called CODECO. The CODECO orchestration framework relies on a data-network-computing approach to define the best infrastructure that can support the operation of next-generation Internet applications across a mobile, heterogeneous Edge-Cloud continuum. The selection of such an infrastructure is aligned with target performance profiles defined by the user, such as resilience or greenness. CODECO proposes to rely on decentralized Artificial Intelligence approaches to provide the most suitable infrastructure to an application deployment, considering infrastructural challenges, such as intermittent connectivity and node failure. This paper explains the current CODECO framework and gives insight into operational use-cases where CODECO is being deployed, as relevant examples of application for such a framework. Recent developments in the creation of the open-source CODECO framework are described and explained, allowing the use of the framework by the research community. The paper then provides a thorough analysis of CODECO's features in comparison with existing orchestration frameworks, explaining the benefits introduced with this dynamic orchestration approach.

INDEX TERMS Edge-cloud, orchestration, Kubernetes, AI/ML, heterogeneous networks, data observability.

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott¹.

I. INTRODUCTION

Next generation Internet applications, particularly those related to the *Internet of Things (IoT)*, are increasingly adopting micro-service architectures. These applications consist of multiple containerized micro-services, often based on Docker,¹ allowing them to be deployed across the *Edge-Cloud Continuum*. This deployment approach provides new opportunities to explore services with reduced latency and optimized energy consumption.

With the increased use of Edge computing [1], [2], approximately 40% of IoT data is already being captured, stored, processed and partially analyzed at the Edge in various vertical domains such as Manufacturing, Energy, and Health. Processing data at the Edge requires a new design paradigm for the Edge-Cloud continuum to address service decentralization, mobility, large-scale dense environments, and multi-tenant support across heterogeneous technologies. Initiatives such as Gaia-X² [3] and the *International Data Spaces Association (IDSA)*³ [4] address the abstraction and interconnection of different data spaces at a service level to shape the design of the Edge-Cloud continuum.

However, flexible and intelligent cross-layer adaptation of the entire infrastructure is essential for supporting the next generation of IoT smart services. This adaptation requires examining the infrastructure from a computing, network, and data observability perspective. Today, such infrastructure is addressed only from a computing perspective. A cross-layer approach that can adapt the entire data-computing-network infrastructure to the needs of applications and users, is essential for handling large volumes of data across a mobile and heterogeneous edge-cloud continuum. Data issues, such as compliance with local and regional regulations, further complicate the picture. To effectively use the exchanged data, different domains require a higher degree of articulation, which requires a secure and interoperable end-to-end data workflow with monitoring, and adaptability across the far Edge and Cloud. Supporting continuous architectural adaptation and making real-time decisions about where data is computed and stored is also a critical aspect to handle.

Addressing these challenges requires the development of solutions capable of automating the management of applications (setup and run-time) across the Edge-Cloud continuum. Currently, *container orchestrators* [5], with *Kubernetes (K8s)*⁴ as the de-facto container orchestration solution, support application deployment and reduce human intervention during application setup and run-time. However, their effective operation across a heterogeneous and mobile Edge-Cloud environment requires adaptation, as they were primarily designed for Cloud-based deployments.

In this context, this perspective paper describes a novel, context-aware and cognitive decentralized container

orchestration framework that is currently under development in the Horizon Europe project *COgnitive Decentralized Edge-Cloud Orchestration (CODECO)*.⁵

The work described in this paper aims at providing a perspective on the CODECO concept and advocate a need of such an orchestration framework in the context of next generation *Cloud-Edge-IoT (CEI)* environments. CODECO is in an early stage of development and will only be fully concluded as a framework by the end of 2025. A first complete release of CODECO is expected in June 2024.

This paper focuses on the presentation and discussion of the CODECO framework. It includes the following contributions:

- describes use-cases that require flexible and dynamic orchestration across Edge-Cloud.
- Presents the novel Edge-Cloud CODECO containerized application orchestration framework, software-based, describing its functional components and operational workflows.
- Defines, via the CODECO experimentation framework, operational guidelines to deploying and testing flexible orchestration in existing operational environments.
- Provides the research community with an explanation on the developed open-source software, in particular regarding the early-release⁶ of CODECO components and tools, such as the integration of context-awareness into the Edge-Cloud orchestration; decentralized learning approaches that can be tested via the provided code; network probing mechanisms.
- Provides the research community with the experimentation approach under development in CODECO, which includes the integration of CODECO into the international experimental testbed EdgeNet.⁷
- Provides a thorough comparison of CODECO against other existing orchestration frameworks thus examining different orchestration features, and challenges.

The remainder of this paper is organized as follows. Section II provides terminology and background on K8s, to assist the reader in better understanding the principles behind the design of the CODECO framework. Section III discusses use-cases where this type of container framework is expected to be applied. Section IV describes the CODECO framework and its sub-components, while Section VII presents existing challenges of orchestration across Edge to Cloud, and how CODECO aims to answer such challenges. Section VIII gives insight on how the CODECO open-source framework can be used for performing experiments. Section IX describes efforts similar to this work, namely, other orchestration frameworks, explaining key contributions from our work. The paper concludes in Section X,

⁵<https://he-codeco.eu>

⁶the first CODECO open-source toolkit is to be released in June 2024 via its public Eclipse repository, <https://gitlab.eclipse.org/eclipse-research-labs/codeco-project>. The current code represents an early-release which is regularly updated

⁷<https://www.edge-net.org/>

¹<https://www.docker.com/>

²<https://gaia-x.eu/>

³<https://internationaldataspaces.org>

⁴<https://kubernetes.io/>

summarising current benefits of the CODECO approach, its main features, and future work, derived from existing challenges.

II. BACKGROUND AND TERMINOLOGY

This section summarizes the basics of the overall K8s operation, and provides a subset of notions and terminology related to dynamic container orchestration [6] to help the reader obtain a quicker grasp of the proposed CODECO architecture. A full set of definitions is available in CODECO Deliverable D9 [7].

An *application* is considered to consist of multiple micro-services that can run autonomously based on container technologies, such as Docker. This is called a *containerized micro-service*, or *containerized application*. *Application workload* refers to the containerized micro-services, namely, the binary system, data, and state, that is, a set of global variables defined in the micro-services and required at run-time. Thus, a containerized application consists of one or more containerized micro-services. These micro-services are connected to each other via specific connection policies. Different application micro-services can be deployed on different nodes, i.e., different cyber-physical systems (including virtual machines). To scale the application, allowing it to meet specific requirements such as bounded latency, it is possible to consider migration strategies, e.g., *transparent replication* or *offloading*, also known as *relocation*. The applications supported in this process can be *stateless* or *stateful*. Stateless applications do not require data storage to work. An example is a Web search. Stateful applications maintain the state on clusters, and require that the state (data, application status) be kept and eventually discovered.

Furthermore, the Edge-Cloud definitions in this paper, including the notions of *far Edge* and *near Edge*, follow the line of thought being driven in the European initiatives EUCEI⁸ and its precursor EU-IoT and *Next Generation IoT (NGIoT)*⁹ and, the vision for smart, decentralized Edge-Cloud environments for IoT applications [8].

As defined in K8s, a *container* is a package with general application settings (workload, state, data) to allow an application to run independently and logically within a *pod*. A pod is a logical wrapper entity for containers to run on a K8s cluster. This logical wrapper “holds” a group of one or more containers with shared storage and network resources, and a common namespace that provides a definition for running containers. Thus, a pod is the unit of replication in a cluster. A (K8s) cluster corresponds to the logical environment in which pods run in a way that has been orchestrated by a human operator (the user).

The high-level operation of K8s is illustrated in Figure 1. A K8s cluster consists of multiple worker nodes, where pods are scheduled, and a master node, which corresponds to the K8s control plane. K8s nodes run in Edge-Cloud

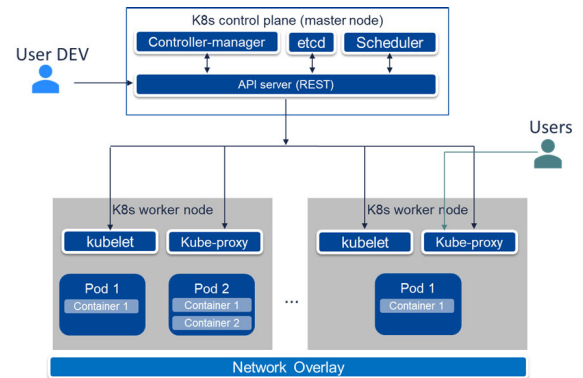


FIGURE 1. High-level perspective of the K8s architecture, 1 cluster, 1 master and 2 worker nodes. User DEV represents a developer wanting to deploy an application in a specific environment. K8s considers that environment to be related with available computational nodes only, networking resources are not managed by K8s. Users represent the K8s users. The figure is supported by the description provided in section II.

nodes, i.e., cyber-physical systems. It should be highlighted that the K8s architecture considers that clusters holding a single master node are extremely vulnerable to failures. Therefore, an architecture involving at least three master nodes is typically considered. For the sake of simplification, this description considers a simple cluster with a single master node.

The *master nodes* (K8s control plane) store configuration and data and manage *worker nodes* and pods in a cluster. For this, the master nodes integrate *etcd*, a key-value internal database, which keeps the configuration of a cluster, is based on the Raft consensus protocol [9], and which stores/manages, among other K8s resources, the *K8s Custom Resource Definitions (CRDs)*, the *API server (K8s API or kube-apiserver)*, *scheduler*, and *controller-manager*. The K8s API is the control plane front-end component.

The *controller* is a control plane component (control loop) that runs processes that continuously check the state of the cluster and compare it to the desired state in *etcd*, and then make or request changes if required. The controller usually performs re-scheduling based on actions provided by the API server, but it can also execute the action itself. For example, the controller can scale the nodes in a cluster. The desired state of a cluster is defined by a user via the API server. In CODECO, a user can be either i) an application developer (user DEV) who wants to deploy an application across the Edge-Cloud; or ii) a K8s infrastructure manager (user MGR) who wants to automate the configuration, deployment, and management of clusters.

In K8s, the API server receives the cluster configuration and application requirements from the user and stores them in a K8s format (*Custom Resource Definitions, CRDs*¹⁰), based on YAML [10], in *etcd*. CRDs provide a definition of *Custom Resources (CRs)*, listing all the configuration available to users.

⁸<https://eucloudedgeiot.eu/>

⁹<https://ngiot.eu/>

¹⁰<https://kubernetes.io/docs/reference/generated/kubernetes-api/v1.28/#customresourcedefinition-v1-apiextensions-k8s-io>

A *K8s operator* (a combination of CRs and a custom controller) is a software extension that provides support for packaging, deploying, and managing K8s resources. The operator configuration is provided to the user in a CRD; therefore, the operator is associated with a CRD/CR. The operator monitors the CR and performs resource-specific actions to ensure that the current state matches the desired state in that resource.

A *scheduler* (in the case of K8s, kube-scheduler; in the case of CODECO, a new scheduler, SWM-scheduler) handles the pod-to-node matching decisions, so that kubelet on the worker nodes can then execute the pods. Kubelet is the entry point to/from the K8s control plane.

For the matching, K8s relies on a *filtering and scoring* approach. In a first phase (filtering), the scheduler checks which nodes can satisfy the scheduling requirements. These nodes are called feasible nodes. In a second phase (scoring), the scheduler ranks the feasible nodes for the “best” pod deployment, by calculating scheduling priorities, also defined in the desired state.

The Edge-Cloud applications are therefore orchestrated via multiple clusters, where an Edge environment, or an Edge-Cloud environment can be within a single cluster (e.g., if under the operation of a same service provider) or within multiple clusters (e.g., across multi-domain environments). It is important to highlight that an Edge node is different from a K8s node. A K8s node may or may not reside on an Edge node.

III. THE NEED: CODECO USE-CASES

CODECO is being deployed on six innovative use-cases in four different European competitiveness markets: Smart Cities, Energy, Manufacturing and Smart Buildings. In this section, a summary of such use-cases is presented for the sake of clarity. A detailed explanation of each use-case, including equipment, technologies, stages and timeline of development is available in the CODECO report D8 [11]. A summarized version detailing pre-conditions, triggers, deployment and performance KPIs is publicly available via a report from the *Alliance for IoT and Edge Computing Innovation (AIOTI)*¹¹ [12].

A. P1: SMART MONITORING OF PUBLIC INFRASTRUCTURE

The overall objective of P1 is to improve traffic flow and pedestrian safety in the city of Göttingen and to contribute to the strengthening of the existing Smart City concept through the implementation of a perimeter road monitoring and analysis system. This system consists of two parts: traffic monitoring at the city Edge and pedestrian distribution monitoring in the city center. By collecting and analyzing valuable data on traffic and pedestrian behavior at the Edge, this use-case aims to optimize management, reduce congestion and improve overall pedestrian safety and

comfort, while also providing valuable insights for urban planning.

P1 is setting up two specific zones in Göttingen: the city periphery, where there is a high volume of vehicular traffic, and the city center, where pedestrian activity is most concentrated. In a first phase of operation, these two areas will be considered as a single cluster (together with the Cloud server(s) operated by the city and by CODECO).

The periphery of the city is being equipped with a combination of thermal cameras, computing units, LiDARs, and communication units. This will enable the real-time collection and analysis of traffic data, tracking vehicle counts and congestion levels. This information can be used to optimize traffic flow, reduce bottlenecks, and improve overall traffic efficiency.

The collected data is relevant to improve pedestrian safety, to manage crowd flow and to notify urban planning initiatives. The back-end data center can receive real-time processing results at the Edge and visualize them to the public.

This pilot scenario, combining technological advancement and data-driven decision-making, is the first step in transforming Göttingen into a truly smart city, improving the quality of life for its residents and visitors alike.

Edge nodes, co-located with the cameras, represent K8s worker nodes; the control plane is expected to reside in the Cloud. Therefore, in the context of this use-case, CODECO is being set up to orchestrate (reallocate) resources across Edge-based environments to support the degree of control decentralization. The overall goal of this use-case is to improve traffic flow, pedestrian safety and the smart city concept in Göttingen by implementing a road/street monitoring and analysis system.

In this use-case, the advantages that CODECO expects to bring are:

- **Scalability and Resilience:** CODECO facilitates system scalability and resilience by enabling each location with an Edge device and sensors to act as a worker node within the K8s system. This distributed architecture supports independent computation and data processing while ensuring connectivity with the broader network. It effectively handles growing data volumes, traffic demands, and scales to satisfy requirements.
- **Efficient Data Pre-processing and Storage:** The CODECO framework enables orchestration for local data aggregation, improving data processing and storage efficiency. It achieves context-aware placement of application workloads across different Edge nodes deployed in the city.
- **Automated Network Management and Adaptation:** CODECO automates the setup of interconnections for Edge-Cloud operations, reducing manual efforts and time invested in network configuration and maintenance. This streamlines the implementation of a traffic and pedestrian monitoring system, particularly when

¹¹<https://www.aioti.eu>

integrating various network environments (e.g., wireless and cellular), resulting in operational simplicity and reduced complexity.

- **optimization and Valuable Insights:** Through the collection and analysis of data on traffic and pedestrian behavior, this use-case aims to optimize traffic management, alleviate congestion, and enhance pedestrian safety and comfort. The CODECO framework employs data-computing-network orchestration across Edge-Cloud resources, contributing to improved user *Quality of Experience (QoE)*.

B. P2: VEHICULAR DIGITAL TWIN FOR SAFE URBAN MOBILITY

P2 uses the CODECO framework to support a Vehicular Digital Twin aimed at improving the safety of *Vulnerable Road Users (VRUs)* in urban environments. Any mobility-focused Digital Twin requires the comprehensive deployment of ultra-reliable, low-latency services around the domain it supports. From *Vehicle to Everything (V2X)* communication capabilities to *Computer Vision (CV)* detectors capable of tracking all moving parts within the mobility environment. For this reason, the current use-case relies on *V2X Roadside Units (RSUs)* and cameras to collect all the necessary information to track vehicles and pedestrians, and then feed it to the vehicle's Digital Twin, which will detect and warn of dangerous situations or behaviors. The deployment and scalability of this service presents challenges on the infrastructure side, where the information needs to be processed as close as possible to the V2X nodes and low latency communication is required. This, in turn, means keeping track of all the moving parts at all times. The pilot scenario focuses on the mobility environment of the interior and adjacent street of the UPC Campus Nord¹² in Barcelona. This environment offers an interesting balance between walkable pedestrian areas with bicycle lanes and car lanes on the adjacent street. It includes a mix of different modes of transport, with VRUs playing a central role. However, VRUs can find themselves in dangerous situations when sharing space with cars. This scenario provides an ideal testing ground due to its size, allowing the examination of multiple areas and their respective control measures. It also provides a diverse representation of all modes of transport commonly found in urban environments. As a result, this scenario provides the perfect setting to assess and address the challenges associated with different modes of transport to ensure the safety and efficiency of urban transport networks.

In this use-case, the advantages that CODECO expects to bring are:

- **Ultra-Reliable Low-Latency Services:** The provision of ultra-reliable, low-latency services is essential for this use-case, given the need for real-time tracking and communication between V2X nodes and CV detectors. CODECO overcomes the challenges associated with

infrastructure deployment and ensures that information is processed as close to the V2X nodes as possible, thereby minimizing latency and enabling efficient and responsive communication.

- **Security and Transparent Cluster Setup:** CODECO ensures the security of the communication between the different nodes in the system at the network layer and also at the data space layer (privacy preservation), focusing on resource efficiency from a data-computing-network perspective, increasing the resilience of the system and the integrity and confidentiality of the data transmitted within the system.
- **Optimal workload placement via context-aware Edge selection:** CODECO relies on context awareness to support an optimal selection of Edge nodes based on specific constraints (e.g., application constraints). Specific components monitor the status of the infrastructure. CODECO's ML-based orchestration engine supports a long-term analysis based on feedback from the scheduler component.
- **Scheduling and Workload Migration:** CODECO handles the scheduling and workload migration of application modules based on vehicle or pedestrian characteristics. This ensures that information is processed within the appropriate constraints, thereby optimizing system performance and efficiency.

C. P3: MEDIA DELIVERY STREAMING ACROSS DECENTRALIZED EDGES

P3 focuses on the smart and efficient distribution of media content (e.g., video streaming, gaming, *Augmented Reality/Extended reality (AR/ER)* across a multi-domain, multi-cluster Edge-Cloud. The use-case leverages a combined optimization of both connectivity (from the underlying transport network) and computational resources (supporting MDS streamers and distribution logic). P3 promotes a tighter computational/networking integration and optimizes the overall resource usage while achieving a good level of QoE. The use-case focuses on an interaction between a *Media Delivery System (MDS)*, via CODECO, where a specific CODECO component, NetMA (rf. to section IV) relies on a decentralized concept of the IETF ALTO protocol to expose capabilities (e.g., topological information together with associated metrics, available resources, or functions) that promote joint adaptation. CODECO is used to support smart Edge selection taking into consideration both computational and network-awareness, as well as user preferences. The aim of this use-case is to enable a smart and efficient distribution of media content across a multi-domain and multi-cluster Edge-Cloud environment to ensure a high level of QoE for users.

In this use-case, the advantages that CODECO expects to bring are:

- **Orchestration:** The CODECO framework enables the selection of the most appropriate Edge facility based on

¹²<https://www.upc.edu/campusnord/ca>

specific constraints on both the Edge computing (CPU, RAM, and storage) and network (latency, bandwidth) sides. By leveraging CODECO's capabilities, the system can make optimal resource allocation decisions to ensure the efficient delivery of media content.

- **Cognitive Approach and Resource optimization:** CODECO promotes a cognitive approach that facilitates the joint articulation of data, computation and network adaptation. It exposes functionality as a service to support optimal resource usage decisions, improving the performance and overall efficiency of media content distribution.

D. P4: COLLECTIVE DEMAND SIDE MANAGEMENT IN DECENTRALIZED GRIDS

The proposed use-case for the distributed energy management system focuses on the implementation of a decentralized active demand response management system for the decarbonization of buildings. It aims to optimize energy use, improve sustainability and increase the resilience of buildings by integrating renewable energy sources and enabling intelligent demand response actions. The use-case also emphasizes the joint orchestration of computing and networking resources to ensure efficient coordination and management of energy-consuming devices and network infrastructure within buildings. It focuses on achieving a holistic view of data across the CEI continuum, enabling the comprehensive monitoring, analysis and replication of energy-related data. The CODECO framework leverages the power of K8s to build a distributed energy management system. By integrating worker nodes (which correspond to Edge nodes in this use-case), P4 aims to achieve efficient resource utilization, scalability, resilience and adaptability in energy management operations, integrating the energy-related IoT systems and computing requirements. With CODECO, the following benefits are expected to be achieved:

- **Automated Configuration and Cognitive Edge-Cloud Management:** The CODECO framework enables the automated configuration and cognitive management of Edge and Cloud resources. In the context of the use-case, this means that the CODECO framework can dynamically allocate and optimize resources based on demand response requirements, real-time energy data, grid conditions, and consumer preferences. The CODECO automated and cognitive approach ensures the efficient and intelligent management of resources, leading to optimized energy usage and improved sustainability.
- **Efficient Data Collection and Analysis:** CODECO's MDM component (rf. to section IV) provides tools for the efficient collection, analysis and processing of energy consumption data and relevant contextual information. By leveraging MDM capabilities, the use-case can effectively monitor and analyze energy-related data across the CEI continuum. This comprehensive

view of the data enables informed decision-making and proactive management of energy-consuming devices.

- **Resource optimization for Real-Time Demand Response:** CODECO provides the capabilities to optimize the allocation of computing resources. In the use-case, CODECO can be used to allocate resources for real-time demand response decisions. This ensures effective load management and energy optimization by allocating resources in a way that maximizes the efficiency of demand response actions.
- **Holistic View and Comprehensive Monitoring:** The CODECO framework enables a holistic view of data across the CEI continuum. By integrating data from different sources and devices, the use-case can comprehensively monitor energy consumption, grid conditions and other relevant factors. This comprehensive monitoring enables a better understanding of energy patterns, facilitates data-driven decision-making, and supports the replication and analysis of energy-related data for further optimization.

E. P5: DECENTRALIZED CONTROL OF AGVS OVER WIRELESS

Currently, there is an increasing need to consider *Automated Mobile Robots (AMRs)*, such as *Automated Guided Vehicles (AGVs)*. While current AGV fleets are based on pre-defined task assignments and pre-defined paths, there is an urgent need to provide a more flexible control to support fleets with a larger number of AGVs. On the other hand, it is important to support the adaptation of heavy ML-based processes to constrained Edge nodes (AGVs in the use-case). Furthermore, the integration of wireless technologies such as Wi-Fi6 and 5G to support a decentralized and semi-autonomous control of AGVs brings challenges to these use-cases, as the application deployment across an AGV fleet needs to take into consideration aspects such as interference and intermittent connectivity. CODECO as an orchestration framework is applied in this use-case to assist in an optimized deployment of the applications required to sustain a decentralized behavior in the AGV fleet, with the aim of achieving better energy efficiency and a more resilient infrastructure. The use-case explores AGVs being assigned tasks to handle, and potential failures in the fleet, allowing AGVs (with the support of CODECO) to react in a semi-autonomous manner.

The AGVs carry various micro-services (dockerized), such as publish/subscribe communication services; path tracking services. AGVs are considered as K8s worker nodes, whereas the control plane will reside on either static or mobile nodes. AGV micro-services are managed via CODECO, where the CODECO components will be placed over the control and data plane of K8s. In this use-case, the advantages that CODECO expects to bring are:

- **Flexible Control and Task Assignment:** The CODECO Framework provides a flexible control system for AGV

fleets, allowing dynamic micro-service assignment and adaptation across the available mobile nodes. By achieving a higher level of autonomy, the deployed application workload can increase overall efficiency and reduce operating costs.

- **Integration of Wireless Technologies:** The use-case explores the use of network-awareness to perform the distribution of micro-services. It takes into consideration wireless networking aspects such as signal strength, and applies these metrics to the ranking of available nodes to deploy the application. CODECO provides the necessary adaptive capabilities to ensure reliable communication despite potential interference and failures.
- **Federated Clusters and Flexibility:** The ACM component of CODECO supports the setup of single and federated clusters, allowing for better coordination and reduced signalling overhead. This flexibility in cluster configuration enables efficient management of AGVs across different locations. The use-case can leverage CODECO's ACM capabilities to optimize resource allocation, reduce latency, and energy consumption in AGV fleets.
- **Real-time Metadata and ML-based Orchestration:** CODECO provides real-time metadata to support ML-based orchestration. This capability allows for efficient and intelligent decision-making based on the current state of AGVs and the factory environment.
- **Context Modelling and Edge Selection:** CODECO supports context modelling based on data (user, network, computing). This aggregated context is used to best select suitable nodes to perform the required operations in the fleet, e.g., build a map, perform navigation, or take over tasks assigned to other AGVs.
- **Scheduling and Workload Migration with Intermittent Connectivity:** CODECO supports scheduling and workload migration in the presence of intermittent connectivity. AGVs in a wireless environment may face intermittent connectivity, and CODECO provides mechanisms to handle such situations, thereby ensuring the smooth operation of AGV systems.

F. P6: AUTOMATED CROWNSTONE APPLICATION DEPLOYMENT FOR SMART BUILDINGS

CODECO P6 focuses on novel mechanisms for the automated deployments of Smart Facilities (e.g., buildings, offices) by considering applications on the Crownstone Platform.¹³

Crownstones are small IoT devices containing a BTLE MCU, a relays, a dimmer circuit and the ability to measure current and tension with a high sample rate of 5kHz. This enables the firmware on the device to provide accurate insight in the behavior of the grid as well as the potentially connected fixture or appliance. Possible anomalies can be detected swiftly and acted upon. Crownstone is intended

to be encapsulated behind wall sockets, switches, and light fixtures, so is generally not visible. The memory and CPU capabilities allow additional functionality to be installed on the Crownstone in the form of a microapp. These microapps run in a sandbox, are capable of connecting to another device using communication technologies such as BTLE or I2C, and can then be used as an extension of the functionality of the Crownstone (e.g., a PIR sensor or a door lock). The use-case focuses on the deployment and management of these microapps, as they can be many and varied.

In this context, an application is defined as a collection of related functionalities realized by means of a set of interconnected application components which can run either in the Cloud, on the Crownstone Hub, or inside a Crownstone Node. The key issue we will address is how the CODECO technologies can help with automated deployment of multiple applications on the Crownstone platform, both in single cluster situations (where multiple Crownstone Hubs form a single manageable entity with a single user base), and in multi-cluster situations (where multiple Crownstone Hubs form multiple manageable entities with different but potentially overlapping user bases).

The main advantages of the CODECO framework are:

- **Efficient Management of Single and Multi-Cluster Situations:** The use-case addresses both single cluster situations, where multiple Crownstone Hubs form a single manageable entity with a single user base, and multi-cluster situations, where multiple Crownstone Hubs form multiple manageable entities with different user bases. CODECO facilitates the efficient management of these situations. It enables the management and coordination of resources, applications, and user bases in a scalable and flexible manner. This allows for effective management of deployments across different clusters, ensuring optimized performance and user experience.
- **Real-time Metadata Management:** CODECO plays a crucial role in this use-case. It provides real-time metadata that supports the deployment and management of smart office/smart building applications. CODECO collects, analyzes, and processes relevant metadata, including application-related and contextual data. This enables intelligent decision-making and optimization during the deployment process, ensuring efficient resource utilization and enhanced application performance.
- **Streamlined Application Management:** CODECO streamlines the management of interconnected application components. The framework provides a unified approach for managing applications across multiple environments, simplifying deployment, monitoring and maintenance processes. This streamlining of application management improves efficiency, reduces complexity and improves the overall user experience.

¹³<https://crownstone.rocks/>

IV. CODECO FRAMEWORK OVERVIEW

CODECO and its components, represented in Figure 2, form a software-based container orchestration framework that is interoperable with K8s. CODECO aims to support a next generation of container orchestrators that can adapt and learn, developing an appropriate response and adaptation to the diverse requirements coming from the data, the application, the system, the network, and the end-user. To the user, CODECO has a single interface based on the CODECO ACM component. ACM handles the operations required to support an application deployment across far Edge-Cloud, considering the input provided by the user (application requirements, user requirements defined in terms of networking, computing, data). ACM installs the CODECO components and the respective integration points between users and applications, where the user in CODECO is an application developer (user DEV) or a cluster manager (user MGR). ACM takes care of the overall CODECO configuration, the acquisition of new nodes, and the interaction with non-K8s systems. Furthermore, ACM relies on Prometheus¹⁴ and integrates a CODECO monitoring framework, currently focused on infrastructure monitoring (data, network, computing) based on application requirements and still under development. Therefore, ACM is co-located with the control plane of the K8s (master nodes).

The CODECO MDM component provides data workflow observability to the other CODECO components, treating data as an integral part of the application workload, and integrating data observability perspectives from different categories, for example, application, system, and network perspectives, at different points in the CODECO operational workflow.

SWM handles the scheduling and re-scheduling of the application workload, based on the CODECO Application Model (supported by ACM and provided by the user during application setup), based on the novel data-computing-network approach proposed by CODECO. The currently available approach for handling placement decisions relies on a solver which in the future is expected to provide an optimal match between application requirements and available resources (computational, network, data). SWM is also a control plane component, co-located with ACM and the K8s control plane, in master nodes.

PDLC is at the heart of CODECO orchestration. Based on the infrastructure data collected by ACM (via Prometheus), NetMA and MDM, PDLC has two functions. First, it provides data-computing-network node costs (aggregated node costs) based on specific target performance profiles selected by the user (e.g., optimizing the overall infrastructure for resilience). This implies that the notion of a node in the infrastructure embodies network, data, and computing-awareness. Second, it provides an estimate of overall system stability based on privacy-preserving decentralized learning approaches. PDLC

is currently envisioned to run on both master and worker nodes of K8s.

NetMA provides network-awareness to CODECO and handles secure connectivity across pods. For this purpose, NetMA exposes networking parameters that are relevant for reaching a close-to-optimal workload placement. For connectivity, NetMA handles the *Software Defined Network (SDN)*-to-K8s interaction via the L2S-M open-source solution. Its sub-component *Network State Management* handles network monitoring, and also receives network forecasting provided by PDLC.

The monitoring of the overall infrastructure from different perspectives is supported by different CODECO components: NetMA monitors the networking infrastructure, MDM monitors the data workflow, and ACM monitors the system (computational nodes) infrastructure. Before giving insight into each component, the next section provides an explanation of the overall operation of CODECO.

V. OPERATIONAL WORKFLOW EXAMPLES

A. CREATING AN APPLICATION DEPLOYMENT

DEV is a user (application developer) deploying an application consisting of multiple micro-services (multiple containers) across the far Edge to the Cloud. DEV downloads CODECO from the CODECO Eclipse GitLab and follows the instructions to set up ACM. ACM performs cluster sizing based on the *CODECO Application Model (CAM)*, a YAML file accessible by user DEV via the ACM dashboard during application deployment setup. In this file, the user DEV defines aspects such as the desired *Quality of Service (QoS)*, *Quality of Experience (QoE)*, or other desired performance levels for CODECO, based on specific questions provided in the ACM dashboard. The current attributes considered in the CODECO CAM are available via the report [13]. Based on the specific parameters (representing the application requirements), ACM builds the CODECO CAM and makes it available to all CODECO components.

ACM also handles the complete K8s setup (e.g., namespace, databases, secrets) and makes the information available to other K8s components as needed. For example, metadata information, schema, can be passed to MDM (rf. to Figure 2, I-ACM-MDM-2). Application requirements derived from the CODECO Application Model, e.g., dedicated CPU, required bandwidth, are made available to SWM (rf. to Figure 2, I-ACM-SWM-1) and to PDLC (rf. to Figure 2, I-ACM-PDLC-1), for instance.

The exposure of requirements and application/user information also triggers the operation of each CODECO component. PDLC defines the processes for activating the sensing and (decentralized learning) processes for the cluster. SWM makes a request to PDLC to obtain the initial weights to be considered for scheduling optimization (I-PDLC-SWM-1). NetMA triggers the definition of the network overlay when it receives the initial deployment from SWM (I-SWM-NET-1).

After activation, PDLC periodically obtains, via available *Custom Resources (CR)/Custom Resource Definitions (CRD)*

¹⁴<https://prometheus.io/>

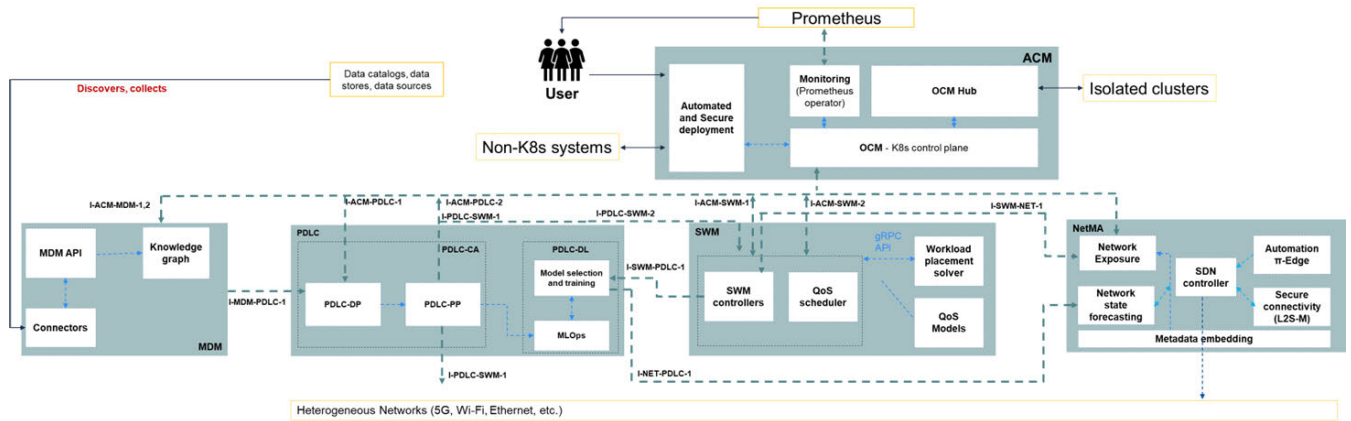


FIGURE 2. The CODECO framework and its software based components. Each component, described in section IV, is being deployed as a set of containerized micro-services. ACM, SWM reside on the control plane.

(I-ACM-PDLC-1,2), metadata provided by the components monitoring the data-computing-network infrastructure, i.e., MDM (data observability metrics, ACM (user aspects and application constrains via the CODECO Application Model), NetMA (network metrics). Then, PDLC assigns suitable nodes a combined data-network-computing cost based on the user selected target profiles, and stores the output on a PDLC CRD, making it available to other components such as ACM and SWM, which may trigger adjustments to the initial setup process.

In parallel, the three components begin monitoring different metrics. NetMA captures network metrics at the node, link, and path levels, from an overlay and underlay network perspective. This information is then exposed via specific CRs that are accessible to all components and used by PDLC and ACM. Similarly, MDM captures data aspects (e.g., data compliance), generates a knowledge graph and provides the output as an MDM CR. ACM captures user preferences and eventually behavior, which may be useful for adapting the overall K8s infrastructure at a later stage.

B. CODECO SUPPORT DURING CLUSTER RUN-TIME

Once the setup is complete, CODECO enters the cluster management phase (application run-time support), targeting user MGR. During this phase, the proposed application (CODECO application workload) has been set up and is running on several containers (1 cluster), 1 or more pods per worker node. PDLC periodically receives data from MDM (I-MDM-PDLC-1); from NetMA and ACM (I-ACM-PDLC-1) and feedback from SWM regarding the placement of the application workload (I-SWM-PDLC-1). Based on this, PDLC periodically evaluates the proposed system performance targets (e.g., greenness, service latency) provided by Bob during application setup, and provides a cost combination per infrastructure element via a CR (I-PDLC-ACM-2). If there is a need for cross-layer redistribution of the application workload, this step triggers a request from ACM to all CODECO components. In this case, PDLC passes a behavior estimate to SWM (I-PDLC-SWM-2) via a specific

CR; SWM starts the workload placement process. Once the process is complete, SWM passes feedback to PDLC (I-SWM-PDLC-1). This will not be an explicit interface; instead, feedback will be provided via specific SWM CRs (currently ApplicationGroup, Application, AssignmentPlan). ACM handles the status back to user MGR, based on the Prometheus CODECO monitoring architecture.

VI. CODECO COMPONENTS

A. ACM: AUTOMATIC CONFIGURATION MANAGER

The CODECO ACM represented in Figure 2 is based on the *Open Cluster Management (OCM)*¹⁵ community-driven project which is the upstream project for Red Hat Advanced Cluster Management.¹⁶ In CODECO, its operation considers three main aspects that address the integration of CODECO across the entire Edge-Cloud infrastructure:

- **Integration points between users and applications.** Mechanisms for users (e.g., user DEV) to control and change the configuration of applications. A key component in this context is the CODECO Application Model (CAM), explained later in this section.
- **CODECO configuration.** A user request during application deployment setup or application run-time implies the activation and eventual configuration of CODECO components.
- **Cluster/federated cluster configuration.** The user in this case (e.g., user MGR) handles the K8s infrastructure. A specific change in the CODECO configuration may imply the need to reinstall or reconfigure a cluster.

The current ACM sub-components are:

- **OCM** is used to enable end-to-end visibility and control (i.e., control-plane functionality) across K8s-based clusters. OCM will be used to provide the main ACM functionality, and it will be extended to provide support and visibility of the newly added CODECO components.

¹⁵<https://open-cluster-management.io/>

¹⁶<https://www.redhat.com/en/technologies/management/advanced-cluster-management>

- **Monitoring.** Different CODECO components (ACM, NetMA, MDM) collect parameters that will be used to assist in a more flexible schedule, computing, network, and data awareness. The CODECO monitoring architecture interfaces with Prometheus via ACM.
- **Automated configuration via Knative and Ansible.** We will use Knative when we need scalable stateless functions that can be easily scaled-in and -out upon load change and Ansible to support deployment management. This is usually a requirement for many stream processing and event-driven functions.
- **Control plane for independent/isolated clusters.** There are multiple open-source technologies handling this problem (e.g., KCP) and CODECO's choice is yet to be decided. Here, the aim is to consider mobile environments where intermittent connectivity may prevent the registration of a cluster to the OCM Hub.

Since ACM is the integration point towards the user as represented in 3, the user can install the CODECO framework by simply installing the *CODECO meta-operator, codecoapp-operator*¹⁷ where this process triggers the installation of the CODECO framework on the cluster to be managed and configured. Also, there may be the explicit need to export more information from the CODECO system to the cluster control-plane and towards the users, which is also performed by ACM.

An additional relevant aspect addressed by ACM is the *CODECO Application Model (CAM)* [7]. CAM is a model for QoS/QoE requirements of an application, provided by user DEV during the setup of the application, covering requirements from a user, application, data, and network perspective. An example of CAM management by ACM is provided in Figure VI-A. The CAM description is semantic (YAML) and defines QoS/QoE requirements with different levels of granularity, e.g., application, micro-service (container), and pods.

B. MDM: META-DATA MANAGER

The CODECO MDM component¹⁸ (rf. to Figure 2) collects, links and enriches metadata related to the applications to be deployed across Edge-Cloud. This metadata helps to better characterize the application deployment in the Edge-Cloud. MDM is therefore a CODECO component that acts as a gateway between the data world (data workflow) and the K8s infrastructure (computing). MDM is based on the metadata management principles established by the IBM Pathfinder [14].

MDM catalogue information includes attributes and properties such as data source, semantic description, classification, identification; data structure (e.g., data type, schema); application-specific information (e.g., data analysis, curation

aspects); network-specific information (e.g., access latency, bandwidth). The catalogue is continuously populated based on a pull model as new datasets are created. Therefore, it is populated by other CODECO components. MDM keeps track of the metadata and provides a common platform where all metadata (multi-cluster) can be shared across multiple domains. MDM also interacts with other catalogues and provides a starting point for interfacing with the Gaia-X service catalogue composition model and architecture to ensure compliance and wider use.

An MDM connector interfaces with a native data system or CRD and pushes metadata into a knowledge graph using the native (internal) MDM API. By adding new connectors or extending the graph model, the system can be extended to collect any metadata required. MDM materializes (subsets of) the event queue in the knowledge graph to meet the needs of other CODECO components.

MDM integrates three subcomponents: i) MDM Controller API, which implements the REST APIs that allow metadata to be pushed into the graph database and the graph to be queried; ii) graph database, which stores the metadata graph; iii) connectors, which collect metadata and push it into the graph database using the MDM Controller APIs.

The **MDM Controller** provides APIs for other CODECO components to query the metadata graph and for MDM connectors to provide metadata. The MDM Controller is thus a required subcomponent for all scenarios where metadata analysis is required for the CODECO use-case. A selected set of MDM connectors, depending on the use-case, will provide metadata that, through this subcomponent, will allow other CODECO components, such as PDLC, to obtain summarized information about the systems and data in the form of parameters for models that provide the best scheduling for a given workload.

The **MDM graph database** component is the back-end repository of the MDM component. The metadata events from all MDM connectors are consolidated in this component, allowing other components to gain insight into the distributed system from a single pane of glass. It is of course possible to request information by directly querying the database using cypher, but other than during development or exploration of the metadata, the MDM component is designed to provide this functionality via the MDM API.

MDM connectors send metadata to MDM in the form of events. Events are structured JSON documents. An event contains the following elements: i) the event type, either “insert” or “delete”; an identifier that uniquely identifies the connector that issued the event; a timestamp; the payload, i.e., the metadata.

Metadata is transmitted as entities and relationships. MDM imposes a basic structure on both entities and relationships to ensure that metadata can be stored as a graph. Entities must have a globally unique identifier and a type. It is the job of the connector to assign these. In addition, entities contain any number of attributes. The mandatory elements of a relationship are source and target entity identifiers and

¹⁷<https://gitlab.eclipse.org/eclipse-research-labs/codeco-project/acm>

¹⁸<https://gitlab.eclipse.org/eclipse-research-labs/codeco-project/metadata-manager-mdm>

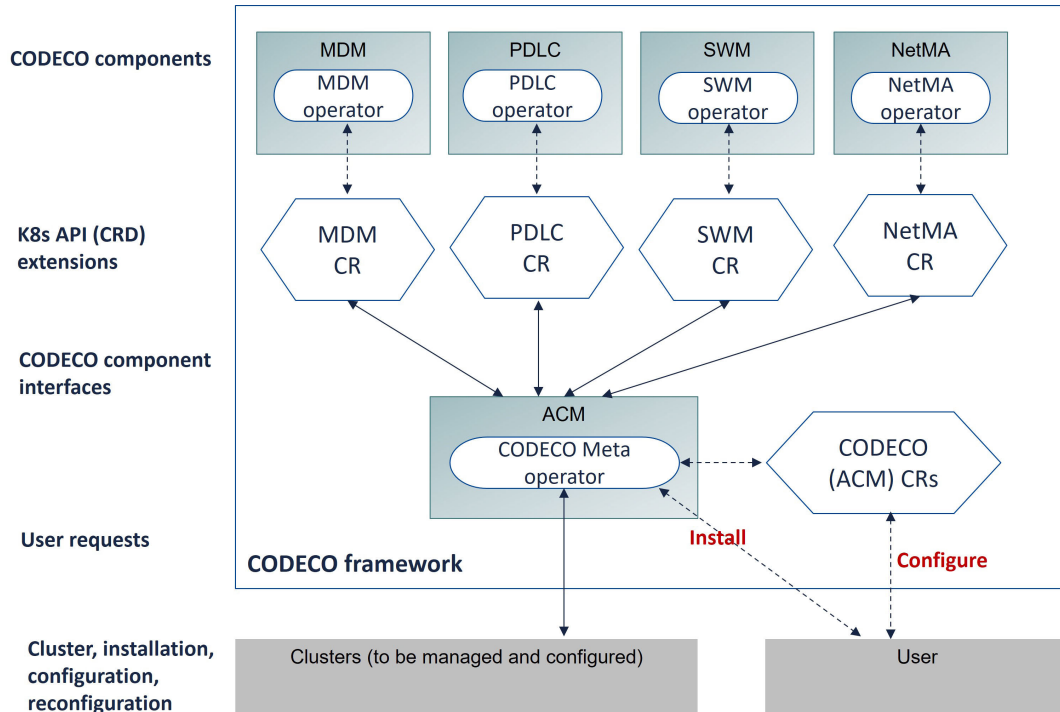


FIGURE 3. The CODECO component ACM, described in section VI-A, is the single point of entrance to the user, and sets the other CODECO components.

a type. These relationships do not have any attributes. MDM does not define or enforce a static data model. Instead, the graph data model is defined by the structure of the entities and relationships inserted into MDM.

C. SWM: SCHEDULING AND WORKLOAD MIGRATION

The CODECO SWM component¹⁹ handles the initial deployment, monitoring and potential migration of application workloads within a single cluster and across multi-cluster environments. This means supporting the efficient placement of applications and their containers across the Edge-Cloud, derived from the information provided by PDLC (e.g., device and node availability, container centrality, and network characteristics). For example, SWM handles the “best” placement (based on context-aware indicators) for the containerized components of an application to be deployed in a cluster, considering the dynamic properties of the available infrastructure, including physical/virtual machines as well as network nodes and links. Furthermore, this placement is dynamically adaptable, which implies achieving efficient (low latency, lower power consumption, data sensitivity, and QoE) migration of containerized micro-services of an application, including their state, across Edge and Cloud, derived from the information provided by PDLC (device and node availability; container centrality, network aspects).

SWM resides on the control plane of K8s. It extends the K8s resource model by some CRs and uses the K8s

controller pattern to implement controllers for these resources as represented in Figure 5.

SWM consists of two subcomponents: The **QoS Scheduler** and the **Workload Placement Solver**. Interaction with these subcomponents is done via K8s CRs. The main interface between the QoS Scheduler and the Workload Placement Solver is a gRPC interface described via Protobuf.

The **QoS scheduler** is a custom scheduler that is built using the K8s scheduling framework. It runs as a pod in the K8s cluster, and it registers as scheduler to the K8s control plane. It implements the so-called scheduling plugins for certain phases of the K8s scheduling life cycle. While the K8s standard scheduler schedules each pod individually, one of the specific mechanisms of the QoS Scheduler is that it decides on the placement of all pods of an ApplicationGroup at once. This is required to consider dependencies between the pods (e.g., communication relations and QoS), and to treat the pod placement as a graph problem, rather than a sequential, linear problem. The deployment of the pods within an ApplicationGroup is held back, until the placement decision is taken, and all communication resources have been confirmed (if applicable). The SWM QoS model consists of two parts: the *SWM Application Model (SAM)* and the *SWM Infrastructure Model (SIM)*. SAM describes the QoS/QoE requirements of an application in specific way that is interpreted by SWM, derived from the global CAM in CODECO. SIM currently describes the hardware entities that form the execution environment for the applications. It consists of computing and network infrastructure, particularly computing nodes, network nodes, and network links. The

¹⁹<https://gitlab.eclipse.org/eclipse-research-labs/codeco-project/scheduling-and-workload-migration-swm>

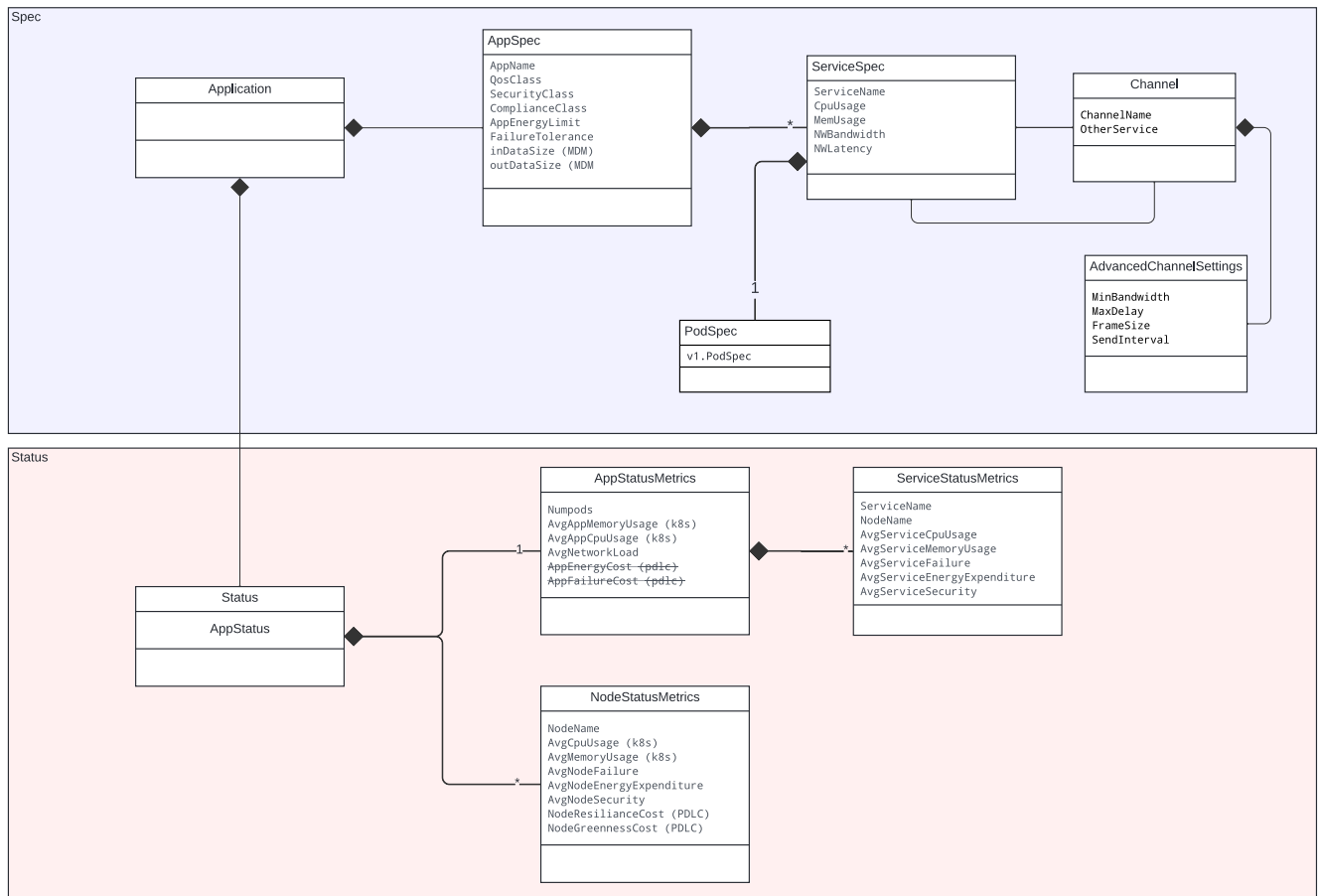


FIGURE 4. Example for a potential definition of the CODECO Application Model (CAM). The CAM, explained in section VI-A, comprises application and infrastructure requirements provided by the user during the setup of an application deployment. It also provides status to the user about the K8s infrastructure and about the application workload.

attributes of SIM express the resources and capabilities of the infrastructure.

The **Workload Placement Solver** is implemented as a gRPC service that handles placement requests. With the request, the client (usually the QoS Scheduler) passes the so-called QoS model as Protocol Buffers (protobuf) messages.

An example of the SWM operation based on Figure 5 is as follows. Via ACM, the user DEV describes the application QoS/QoE requirements (YAML file(s) CAM). SWM relies on this file and translates it into SAM and SIM to create a description of the desired deployment, considering the CRs *ApplicationGroup* and *Application* (including workloads, channels and all relevant attributes). Once the minimum number of Applications that are part of the ApplicationGroup have been created, the ApplicationGroup custom controller collects all the information required for placement (SWM QoS Model). The SAM includes the CRs Applications, Workloads, Interfaces. The SIM is compiled by retrieving information from the associated infrastructure custom resources: *Node*, *Endpoint*, *NetworkLink*, *NetworkNode*, *NetworkPath*. The *ApplicationGroup* CR calls the Workload Placement Solver and passes the QoS model. The Workload Placement Solver determines a placement for the application

workloads, taking into account all constraints and (if it is an optimization solver) optimized according to the defined objective. The result is returned and placed in the CR AssignmentPlan.

Then a pod is created for each *Workload* that could be placed, and the preferred node for the pod is set according to the placement decision. However, the deployment of the pods is still delayed. For each *Channel* between *Workloads* that could be placed, a CR Channel is created. Channels between workloads placed on the same node are marked as “loopback channels” (they do not require connections over the network).

For all other channels, the relevant network controller (which monitors the creation of the channel) will react accordingly. Depending on the network technology and implementation of the network controller, a response could be to monitor the occupied bandwidth for the channel and/or to reserve/establish an end-to-end channel through the network. In any case, the success or failure to establish an end-to-end channel must be reported via the state of the CR channel. Once all channels of the ApplicationGroup are in the state “ACKNOWLEDGED”, the deployment of the pods of the ApplicationGroup will be initiated. K8s will

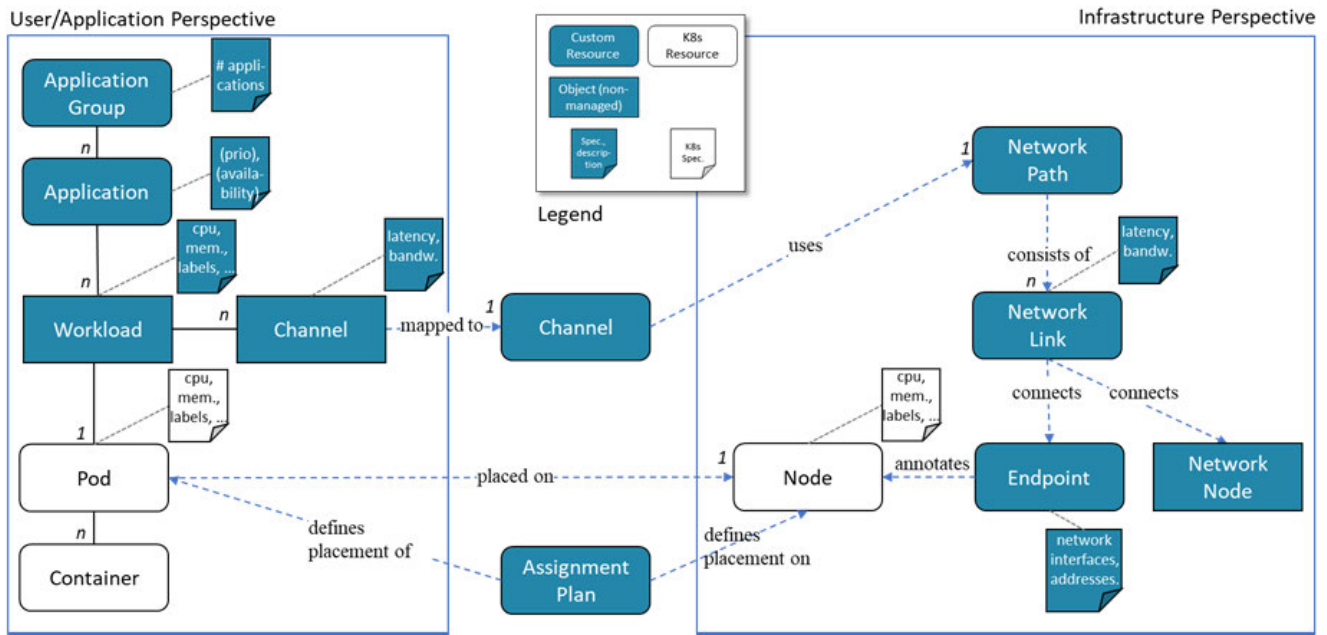


FIGURE 5. Representation of the CODECO SWM component Custom Resources, from the user and from the infrastructure (network, compute) perspective. The CODECO SWM component is explained in section VI-C.

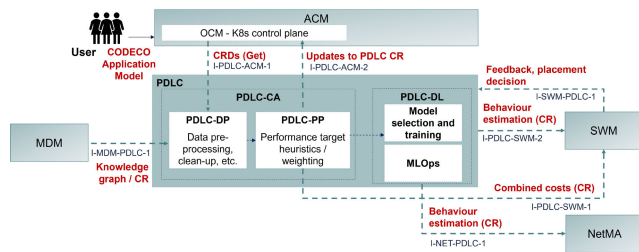


FIGURE 6. Internal architecture and interactions between the PDLC subcomponents and other CODECO components.

deploy the pods, resulting in the download and launch of the corresponding Containers in the selected Nodes, and the Applications will become operational.

D. PDLC: PRIVACY-AWARE DECENTRALIZED LEARNING AND CONTEXT AWARENESS

PDLC²⁰ is the heart of the CODECO cognitive orchestration and currently consists of two sub-components: PDLC-CA (Context-awareness) and PDLC-DL (privacy preserving Decentralized Learning). The PDLC sub-components provided in the CODECO GitLab public repository and their interactions with other PDLC components are shown in Figure 6.

PDLC-CA is responsible for the integration of context-awareness into the CODECO framework. The sub-component obtains data from other components (e.g., network metrics from NetMA; user metrics from ACM), pre-processes such data (PDLC-DP) and generates context-awareness based on a specific performance profile

(PDLC-PP) requested by the user, e.g., optimal greenness of the overall system. Specific parameters and categories of metrics to be considered are available in prior work [6], [7], and are illustrated in Figure 7.

For instance, ACM obtains application and user behavior/preferences during the setup of an application deployment via CAM. MDM can tag events that relate to data observability, e.g., if a dataset is not updated within a specified time, or if a dataset grows to exceed a specified size, a trigger can be generated to schedule application to handle the condition. NetMA collects network metrics via ALTO to assist in a better definition of the infrastructure, beyond the usual view of resources provided in K8s. The pre-processed datasets are then made available to the context-aware performance profiling block of the sub-component PDLC-CA, named as PDLC-PP. This micro-service performs a combination of the received data sets based on pre-configured heuristics that aim at providing a measure of performance, for a specific performance efficiency profile. For instance, assuming a user wants to optimize the system for greenness, then this block would select and combine weighted context datasets (e.g., hop count, energy consumption) in accordance with a specific function (e.g., product of hop count and energy consumption).

Hence, this micro-service is being developed to provide support to other plugins in K8s (towards kube-scheduler, I-PDLC-ACM-2) and to provide an aggregated result to SWM (I-PDLC-SWM-1). PDLC-CA can be tested via the initial version that creates an aggregate perspective on the infrastructure greenness and resilience.²¹

²⁰<https://gitlab.eclipse.org/eclipse-research-labs/codeco-project/privacy-preserving-decentralised-learning-and-context-awareness-pdlic>

²¹<https://gitlab.eclipse.org/eclipse-research-labs/codeco-project/privacy-preserving-decentralised-learning-and-context-awareness-pdlic/context-awareness/pdlic-pp>

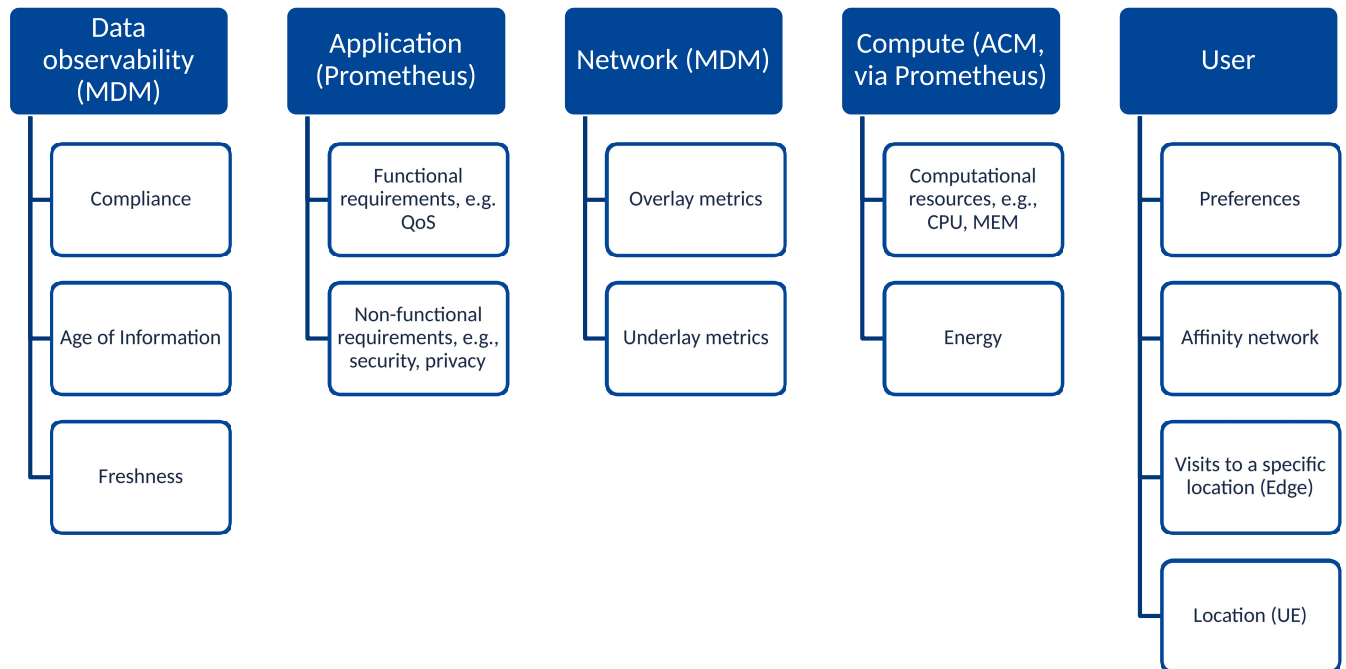


FIGURE 7. Categories of context-aware parameters being used in CODECO. Data observability parameters are provided by component MDM; application parameters are provided by ACM, based on Prometheus and Kubernetes resources. Network metrics are provided by NetMA. User preferences are provided by the user during application setup via the CAM. System parameters are provided by the CODECO internal components, such as NetMA, PDLC, and ACM.

PDLC-DL goes a step further in adding intelligence to the CODECO framework by using the collected raw data and generated context information by PDLC-CA to train decentralized learning models to provide forecasts and predictions about the future behavior of infrastructure nodes and deployed applications, providing this forecasting results to SWM as a cost-based recommendation of suitable nodes for different applications. Currently, the following decentralized learning approaches are being analyzed in PDLC-DL. Proofs-of-concept are available to the reader via the CODECO Eclipse GitLab.²²

- 1) *Reinforced learning (RL)* is being used to provide suitable node recommendations, with the objective of balancing different metrics usage (e.g., CPU, RAM, bandwidth, energy).
- 2) *Graph Neural Network Models (GNNs)* are being applied to perform exhaustion prediction for the values of the monitored metrics of a cluster node.

1) NODE RECOMMENDATIONS BASED ON RL

PDLC is exploring RL to provide CODECO with a way to consider the different proposed metrics (computing, network, data) to achieve a fair environment in the sense that resources are kept at their lowest values, while the number of pods being allocated is kept at a maximum. The importance assigned to these two objectives can be balanced with two weights that largely depend on the use case at hand. The

²²<https://gitlab.eclipse.org/eclipse-research-labs/codeco-project/privacy-preserving-decentralised-learning-and-context-awareness-pdlc/decentralised-learning/model-selection-and-training>

existing code provides the reader with the possibility of exploring two RL algorithms, *Deep Queue Learning (DQN)* and *Proximal Policy Optimization (PPO)*. The model has been developed with a clear objective in mind, that being the ability to easily expand the model in the future with PDLC-CA parameters and make it use-case adaptive. Furthermore, it serves as a foundation for future expansion to multi-agent implementation.

In the current release, the state of the system at a time instance t consists of the following elements:

- $\{C_t(n) \mid \forall n\}$: the set of the used CPU cores C of each system node n at time instance t .
- $\{M_t(n) \mid \forall n\}$: the set of the used memory M of each system node n at time instance t .
- (C_p, M_p) : the requested processing power and memory size of the next pod to allocate p .

Each time an allocation request is received, an allocation of the pod p to a node n is suggested. Pods that are already allocated can be suggested a reallocation or can retain the current allocation if it is still optimal. Furthermore, the model uses a ‘fake’ node that holds all the pods in the system that cannot be allocated at instance t . This fake node is not taken into account when calculating the system’s workload as provided in Eq. 1.

To model the reward function of the RL agent, we included two components:

- A workload balancing component W_t defined in Eq. 1: calculated as the standard deviation of the workloads of all system nodes $W_t(n)$. We calculated the workload

of a node n at instance t by summing the used CPU $C_t(n)$ and RAM $M_t(n)$ of the node, normalized against the node's maximum CPU and RAM, $C(n)$ and $M(n)$ respectively. We introduced two weights w_c and w_m to take into account the relative importance of the CPU and RAM usage in calculating this workload in order to accommodate for different use cases where more CPU-intensive or RAM-intensive tasks need to be allocated.

$$W_t = \sigma(W_t(n) \mid \forall n)$$

$$W_t(n) = \left(\frac{C_t(n)}{C(n)} \times w_c\right) + \left(\frac{M_t(n)}{M(n)} \times w_m\right)$$

where $w_c + w_m = 1$ (1)

- An encouragement component e_t calculated as the number of allocated pods normalized against the total number of pods pending to be allocated at instance t . We introduced this component to encourage the RL agent to allocate as many pods as possible while respecting the workload balancing introduced by the first component. This would avoid issues with the agent learning not to allocate any pods, as doing so would achieve optimal balancing between nodes at 0% of resources consumed.

Combining these two components, we arrive at the reward function r_t defined in Eq. 2. Note that since the RL agent aims to maximize the obtained reward, we multiplied the workload balancing component by -1 . This multiplication effectively means that the agent now aims at minimizing the standard deviation of workloads, thus achieving a higher workload balance. In the future, this function will be extended to include other parameters from the CODECO Application Model (Fig. VI-A) and parameter weighting will need to be studied and optimized thoroughly per use case.

$$r_t = -1 * W_t + e_t \quad (2)$$

2) RESOURCE MONITORING ESTIMATION BASED ON GNN

The current proof-of-concept²³ integrates testing performed with two GNN models, a *Spatio-Temporal Graph Neural Network (STGNN)* [15] and an *Attention-Temporal Graph Convolution Network (A3T-GCN)* [16].

These two GNN models can provide predictions for the monitored metrics of a cluster's nodes. Both models take as input historical time series data of each node (e.g., CPU or memory usage) as well as information about the topology of the cluster, so that they can take into account the spatial as well as the temporal dependencies of the nodes and provide predictions about the above parameters in future time steps. These predictions will be fed as input features to the RL models of PDLC-DL to improve their performance and help them to provide an improved pod allocation plan to SWM. In addition, the predicted parameters can be fed as input to the

²³https://gitlab.eclipse.org/eclipse-research-labs/codeco-project/privacy-preserving-decentralised-learning-and-context-awareness-pdlc/decentralised-learning/model-selection-and-training/-/tree/main/ICOM-GNNs?ref_type=heads

ACM component to provide insight into the future resource usage of the nodes and allow SWM to make informed decisions and trigger adjustments to the initial setup.

The STGNN can predict future values of nodes' metrics based on historical observations, by modelling both spatial and temporal dependencies among nodes. The nodes' topology is mapped into a graph structure and the model consists of a Graph Convolution Layer and a Recurrent Neural Network layer. The Graph Convolution Layer applies graph convolution to the input to get the nodes' representations over time, so that for each time step, a node's representation is informed by its neighbours' representations. The Graph Convolution is computed as in (3).

$$h_i^{l+1} = \sigma(b^l + \sum_{j \in N(i)} \frac{1}{c_{ij}} h_j^l W^l) \quad (3)$$

where $N(i)$ is the set of neighbors of node i , c_{ij} is the product of the square root of node degrees (i.e., $c_{ij} = \sqrt{|N(i)|} \sqrt{|N(j)|}$), and σ is an activation function.

The nodes' representations are computed by multiplying the input features by the node's own weight and then each node's updated value is calculated by aggregating the neighbors' representations and then multiplying the results by the node's weight. The output of the layer is computed by combining the nodes representations. Based on the input, the graph convolution layer produces new tensor that captures the representations of nodes over time. To process the nodes' representations over time, a Recurrent Neural Network layer is utilized, in this case a *Long Short-Term Memory (LSTM)* layer.

Regarding A3T-GCN, this model is an extension of the Temporal Graph Convolutional Network (T-GCN) model and additionally uses an attention mechanism. T-GCN uses a GCN for the spatial aggregation, in order to capture the topological structure of the data and a Gated Recurrent Unit (GRU), in order to capture the temporal features using the time series with spatial features. The T-GCN model takes as input n historical time series data to obtain n hidden states (h) that cover spatiotemporal information: $h_{t-n}, \dots, h_{t-1}, h_t$.

h_t is calculated as $h_t = u_t * h_{t-1} + (1 - u_t) * c_t$, where u_t at time t is the update gate, meaning the factor that controls the degree to which the previous status is brought into the current status and c_t is the memory content stored at time t . u_t is defined as $u_t = \sigma(W_u[f(A, X_t), h_{t-1}] + b_u)$, where $f(A, X_t)$ represents the graph convolution process, with A being the adjacency matrix (graph) and X_t being the input features at time t , and W, b represent the weights and deviations of the training process respectively. Additionally, c_t is calculated as $c_t = \tanh(W_c[f(A, X_t), r_t * h_{t-1}] + b_c)$, where r_t corresponds to the reset gate, which is used to control the degree of ignoring the previous status, and is calculated as $r_t = \sigma(W_r[f(A, X_t), h_{t-1}] + b_r)$.

Moreover, an attention mechanism is utilized, in order to re-weight the influence of historical values and to capture the data variation trends. The hidden states are given as input to the attention model and the weight $a_{t-n}, \dots, a_{t-n}, a_t$ of each

hidden state is calculated by the softmax function, using a multilayer perception. A weighted sum is used to calculate the context vector that captures the variation information and for the output, the forecasting results go through a fully connected layer.

Furthermore, PDLC-DL will use MLOps techniques to deploy these ML models and guarantee that they efficiently receive data from other components and output a multi-objective estimation. The steps in the pipeline will include the processes needed for the (re)-training of the decentralised models. Once the learning process is complete, the CODECO MLOps pipeline will proceed with deploying the trained models to their target environment/ proper Edge nodes. This involves packaging the models, integrating them into the target system, and monitoring their performance and behavior. Model monitoring will help in detecting anomalies, ensuring model fairness and accuracy, and triggering retraining or updating of models, when necessary, which leads to a completely self-training and self-healing Edge-Cloud continuum.

E. NETMA: NETWORK MANAGEMENT AND ADAPTATION

NetMA²⁴ is an advanced network management and adaptation solution designed to streamline the configuration of interconnections, enhancing the flexibility of Edge-Cloud operations. It effectively addresses the integration of inter-networking control, catering to diverse network environments, including fixed, wireless, and cellular networks that are anticipated to be managed by CODECO. Within CODECO, NetMA handles critical aspects such as network softwarization, semantic interoperability, secure data exchange, predictive behavior, and integrated network capability exposure through standard-based mechanisms and K8s APIs. Furthermore, AI/ML techniques are employed to give insights from network events, facilitating closed-loop automation and adaptive control. In the area of network softwarization, NetMA focuses on providing *Function-as-a-Service (FaaS)* to the Edge and automating network resource management to meet specialized service requirements. It actively promotes the integration of diverse networking domains and extends the physical reach of computing facilities, ensuring seamless semantic compatibility among internetworking services. In NetMA, **network exposure** handles the exposure of CODECO networking metrics to other CODECO components (e.g., ACM, SWM). The exposure will be handled periodically and may also be handled on-demand. Internal NetMA components, such as the Secure Connectivity component will also request specific data from this component.

The network exposure module is expected to provide state information at a link level, state information at a path level. Examples of parameters considered in NetMA are provided in Table 1.

²⁴<https://gitlab.eclipse.org/eclipse-research-labs/codeco-project/network-management-and-adaptation-netma>

TABLE 1. Minimum subset of network metrics expected to be made available by NetMA.

ID (attribute)	Description
node_name	Identifier of the node in K8s
link_id	Identifier of the link of a node
link_failure	Number of failures of a link
node_net_failure	Sum of elink_failure and link_failure
ebw	Egress bandwidth for a node - sum of bw for all Egress links on that node
ibw	Ingress bandwidth for a node - sum of bw for all ingress links on that node
uid_visits	Number of visits from a UE (or UID) to the node. May be relevant in case of mobile environments, UID visits to a node, for instance. Can be based on nr of visits x duration of visit.
geolocation	location of the UE (geo-coordinates translated to string)
zone	Associated to an inter or intra-routing approach; inter, AS
node_degree	In a network, the degree of a node is defined as the sum of all Edges connected to it (later, ingress and egress links)
latency	average latency derived from elink latency; can be based on RTT
path_length	Number of hops traversed; Average Shortest Path Length
link_energy	The average expenditure of energy over a time window (EMA) for a specific link

The **network state management** subcomponent component focuses on monitoring network data using a network performance probe. The probe is designed to measure the proposed network metrics (rf. to Table 1).

The **NetMA MEC Enablement** brings the possibility to integrate data derived from far Edge devices and non-K8s systems, by providing an integration with the ETSI *Multi-Access Edge Computing (MEC)* APIs.

An example usage scenario is as follows. User DEV requests via the CODECO ACM the installation of a distributed application containing multiple micro-services across the Edge-Cloud continuum. The request includes information about the MEC APIs that the application wants to use. CODECO provides an optimal operating environment (cluster, multi-cluster) for the application to run, placing the different micro-services across the far Edge-near Edge Cloud. CODECO allows the micro-services running on the far Edge to use the requested MEC APIs that exist on the MEC platforms on specific near Edge nodes.

One such example is the use of the MEC Location API by a streaming service to perform resource reallocation (e.g., channel bandwidth) to mobile users, depending on their mobility state, to relieve an overloaded antenna. This can be done by reducing the resolution and thus the bandwidth provided to mobile users evaluating the service on a mobile node, e.g., a car.

The **NetMA secure connectivity** subcomponent serves as the primary connectivity mechanism within the context of the project, and is based on L2S-M [17]. In short, L2S-M enables the creation and management of virtual networks in micro-services-based K8s platforms, allowing workloads (or as they are commonly referred, pods) to have secure and isolated link-layer networks. L2S-M achieves this virtual networking

model through a set of *Programmable Link Layer Switches (PLS)* distributed across the platform, which form an overlay network relying on IP tunnelling mechanisms (specifically, using *Virtual Extensible Local Area Networks (VxLANS)*). This overlay of programmable link-layer switches serves as the basis for creating virtual networks using *Software Defined Networking (SDN)*.

To support the fully programmable aspect of the overlay, L2S-M uses an SDN controller to inject the traffic rules into each of the switches, facilitating the implementation of distributed traffic engineering mechanisms across the programmable data plane. For example, priority mechanisms could be implemented in certain services that are sensitive to specific network requirements, e.g., latency. Figure 8 provides a representation on the current design of the NetMA secure connectivity.

This design considers a new module that has been incorporated into L2S-M to address the first of the above lines (i.e. collecting performance overlay information), referred to as *L2S-M Performance Measurements (LPM)*. This module is designed to flexibly and automatically collect performance metrics of the connectivity provided by L2S-M within a single K8s cluster. To achieve this, LPM performs a comprehensive network performance profiling of the overlay network, taking into account various network performance metrics (e.g., available bandwidth, end-to-end delay, etc.). LPM then facilitates the publication of the collected metrics via its LPM Collector component and a dedicated HTTP endpoint within the cluster. All details of the development of LPM are carefully documented in the CODECO repository.²⁵

The operation represented is as follows:

- 1) L2S-M collects different overlay network performance metrics through the LPM module (single cluster overlay information). This information is necessary for the internal operations of L2S-M, as well as to provide it to the SWM component.
- 2) The sCCO discovers the overlay network topology leveraging the L2S-M SDN controller and receives the performance metrics from the LPM Collector.
- 3) Then, the sCCO uses its plugins to expose relevant information to other CODECO components in the form of CRs. In particular, the overlay network topology, and its performance metrics are provided to the SWM to determine the network path selections.
- 4) In addition, the sCCO processes and handles requests (in the form of CRs) from other CODECO components. For instance, the SWM requests the creation of a virtual network to connect two different pods. The request specifies the network path to be used and QoS demands through the appropriate CRs.
- 5) To create the network path, the sCCO installs at every PLS involved (through its control network) the appropriate traffic-flow rules using its SDN controller.

- 6) The sCCO confirms the creation of the network path and the QoS demands. Eventual QoS situations (e.g., link-congestion) are notified to SWM using the appropriate CR.

VII. ORCHESTRATION CHALLENGES AND CODECO ANSWERS

The overall aim of CODECO is to contribute to a smoother and more flexible support of services across the Edge-Cloud continuum via the creation of a novel, cognitive Edge-Cloud management framework. The focus is on a smarter management of highly distributed environments based on heterogeneous networks and integrating mobile, resource-constrained devices. This section goes over the main challenges detected with container orchestrators, and how CODECO answers such challenges.

A. AUTOMATED EDGE CONFIGURATION AND CONTEXT-AWARE MANAGEMENT

K8s and similar orchestrators base their management operations on a set of scripts/playbooks defining/running the steps required to achieve a given desired status, be it infrastructure or application configurations. More recently, with the K8s and operators' trend, this is being replaced by a declarative model, where the desired status is defined and then a set of controllers oversees all the required actions to make the real status match the intended one. This is already a common pattern in K8s or OpenShift environments. There are different tools available to write operators and enhance the K8s APIs with them. But there is still a need for multiple clusters, as well as for cooperation/synchronization between them. To increase the degree of automation CODECO integrates via its component ACM novel automated management mechanisms capable of supporting the setup and application runtime workflow in a way that integrates adaptation of the infrastructure and node resources reducing human intervention. To bring a holistic approach where the infrastructure integrates computational, networking and data observability resources, CODECO introduces the CODECO Application Model (CAM). The CAM is a model based on YAML that captures application and user requirements from a node, network and computational perspective during the application setup. This is essential to allow CODECO to define an optimal and flexible architecture for a specific application and its interconnected micro-services. Additionally, the CAM provides the end-user with status information about the infrastructure and the application status, based on information monitored by CODECO components.

B. LACK OF CONTEXT-AWARENESS AND LIMITED VIEW OF THE INFRASTRUCTURE

Adaptive processes in the context of Edge-Cloud environments supporting heterogeneous mobile devices require context-awareness. Context can be derived from different indicators, e.g., application goals and requirements; node and network resources; surrounding environment aspects (e.g.,

²⁵<https://gitlab.eclipse.org/eclipse-research-labs/codeco-project/network-management-and-adaptation-netma/secure-connectivity>

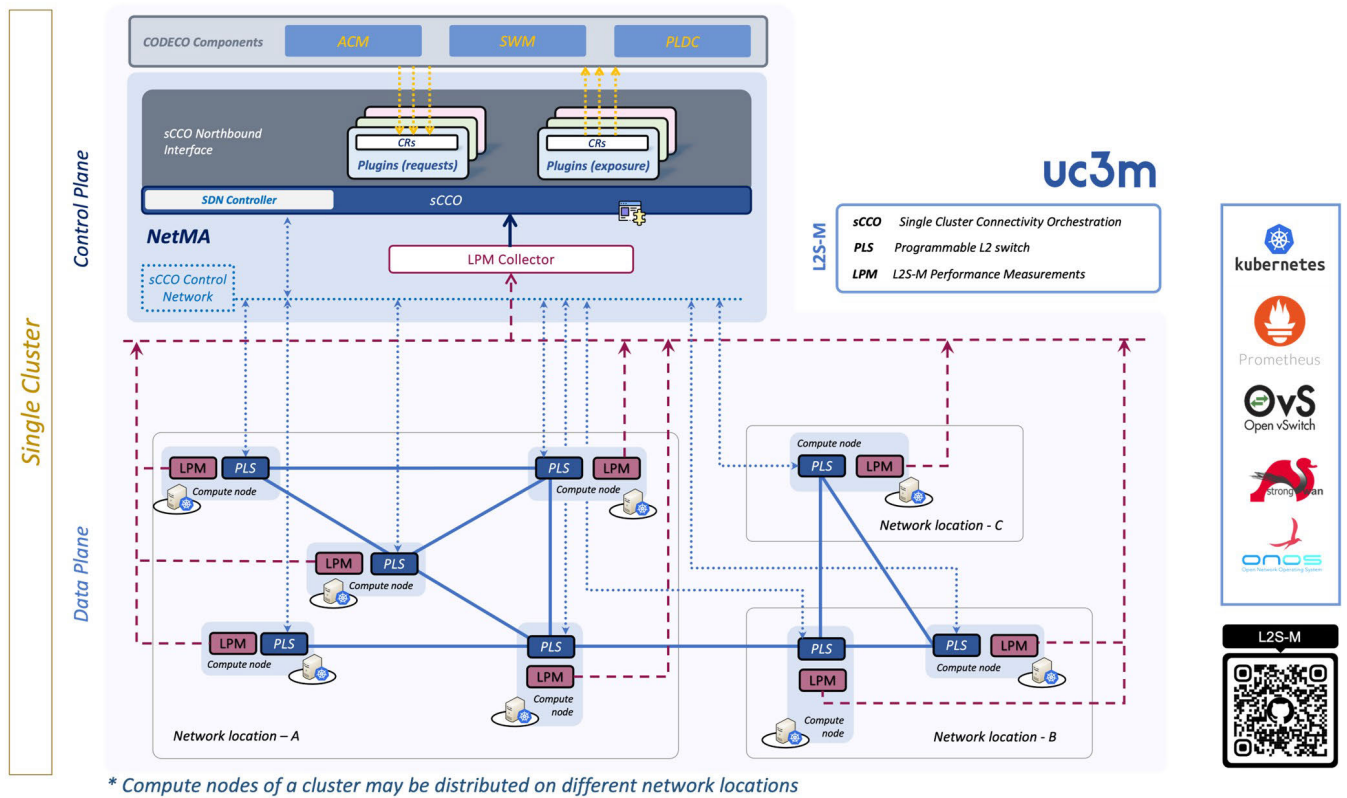


FIGURE 8. Example of secure connectivity provided by NetMA considering a single cluster deployment. NetMA and its secure connectivity approach are described in section VI-E.

location, nearby nodes, etc.). The integration of context-awareness into the network and applications to allow for adaptation has been so far handled in an ad-hoc way, often associated to the specific service to be provided. This adaptive process requires distributed behaviour learning and inference techniques that can support decentralization across Edge-Cloud while preserving the privacy of the raw training data. Relevant in this context is the analysis and proposal for hybrid FL that addresses the Edge-Cloud continuum as a multi-layer, cluster-based structure. Current orchestrators miss a holistic perspective of the application (and its micro-services) to be deployed and managed. It also requires a detailed perspective of the overall infrastructure which relates with data, computational and networking. In contrast, orchestrators focus on the computational aspects only. A flexible management requires the integration of more knowledge on the application, infrastructure, as well as situation/environment. However, by integrating more knowledge, the complexity of selecting an optimal graph to deploy or re-deploy the application workload. To address this complexity, CODECO proposes to consider a specific set of data, network, compute metrics that are sufficient to capture the overall infrastructure status from a network perspective, data observability perspective and computational perspective, so beyond the current orchestration approach. To reduce the complexity of injecting a heavy list of parameters directly into the scheduling process, CODECO considers a meta-data

aggregation approach (component PDLC) which provides nodes with a cost associated with optimization target profiles proposed by the user, for instance, greenness, resilience. This data aggregation considers data observability, network, and computational metrics monitored by CODECO components, or already available in the K8s ecosystem.

C. CROSS-LAYER ADAPTIVE WORKLOAD MIGRATION

K8s scheduling approach focus on the optimization of the infrastructure from a computational perspective. K8s today supports aspects such as autoscaling or load-balancing. In the context of the CODECO SWM component CODECO is advancing the scheduling via a novel approach for graph optimization, which considers the infrastructure as a set of computational, networking, and data workflow resources. Moreover, the monitored metrics in CODECO together with AI/ML are used in the component PDLC to provide SWM with additional information on the stability of the overall infrastructure. PDLC is studying different AI/ML approaches and in particular approaches focusing on federated clusters (decentralised AI), to assist in providing the CODECO scheduler (or any scheduler capable of accepting new metrics) in making a weighted (informed) decision about the placement. This analysis took into consideration AI/ML approaches capable of providing privacy preservation aspects - however, this aspect will become more relevant in federated

cluster environments, which is a future aspect to be developed in CODECO.

D. AI APPLICATION IN ORCHESTRATION

Adaptive processes in the context of Edge-Cloud environments supporting heterogeneous mobile devices benefit from the application of AI. Moreover, considering that the infrastructure is mobile, any adaptive process needs to integrate a way to handle distributed behaviour learning and inference techniques that can support decentralization across Edge-Cloud while preserving the privacy of the raw training data. Relevant in this context is the application of Federated Learning (FL) in Fog computing, and hybrid FL that addresses the Edge-Cloud continuum as a multi-layer, cluster-based structure. Split Learning (SplitNN) is a more recent distributed and private deep learning technique that can be used across edge-cloud devices while improving in terms of scalability and minimizing the need to share raw data directly. Another relevant learning techniques regards Swarm Learning, an immensely powerful technological concept for industrial applications that involve cyber-physical systems, as it can provide flexibility in learning based on the interaction of IT systems, CPS systems and humans. It refers to the deployment and use of specialized AI solutions that mimic the decision making of swarms i.e., solutions that synthesize solutions from decentralized self-organized agents that operate autonomously based on local information. This decentralized operation of swarm intelligence systems obviates the need for centralizing knowledge, thus offering speed, scalability, and potential for devising optimal solutions. These properties are highly desirable in the case of deployment reconfiguration scenarios, where new optimized workflows must be devised over a new or altered configuration of human workers and cyber-physical systems. Swarm intelligence has been used in many different production scenarios in the manufacturing context (e.g., production scheduling) with machines and cyber-physical systems playing the role of swarms (or “modules”) in different granularities. Recently the H2020 MAS4I project which has devised a multi-agent architecture that enables interoperability and collaboration towards realizing autonomous modular production and human assistance. CODECO goes beyond the state of the art by proposing to rely on decentralised AI/ML approaches, such as SplitNN, GNN and Swarm Learning, in considering a cross-layer approach involving parameters collected from the network, application requirements, data models and meta-data compliance aspects and user behaviour. CODECO will support the automated deployment and orchestration of Edge-based services, via the support of elastic models which will be exploited both for the initial setup and for runtime adaptations, while also demonstrating the merits of the swarm intelligence concept for modular and reconfigurable allocation of resources in a cloud/edge environment, towards optimization in complex scenarios where resource allocation develops on both Cloud software and Edge cyber-physical systems. Specifically, CODECO

will design and implement decentralized agents that will directly map to the resource management components of the CODECO Cloud-Edge infrastructure. These modules will be enhanced with decentralized decision support algorithms that consider local information only, while being able to contribute to the global optimization through their participation in the swarm network. To this end, appropriate swarm algorithms (e.g., Ant Colony Optimization) are being explored. Leveraging the self-organizing nature of the swarms and standardized interfaces to Cloud-Edge devices the project will significantly accelerate the (decentralized) optimization of cognitive Cloud reconfiguration use-cases (e.g., optimization of resource allocation in heterogeneous Cloud-Edge scenarios). The automated adaptation provided by CODECO will be developed in the component integrating Context-awareness, Decentralised Learning, and Inference. A summary of the CEI technology enablers handled in CODECO is provided in Table 2.

VIII. EXPERIMENTING WITH CODECO

This section covers current efforts concerning experimentation in CODECO. The section provides input into the CODECO data generator and describes the progress made so far in enabling the integration of CODECO in EdgeNet, as well as the initial efforts to explore the EdgeNet intrinsic features that are relevant to our project and how they can potentially be leveraged to accommodate external experimenters. In addition to EdgeNet, CODECO expects to consider additional experimental infrastructures including a CODECO shared facility (public Cloud based); CODECO partners’ testbeds, as well as the open SLICES²⁶ and CloudLab²⁷ testbeds.

A. CODECO EXPERIMENTATION APPROACH

Figure 9 illustrates the novel CODECO experimentation system and its basic workflow [18]. This system is based on K8s CRDs and customized operators, so the involved controllers do not communicate directly. Instead, they watch the changes of particular custom resources and respond to these updates. The main components of the system are the **Experiment** and the **Infrastructure Controllers**.

The Experiment Controller receives the definition of an experiment in YAML format, which includes details such as the number of replications, benchmarks to execute, parameters of the requested resources, and the application to run in CODECO. Its first step is to communicate with customized Infrastructure Controllers that act as drivers for heterogeneous infrastructures. These controllers communicate with the Experiment Controller via a uniform interface and receive specific resource demands, such as the number of servers for masters and workers of a particular type. They then act upon these requests by communicating with the infrastructures they are responsible for, using technology-specific interfaces to allocate resources and return their configurations, including

²⁶<https://portal.slices-sc.eu>

²⁷<https://cloudlab.us>

TABLE 2. Adaptive CEI technology enablers and answers from CODECO.

Aspect	Description	Where it is applied
Decentralised AI	CODECO is evaluating the role of distributed and split learning approaches; swarm learning; and adaptation of ML models to best serve next generation services needs and constraints.	PDLC expects to validate and rely on decentralized/hybrid federated AI to support the operation of CODECO across federated, multi-tenant environments. Estimations are relevant to SWM (scheduler) and to NetMA (network adaptation).
Edge computing	CODECO overall orchestration approach is based on the architectural principles of Edge and Fog computing.	The overall flexible orchestration of containerized applications reflects the computational abstraction principles of Edge computing.
Cross-layer orchestration	On the one hand, the networking infrastructure needs to become more flexible, intent and context-driven, i.e., the network should be seen and worked upon as being a single system. On the other hand, joint computational and networking paradigms, that integrate, by design, aspects such as security/trustworthiness, mobility support, decentralised and flexible naming spaces are key to build a smart, decentralized Edge ecosystem.	Several CODECO elements monitor the infrastructure (MDM – data; NetMA – network; ACM – compute). ACM provides a cross-layer view on collected metrics to the whole CODECO system. PDLC relies on such metrics to predict infrastructure behaviour. SWM relies on this estimation to weight the decisions of rescheduling applications across Edge-Cloud.
Security and privacy	Privacy and security automation are aspects also addressed in the automation layer of CODECO.	ACM provides privacy and security based on user requirements; PDLC relies on privacy by design in the proposed AI-based orchestration; NetMA provides secure channels between application micro-services via the L2S-M secure connectivity approach.
Cross-layer context-awareness	Context-awareness and behaviour inference shall be considered to integrate into the management of containers aspects such as device usage, application requirements, prediction of mobility patterns, desired levels of privacy and security, to best assist the deployment of applications and services, and to provide a more dynamic network control management.	PDLC relies on context-awareness to provide a tuning of the overall Edge-Cloud environment based on target performance profiles provided by the user, e.g., optimize the overall CEI infrastructure for greenness.
Network probing	CODECO extends the notions of CEI infrastructure to integrate network probing.	NetMA provides network metrics which are used by PDLC and also SWM.
Network management	CODECO provides, via NetMA, secure and adaptive connectivity between micro-services based on the open L2S-M proposal. Network adaptation is supported in NetMA, triggered by requests from SWM and also based on detection of abnormal situations (e.g., high load on a link).	NetMA integrates different sub-components supporting secure connectivity, network monitoring and ML-based adaptation.
FaaS	The locality aspects inherent to Edge computing require an adequate abstraction approach to support operations across Edge-Cloud in a smooth way. Such abstraction is being advocated by the Cloud model of Function-as-a-service (FaaS). However, the higher degree of decentralisation in Edge environments creates challenges to FaaS. For instance, function chaining brings additional overhead; balancing the performance and isolation provided by sandbox environments;	Integrating additional requirements, e.g., data locality, network requirements, application requirements are still aspects to be handled in NetMA, in particular considering the operation of CODECO in federated clusters.
Semantic service composition	CODECO internal and external interfaces consider notions of semantic service composition and aggregation. The cross-layer approach of CODECO relies on semantic service composition principles, present also in K8s.	In CODECO, existing semantic service composition principles of K8s are extended to allow for a notion of application (CODECO Application Model) without jeopardizing the application security and privacy. The user provides application requirements to the system based on a proposed semantic abstraction.
Configuration management	Automated configuration management concerns the installation, configuration and management of multiple distributed and heterogeneous clusters, as well as the mechanisms to deploy and manage applications on top of them.	ACM provides a single point of entry to the CODECO users and reduces the need for human intervention.

IPs and hostnames. All access credentials and SSH keys are exchanged using Kubernetes secrets.

An equivalent process is followed by an EdgeNet Infrastructure Controller that allocates resources in the EdgeNet infrastructure. For example, it can deploy service consumers globally that participate in the benchmarking. Furthermore, we also support the automated installation of our own

EdgeNet instantiation as part of the experiment definition.²⁸

Our next plans include experimenting with multi-cluster EdgeNet installations, such as between regular and CODECO

²⁸<https://gitlab.eclipse.org/eclipse-research-labs/codeco-project/experimentation-framework-and-demonstrations/Edgenet-framework>

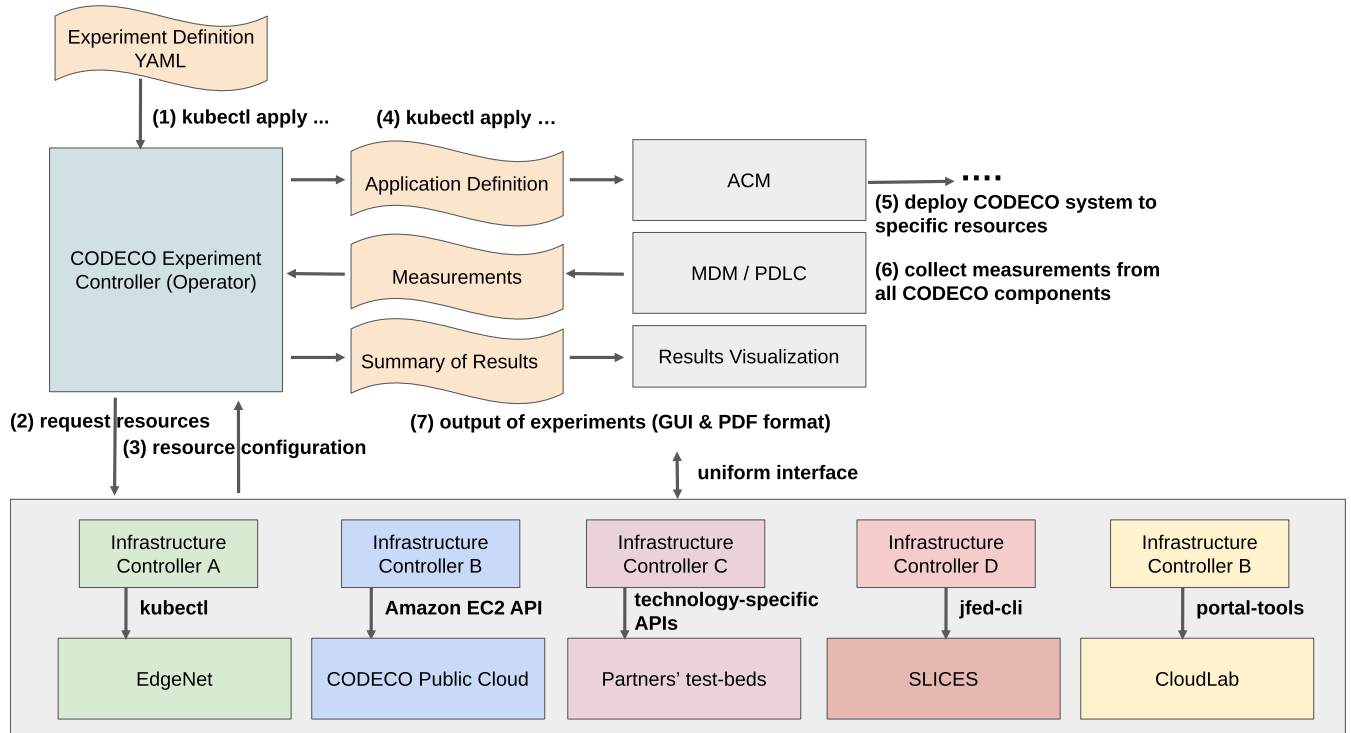


FIGURE 9. The CODECO experimentation framework, described in section VIII-A., and its approach to support experimentation in large-scale environments, e.g., EdgeNet, SLICES, Cloudlab, or dedicated shared cloud testing environments.

EdgeNet deployments, based on the recently introduced federation capabilities of EdgeNet software.

With the resource allocation in place, the Experiment Controller can now form an application definition and pass it to the CODECO ACM. During the application’s operation, other CODECO components, such as PDLC, and MDM, communicate or produce measurements, which are then retrieved from the Experiment Controller. After the completion of the application or the experiment, the Experiment Controller consolidates all the inputs and provides the experiment output in YAML format. This output is then passed to the Results Visualization component, which produces graphs for visualization purposes and generates the output in PDF format.

B. THE CODECO DATA GENERATOR

A major problem in the Edge-Cloud continuum resource orchestration domain is the limited amount of data (or lack thereof) prior to application deployment and execution. To address this issue, a synthetic data generator²⁹ has been implemented. Overall, the CODECO Data Generator (DG) mimics the process of the cross-layer data collection from the CODECO components (ACM, MDM, NetMA), and as output results in a consistent data format for further analysis. In order to evaluate the functionality of the DG, a custom application (sample) is employed, operating within a default

²⁹<https://gitlab.eclipse.org/eclipse-research-labs/codeco-project/experimentation-framework-and-demonstrations/data-generators-and-datasets/synthetic-data-generator>

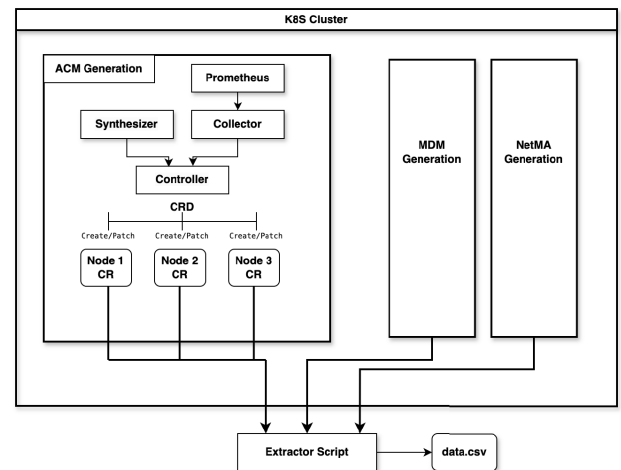


FIGURE 10. The CODECO data generator provides synthetic K8s infrastructure data, based on the CODECO metrics concept and on the resource models for the CODECO monitoring components (ACM, MDM, NetMA).

cross-architecture K8s cluster to generate the necessary data. DG integrates two components. The **DG Collector** gathers values for already defined metrics for which no more calculations are needed, based on the cross-layer attributes provided in the CODECO D11 report [13]. Currently, metrics that are already available in K8s/Prometheus are fed to the data generator, as Prometheus is the basis for the CODECO monitoring aspects and it constitutes a well-suited monitoring solution for Edge-Cloud orchestration. The **DG Synthesizer** is capable of handling composite metrics that cannot be directly

TABLE 3. Comparison of CODECO with related Edge-Cloud orchestration frameworks.

Framework	AC	DSWM	CA-DL	NMA	MDM
MiCADO-Edge [24]	✓	✓		✓	
Sophos [25]	✓	✓	✓	✓	
Oakestra [26]		✓		✓	
DECIDE [27]	✓	✓	✓		✓
HomeCloud [28]				✓	
Apollo [29]		✓			
ORCH [30]		✓			
ECCO [31]		✓			
COSMOS [32]		✓	✓		
LeSO [33]	✓	✓			
Caravela [34]		✓		✓	
BumbleBee [35]			✓	✓	
SODA [36]		✓		✓	
CHARIOT [37]	✓	✓			✓
FogBus2 [38]		✓			
IaaS-SDN [39]				✓	
SDO [40]		✓		✓	
SD-ECC [41]				✓	
SDRM [42]				✓	
CTOSO [43]		✓		✓	✓
RLSK [44]		✓	✓		
KEIDS [45]		✓	✓		
[46]		✓	✓	✓	
[47]		✓			
[48]		✓	✓	✓	
[49]		✓	✓		
[50]		✓	✓		
[51]		✓	✓		
[52]		✓	✓		
CODECO	✓	✓	✓	✓	✓

acquired via the CODECO monitoring architecture. For instance, features such as data freshness (healthiness of the node based on data freshness) may not be directly retrievable from other components. This component will further be explored in alignment with the data aggregation aspects under development in the CODECO PDLC (PDL-C) component. Figure 10 provides an overview on how the DG acts interacts other CODECO components.

IX. RELATED WORK

This section reviews previous work on Edge-Cloud orchestration frameworks, summarizes their functionalities, and compares them with our proposed framework. Table 3 lists the reviewed frameworks and highlights their capabilities with respect to five functionality categories: 1) Automated Configuration (AC), 2) Dynamic Scheduling and Workload Migration (DSWM), 3) Context-Awareness and Decentralized Learning (CA-DL), 4) Network Management and Adaptation (NMA), and 5) Metadata Management (MDM). For further reading on Edge-Cloud orchestration frameworks, we refer the reader to surveys such as [19], [20], [21], [22], and [23].

As can be seen from Table 3, none of the studied frameworks addresses all of the identified feature categories. Furthermore, the majority of the studied frameworks focus on single-cluster orchestration, with only a few explicitly addressing multi-cluster and multi-Cloud scenarios [24], [26], [38], [52]. In-cluster distributed and decentralized solutions are more common [29], [30], [31], [34], [37], [40],

[46]. However, the most ubiquitous setup we found consists of single-cluster centralized orchestration.

In contrast, the CODECO framework considers all five feature categories, from automated configuration to network management and adaptation. The integration of cross-layer context awareness and decentralized learning is particularly novel in CODECO, as is the inclusion of a metadata management layer, which is only considered in a handful of previous works. Furthermore, CODECO follows a decentralized operating paradigm, utilizing decentralized learning and decision making, and its design also considers use in federated cluster scenarios. The following subsections provide a more detailed comparison between CODECO and previous work in each of these categories.

A. AUTOMATED CONFIGURATION

Several works have addressed the issue of automated application configuration, deployment and customization in different ways in the reviewed literature. For example, DECIDE proposes a manager component that provides a test environment to simulate different infrastructure and application deployment scenarios [27]. On the other hand, CHARIOT includes a custom language to model configuration information as a set of constraints and a finite look-ahead algorithm to compute the optimal configuration settings [37], while LeSO manages the deployment of micro-services as sub-slices at the Edge [33].

Other frameworks have taken approaches that are coupled with specific container orchestration tools, such as K8s. In this sense, Sophos extends the K8s control plane with a controller that periodically updates the application configuration graph with the set of inter-pod affinity rules between the application micro-services [25]. On the other hand, MiCADO-Edge automatically deploys complex sets of interconnected micro-services using KubeEdge,³⁰ an open-source orchestrator that extends Kubernetes clusters to non-Cloud workers [24]. Examples of other Edge-oriented K8s automation facilities include Knative³¹ for rapid event-driven scalability and serverless deployment, KCP³² for multi-cluster solutions designed for Edge-Cloud deployments, and Flotta³³ to meet the stringent requirements of even more constrained environments such as the far Edge.

Despite recent advances in multi-cluster, Edge-focused automated configuration systems, most approaches only partially address the challenges posed by the Edge-Cloud continuum, such as providing seamless end-to-end connectivity, handling device and communication protocol heterogeneity, and effectively dealing with scalability and elasticity issues.

CODECO aims to provide robust provisioning, configuration and synchronization mechanisms specifically designed

³⁰<https://kubeEdge.io/en>

³¹<https://knative.dev>

³²<https://github.com/kcp-dev/kcp>

³³<https://project-flotta.io>

for federated cluster environments spanning the Edge-Cloud continuum. It will do this by building on several existing technologies (e.g., OCM, Flotta, KCP, Knative) and extending their capabilities to efficiently support multi-clustering and resource management at the Edge. In addition, CODECO aims to explicitly cover novel Cloud-to-Edge use-cases, addressing a variety of challenges not previously considered in a single framework. To evaluate the ability of the CODECO framework to address the specific requirements of such use-cases, the technical advances provided by CODECO, including ACM performance, will be extensively evaluated in real-world experiments.

B. ADAPTIVE SCHEDULING AND WORKLOAD MIGRATION

CODECO provides adaptive scheduling, workload orchestration and migration via the SWM component, which specifically addresses synchronization issues in multi-cluster environments.

Existing scheduling and workload migration mechanisms address synchronization issues in single-cluster environments, focusing on specific aspects [53]. For example, the Intel Telemetry Aware Scheduler (TAS) [54] supports telemetry-aware scheduling and intelligent workload placement in Kubernetes, enforcing a user-defined telemetry policy based on computing node health metrics. The lightweight Kubernetes-based Event Driven Autoscaler (KEDA)³⁴ allows pods to be invoked based on external events, extending the native autoscaling capabilities of K8s. KubeSphere³⁵ is a well-known scheduling mechanism in hybrid multi-Clouds that dispatches tasks to connected K8s clusters based on custom policies and fairness goals, eliminating the need to hold tasks for later scheduling [55].

Current scheduling solutions are also limited by the lack of network awareness in scheduling decisions [56]. The K8s network-aware scheduler plugin addresses this issue by enabling latency- and bandwidth-aware pod scheduling that considers both the application and infrastructure network topology. It establishes network weights between regions and zones to reduce latency. However, the scheduler has known limitations, including the lack of a dedicated controller (such as the network-topology-controller project³⁶) to handle bandwidth allocation and update network weights based on real-time latency measurements. It also introduces a custom plugin that cannot be combined with other plugins accessing the same extension point, potentially leading to blocking decisions and deadlocks in sequential pod scheduling. As of now, Seamless Computing is based on a comprehensive QoS model [57] that considers application requirements and infrastructure capabilities (computing, storage, network) to optimize the deployment of distributed applications across the Edge-Cloud.

When considering federated clusters, additional challenges need to be addressed. Multi-cluster systems currently lack concrete co-scheduling mechanisms, and only recently have new synchronization mechanisms been proposed. These include the K8s Sigs co-scheduling plugin³⁷ (which is in beta status), Admiralty,³⁸ the k8s-spark-scheduler³⁹ (no longer maintained), and recent research efforts such as RLSK [44] and Twine [58].

CODECO extends the notion of seamless computing [59] through advanced scheduling, workload orchestration, and migration, particularly in federated clusters. It incorporates different categories of data (data-computing-network) and context-awareness into the scheduling loop. For the network-awareness integration, CODECO follows the network-aware scheduler probing proposal for bandwidth and latency as a starting point [60]. It integrates the estimation provided by PDLC (ML/DL decentralized, on-demand approach) to enable well-informed scheduling and migration decisions. Finally, it supports workload migration in challenging cases, such as highly heterogeneous environments with cluster-specific requirements, mobile networks with intermittent connectivity, and scenarios with mobile far Edge nodes that require automated remote reconfiguration of computing and data processing modules.

C. CONTEXT-AWARENESS AND DECENTRALIZED LEARNING

In dynamic and heterogeneous CEI environments, the use of context information in the orchestration decisions faces significant challenges, such as limited indicators, diverse interconnected devices and data types, and resource constraints [61]. The most common context-aware approaches in orchestration frameworks have been network and resource-aware [25], [45], [49] and application-aware [35], whereas additional context indicators have been proposed in [61]. For more detailed discussions, multiple surveys have been elaborated on the status of integrating context-awareness into the Edge-Cloud continuum [8], [62], [63]. It is relevant to highlight the recent effort of the EUCloudEdgeIoT (EUCEI)⁴⁰ initiative in creating a reference architecture for the Edge-Cloud continuum, based on input and efforts under development in European projects, in association with topics such as cognitive computing, meta-OS, swarm computing. In CODECO, which is one of the projects contributing to the creation of the EUCEI reference architecture, the main differentiator is the use of combined sensing approaches to integrate context (combined heuristics derived from network, data observability and computing metrics regularly monitored by CODECO), thus increasing flexibility by considering a cross-layer metrics approach, while at the same

³⁴<https://keda.sh>

³⁵<https://www.kubesphere.io>

³⁶<https://github.com/jpedro1992/network-topology-controller>

³⁷<https://github.com/kubernetes-sigs/scheduler-plugins/tree/master/pkg/coscheduling>

³⁸<https://github.com/admiraltyio/admiralty>

³⁹<https://github.com/palantir/k8s-spark-scheduler>

⁴⁰<https://eucloudedgeiot.eu/>

time providing a way that can support scalability, assuming the use of different metrics.

However, the integration of context into the orchestration process based on data aggregation is not a trivial process. The optimal placement optimization would require working with all of the feasible combinations of metrics in different categories (data, network, computing), which from a time and processing complexity perspective is not suitable for the real-time operation required in the Edge-Cloud. Moreover, different situations may require different approaches to handle the combined context data aggregation. For instance, in mobile scenarios the relevancy of networking metric variations is more relevant than in scenarios where nodes are static.

AI/ML approaches are therefore crucial to assist in balancing the system performance, and understanding if the proposed combined context-awareness can be beneficial across different scenarios.

In the adaptive provisioning process, reconfiguration has been studied to improve orchestration decisions. Several techniques have been proposed [27], [32], [47], [48], with the most common being *reinforcement learning (RL)* [44], [50], [51]. Distributed behavioral learning and inference techniques that can support decentralization across the Edge Cloud while preserving the privacy of the raw training data, such as Federated Learning and Decentralized Learning, have also been considered [46], [52]. The benefits of applying decentralized learning include reduced latency and bandwidth consumption, distributed and asymmetric model training, enhanced security/privacy, and efficient computational load distribution at the Edge. Several related literature provides more details about the challenges, opportunities, and benefits of applying distributed learning in Edge-Cloud environments [64], [65], [66].

CODECO's PDLC component supports a variety of cross-layer input parameters, elastic models adaptable to context changes, and joint orchestration of data, computing, and network resources. The variety of parameters collected and their integration into aggregated performance metrics distinguishes CODECO from previous work focused on either the network or application layer. In addition, PDLC includes a decentralized learning framework that exploits the collected input parameters and focuses on: 1) adaptive processes that consider the requirements and restrictions of all CODECO components, 2) cross-layer intelligence without the need to share raw data between different nodes and clusters, and 3) privacy-preserving intelligence across the Edge-Cloud continuum, complementing existing works [46], [52].

D. CROSS-LAYER NETWORK MANAGEMENT AND ADAPTATION

CODECO proposes the NetMA component to handle the K8s underlay network for application deployments across the Edge-Cloud continuum, providing: (i) network connectivity and communication using K8s *Container Network*

Interface (CNI) plugins; (ii) interconnection of diverse Edge-Cloud environments and across federated clusters; (iii) exposure/provisioning of network information towards other CODECO components based on the ALTO protocol; and (iv) integration of AI/ML techniques to predict network behavior.

Cluster networking is based on the K8s CNI plugins that provide overlay or underlay networking capabilities for pod communication. Several plugins are available, including Calico,⁴¹ Flannel,⁴² Weave,⁴³ Cilium,⁴⁴ Canal⁴⁵ Antrea,⁴⁶ Kube-OVN,⁴⁷ OVN-K,⁴⁸ and Multus,⁴⁹ which implement different network models (i.e., overlay, underlay or hybrid), use different tunneling options (e.g., VXLAN, IPsec, GRE, or Geneve), and offer additional features, such as multi-casting, encryption, IPv6, IPVS/LVS, bridging and eBPF. A number of performance comparisons, e.g., [67], document that Flannel and Weave produce the lowest overhead, while Calico, Cilium, and OVN offer advanced features at the expense of overhead.

Other solutions, such as Skupper⁵⁰ and Submariner,⁵¹ implement direct networking capabilities between K8s clusters. Skupper provides bi-directional communication and service discovery between clusters, while Submariner provides secure connections, unified IP address spaces, and network routing between clusters. They also support service meshes such as Istio⁵² and Consul.⁵³ Service meshes are based on "sidecar" proxies and control service-to-service communications, handle network traffic and enforce policies between those services.

In addition, a critical problem is the lack of sufficient knowledge about the underlying network infrastructure. The Application-Layer Traffic Optimization (ALTO) [68] protocol provides a standardized framework for exposing network information, such as network topology and link bandwidth, to components and applications. ALTO uses key abstractions known as network and cost maps.

CODECO plans to use the OVN-K networking plugin, which is also used by Microshift, a lightweight K8s distribution. The plugin is built on top of the OVN networking backend and provides an overlay-based implementation using the Geneve protocol. It also provides K8s-specific APIs for efficient management of network traffic. In addition, we also consider the Link-Layer Secure connectivity for micro-service platforms (L2S-M)⁵⁴ as a starting point. L2S-

⁴¹<https://github.com/projectcalico/calico>

⁴²<https://github.com/flannel-io/flannel>

⁴³<https://github.com/weaveworks/weave>

⁴⁴<https://github.com/cilium/cilium>

⁴⁵<https://github.com/projectcalico/canal>

⁴⁶<https://github.com/antrea-io/antrea>

⁴⁷<https://github.com/kubeovn/kube-ovn>

⁴⁸<https://github.com/ovn-org/ovn-kubernetes>

⁴⁹<https://github.com/k8snetworkplumbingwg/multus-cni>

⁵⁰<https://skupper.io>

⁵¹<https://submariner.io>

⁵²<https://istio.io>

⁵³<https://github.com/hashicorp/consul-k8s>

⁵⁴<https://github.com/Networks-it-uc3m/L2S-M>

M is a Kubernetes operator that enables virtual networking using an SDN-based data plane. It allows the creation of network paths based on different algorithms, such as reactive, metric-based, or geographic-based approaches [69]. NetMA aims to automate the interconnection of diverse Edge-Cloud environments, including wireless, cellular, and fixed networks, by providing automated handling of the interconnection process. In addition, it will expose network information such as topology and link bandwidth through extensions to the ALTO protocol. Finally, NetMA will leverage AI/ML techniques to predict the behavior of network KPIs and facilitate real-time adjustments.

E. METADATA MANAGEMENT

Very few orchestration frameworks in the reviewed literature explicitly address the issue of metadata management. DECIDE proposes a central repository to store infrastructure and application metadata as well as monitoring data, as well as monitoring data [27]. Similarly, the CHARIOT architecture includes a data storage layer that provides a generic and unified system state, extended with replication to avoid single points of failure [37]. A different approach is taken by CTOSO, which exposes the metadata collected by IoT devices as services in the uplink transmission to the Cloud via MEC servers [43].

Outside of the realm of orchestration frameworks, novel metadata management concepts such as data mesh [70] and data fabric [71], [72] have recently emerged to mitigate the problems of inflexibility to changing requirements and scalability of traditional storage solutions such as data warehouses and data lakes. They operate in a distributed manner, processing data locally and publishing it to centralized metadata catalogues, and can provide critical support for heterogeneous data from distributed environments and across clusters and different organizations. In addition, advanced Cloud service catalogues, such as Gaia-X [4], provide secure data sharing at scale based on real-time data observability with connectors to local distributed data systems, providing significant privacy, scalability, and performance benefits by collecting only metadata centrally, not data.

CODECO aims to advance metadata management in orchestration frameworks beyond the state of the art by leveraging data mesh and data fabric concepts to address the management challenges associated with security, privacy, regulation, and decision making based on cross-layer data collected from various distributed components. This is achieved via the MDM component, which includes a real-time updated graph that captures diverse and extensible sources of metadata (i.e., an enterprise data map). MDM provides an extensible connector model where multiple connectors can report information about the same data set. Connectors can also integrate directly with data stores to discover existing records and data structure, or interface with systems that analyze records to automatically determine data characteristics (e.g., quality, sensitivity). CODECO

connectors can be deployed across the CEI continuum, supporting orchestration decisions at the multi-cluster level and addressing data issues (performance, compliance).

X. CONCLUSION, IMPACT OF CEI EMERGING TRENDS, AND FUTURE RESEARCH DIRECTIONS

This paper presents the CODECO orchestration framework, which is being developed to support the deployment of applications across Edge-Cloud in a way that can truly embrace the notion of IoT infrastructure, considering three axes: computing, network, and data observability. The paper presents use-cases that explain the benefits CODECO can bring to various competitiveness sectors such as manufacturing, smart cities, mobility, and energy. It then presents the current CODECO framework and its components.

Based on the description of CODECO and a thorough comparison with related work, the CODECO enhancements can be summarized as follows:

- **Automated configuration**, focusing on supporting application setup and application run-time across Edge-Cloud, by considering computing, network, and data observability aspects.
- **Data as a resource**. CODECO addresses data as a resource in the sense that available snapshots from the overall Edge-Cloud infrastructure, integrating different perspectives (application, user, system, data, network) at different instants of the CODECO operational workflow can be provided to different CODECO components, to assist in preventing, during placement of applications, aspects such as lack of data compliance.
- **Dynamic scheduling and workload migration**. CODECO builds on the concept of seamless computing integrating QoS models that consider data-network-computation requirements to provide a best match between applications and available infrastructure (nodes, their computational and data properties, as well as network nodes and links), and to schedule and re-schedule application workloads across single cluster and federated cluster environments, considering application and user requirements.
- **Context-awareness and privacy preserving decentralized learning**. CODECO relies on context-awareness to be able to achieve a joint data-network-computing orchestration, and on privacy-preserving decentralized learning and inference to best support readjustment of aspects such as the processing capability, computational resources, networking resources and interconnections in real-time.
- **Infrastructure adaptation based on a cross-layer data-computing-network approach**. CODECO provides exposure of networking metadata via the ALTO protocol [73], and assists in adapting not just computational (node resources) but also the networking infrastructure interconnecting such nodes, via an OSI Layer 2 encapsulation approach.

As has been referred throughout the paper, there is an ongoing evolution and convergence of IoT, Edge and Cloud technologies to meet the increasing demands of next generation IoT applications across different sectors. A key initiative in this context is EUCEI, which is providing a way for emerging trends to converge, and for different solutions and projects to adapt by defining taxonomy, steps, and models towards a reference CEI architecture [73]. Different projects, such as CODECO, are currently contributing to the development of this architecture, and therefore aligning with the emerging CEI trends in an agile way.

Although ambitious, CODECO currently provides a set of open-source tools that allow the research community to further explore the proposed concepts and embrace the idea of flexible, cross-layered, and cognitive orchestration.

The deployment of a framework such as CODECO is not trivial and poses significant challenges to the overall flexible Cloud-Edge-IoT orchestration, in particular considering multi-tenant, federated environments. This is currently the next step in the CODECO research development, which brings several challenges that can guide future research.

First and foremost, it is essential to define an application abstraction model that can be easily adapted and translated into the orchestration engine as a set of application requirements. This is currently embodied in the CAM concept in CODECO, which needs to be further refined in order to bring application requirements into the different CODECO components. Moreover, in federated environments, the CAM model will regularly get information from CODECO components across different clusters. This requires a semantic hierarchical design that can scale across hundreds of clusters, and eventually involving thousands of heterogeneous nodes. A second challenge is the integration of privacy-preserving decentralized learning in a mobile and heterogeneous Edge-Cloud environment. While for a single cluster operation, federated learning patterns are suitable, when addressing federated cluster environments, the need for a decentralized AI/ML pattern increases. Swarm learning may be relevant in this context; however, the use of Ledger technologies brings significant weight which may not be compatible with the operation of CODECO across far Edge to Cloud. These aspects are currently being analyzed in the CODECO PDLC component. A third challenge relates to the use of a more complex set of metrics at different levels of the OSI layer model (data observability, computing, network), with their regular monitoring and injection into the CODECO components. The different metrics have different polling periods; changes must be regularly monitored and communicated to the different CODECO components. This is currently under the supervision of the CODECO monitoring architecture, which is part of the ACM. In addition, the more parameters that are taken into account, the higher the complexity associated with the optimization of the application workload placement. CODECO addresses this situation by considering methods to combine metrics based on specific target profiles specified by the user, e.g.,

energy efficiency, resilience. By considering the combination of metrics (currently in PDLC), it is possible to reduce the weight of the overall placement optimization process (supported by SWM). On the other hand, the combination of metrics may reduce the fine-grained tuning of the placement. Therefore, it is important to analyze different approaches to combine metrics based on specific target profiles in future work. A fourth challenge relates with the approach followed in SWM to place the application workload. SWM relies on a solver. The convergence times in federated environments need to be tested, and an analysis on the waiting times of the solver to deploy workloads needs to be analysed under different conditions.

ACKNOWLEDGMENT

The authors would like to thank all members of the CODECO consortium for their valuable contributions to CODECO deliverables D8, D9, and D11.

REFERENCES

- [1] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An overview on edge computing research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020.
- [2] J.-H. Huh and Y.-S. Seo, "Understanding edge computing: Engineering evolution with artificial intelligence," *IEEE Access*, vol. 7, pp. 164229–164245, 2019.
- [3] H. Tardieu, "Role of Gaia-X in the European data space ecosystem," in *Designing Data Spaces: The Ecosystem Approach To Competitive Advantage*. Cham, Switzerland: Springer, 2022, pp. 41–59.
- [4] A. Braud, G. Fromentoux, B. Radier, and O. Le Grand, "The road to European digital sovereignty with GAIA-X and IDSA," *IEEE Netw.*, vol. 35, no. 2, pp. 4–5, Mar. 2021.
- [5] E. Casalicchio, *Container Orchestration: A Survey* (Systems Modeling: Methodologies and Tools). Cham, Switzerland: Springer, 2019, doi: [10.1007/978-3-319-92378-9_14](https://doi.org/10.1007/978-3-319-92378-9_14).
- [6] R. C. Sofia, D. Dykemann, P. Urbanetz, A. Galal, and D. Dave, "Dynamic, context-aware cross-layer orchestration of containerized applications: A perspective," *IEEE Access*, vol. 11, pp. 93129–93150, 2023.
- [7] R. C. Sofia et al., Jul. 2023, "CODECO D9—Technological guidelines, reference architecture, and initial open-source ecosystem design v1.0," *Zenodo*, doi: [10.5281/zenodo.8143860](https://doi.org/10.5281/zenodo.8143860).
- [8] R. C. Sofia and J. Soldatos, 2021, "A vision on smart, decentralised edge computing research directions," *Zenodo*, doi: [10.5281/zenodo.5837299](https://doi.org/10.5281/zenodo.5837299).
- [9] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *Proc. USENIX Conf. USENIX Annu. Tech. Conf.*, 2014, pp. 305–320.
- [10] O. Ben-Kiki, C. Evans, and B. Ingerson. (2009). *Yaml Ain't Markup Language (YAML) Version 1.2*. [Online]. Available: <https://yaml.org/spec/1.2/>
- [11] R. Sousa, J. Soldatos, D. Matzakou, X. Fu, L. C. Murillo, D. Jimenez, Jordi-Marias, R. C. Sofia, and A. Stam, Jul. 2023, "CODECO D8—Fine-grained use-case design and business impact," *Zenodo*, doi: [10.5281/zenodo.8143800](https://doi.org/10.5281/zenodo.8143800).
- [12] *IoT and Edge Computing Impact on Beyond 5G: Enabling Technologies and Challenges Release 3.0*, AIOTI WG Standardisation, Brussels, Belgium, 2013.
- [13] N. Psaromanolakis et al., Oct. 2023, "CODECO D11—CODECO basic operation and open toolkit v1.0," CODECO Deliverables Series, *Zenodo*, doi: [10.5281/zenodo.10148462](https://doi.org/10.5281/zenodo.10148462).
- [14] S. Rooney, L. Garcés-Erice, D. Bauer, and P. Urbanetz, "Pathfinder: Building the enterprise data map," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Orlando, FL, USA, Dec. 2021, pp. 1909–1919.
- [15] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.
- [16] J. Bai, J. Zhu, Y. Song, L. Zhao, Z. Hou, R. Du, and H. Li, "A3T-GCN: Attention temporal graph convolutional network for traffic forecasting," *MDPI ISPRS Int. J. Geo-Inf.*, vol. 10, no. 7, 2021, Art. no. 485, doi: [10.3390/ijgi10070485](https://doi.org/10.3390/ijgi10070485).

- [17] L. F. Gonzalez, I. Vidal, F. Valera, R. Martin, and D. Artalejo, "A link-layer virtual networking solution for cloud-native network function virtualisation ecosystems: L2S-M," *Future Internet*, vol. 15, no. 8, p. 274, Aug. 2023, doi: [10.3390/fi15080274](https://doi.org/10.3390/fi15080274).
- [18] G. Koukis, S. Skaperas, I. Angeliki Kapetanidou, V. Tsaoussidis, and L. Mamatas, "An open-source experimentation framework for the edge cloud continuum," in *Proc. Comput. Netw. Experim. Res. Testbeds Workshop (CNERT)*, 2024, doi: [10.48550/arXiv.2403.10977](https://doi.org/10.48550/arXiv.2403.10977).
- [19] A. Ullah, T. Kiss, J. Kovács, F. Tusa, J. Deslauriers, H. Dagdeviren, R. Arjun, and H. Hamzeh, "Orchestration in the cloud-to-things compute continuum: Taxonomy, survey and future directions," *J. Cloud Comput.*, vol. 12, no. 1, 2023, Art. no. 135.
- [20] P. Muzumdar, A. Bhosale, G. P. Basyal, and G. Kurian, "Navigating the Docker ecosystem: A comprehensive taxonomy and survey," *Asian J. Res. Comput. Sci.*, vol. 17, no. 1, pp. 42–61, 2024.
- [21] J. Ren, D. Zhang, S. He, Y. Zhang, and T. Li, "A survey on end-edge-cloud orchestrated network computing paradigms: Transparent computing, mobile edge computing, fog computing, and cloudlet," *ACM Comput. Surv.*, vol. 52, pp. 1–36, Oct. 2019.
- [22] B. Costa, J. Bachiega, L. R. de Carvalho, and A. P. F. Araujo, "Orchestration in fog computing: A comprehensive survey," *ACM Comput. Surv.*, vol. 55, no. 2, pp. 1–34, Jan. 2022, Art. no. 29, doi: [10.1145/3486221](https://doi.org/10.1145/3486221).
- [23] R. Mahmud, K. Ramamohanarao, and R. Buyya, "Application management in fog computing environments: A taxonomy, review and future directions," *ACM Comput. Surv.*, vol. 53, no. 4, pp. 1–43, 2020.
- [24] A. Ullah, H. Dagdeviren, R. C. Ariyattu, J. DesLauriers, T. Kiss, and J. Bowden, "MiCADO-edge: Towards an application-level orchestrator for the cloud-to-edge computing continuum," *J. Grid Comput.*, vol. 19, no. 4, pp. 1–28, Nov. 2021, doi: [10.1007/s10723-021-09589-5](https://doi.org/10.1007/s10723-021-09589-5).
- [25] A. Marchese and O. Tomarchio, "Sophos: A framework for application orchestration in the cloud-to-edge continuum," in *Proc. 13th Int. Conf. Cloud Comput. Services Sci.*, vol. 1, 2023, pp. 261–268.
- [26] G. Bartolomeo, M. Yosofie, S. Bäurle, O. Haluszczynski, N. Mohan, and J. Ott, "Oakestra: A lightweight hierarchical orchestration framework for edge computing," in *Proc. USENIX Annu. Tech. Conf. (USENIX ATC)*. Boston, MA, USA: USENIX Association, Jul. 2023, pp. 215–231. [Online]. Available: <https://www.usenix.org/conference/atc23/presentation/bartolomeo>
- [27] J. M. Kunkel, C. Boehme, J. Decker, F. Magugliani, D. Pleiter, B. Koller, and B. Yaman, "DECICE: Device-edge-cloud intelligent collaboration framework," in *Proc. 20th ACM Int. Conf. Comput. Frontiers*, May 2023, pp. 266–271.
- [28] J. Pan, L. Ma, R. Ravindran, and P. TalebiFard, "HomeCloud: An edge cloud framework and testbed for new application delivery," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–6.
- [29] F. Smirnov, C. Engelhardt, J. Mittelberger, B. Pourmohseni, and T. Fahringer, "Apollo: Towards an efficient distributed orchestration of serverless function compositions in the cloud-edge continuum," in *Proc. 14th IEEE/ACM Int. Conf. Utility Cloud Comput.*, Dec. 2021, pp. 1–10.
- [30] K. Toczé and S. Nadjm-Tehrani, "ORCH: Distributed orchestration framework using mobile edge devices," in *Proc. IEEE 3rd Int. Conf. Fog Edge Comput. (ICFEC)*, May 2019, pp. 1–10.
- [31] V. Cozzolino, J. Ott, A. Y. Ding, and R. Mortier, "ECCO: Edge-cloud chaining and orchestration framework for road context assessment," in *Proc. IEEE/ACM 5th Int. Conf. Internet-of-Things Design Implement. (IoTDI)*, Apr. 2020, pp. 223–230.
- [32] G. Papathanail, I. Fotoglou, C. Demertzis, A. Pentelas, K. Sgouromitis, P. Papadimitriou, D. Spatharakis, I. Dimolitsas, D. Dechouniotis, and S. Papavassiliou, "COSMOS: An orchestration framework for smart computation offloading in edge clouds," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp.*, Apr. 2020, pp. 1–6.
- [33] S. Arora, A. Ksentini, and C. Bonnet, "Lightweight edge slice orchestration framework," in *Proc. IEEE Int. Conf. Commun.*, May 2022, pp. 865–870.
- [34] A. Pires, J. Simão, and L. Veiga, "Distributed and decentralized orchestration of containers on edge clouds," *J. Grid Comput.*, vol. 19, no. 3, p. 36, 2021.
- [35] H. Lee, S. Noghabi, B. Noble, M. Furlong, and L. P. Cox, "BumbleBee: Application-aware adaptation for edge-cloud orchestration," in *Proc. IEEE/ACM 7th Symp. Edge Comput. (SEC)*, Dec. 2022, pp. 122–135.
- [36] Y. Liu, Z. Zeng, X. Liu, X. Zhu, and M. Z. A. Bhuiyan, "A novel load balancing and low response delay framework for edge-cloud network based on SDN," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5922–5933, Jul. 2020.
- [37] S. Pradhan, A. Dubey, S. Khare, S. Nannapaneni, A. Gokhale, S. Mahadevan, D. C. Schmidt, and M. Lehofer, "CHARIOT: Goal-driven orchestration middleware for resilient IoT systems," *ACM Trans. Cyber-Phys. Syst.*, vol. 2, no. 3, pp. 1–37, 2018.
- [38] Q. Deng, M. Goudarzi, and R. Buyya, "FogBus2: A lightweight and distributed container-based framework for integration of IoT-enabled systems with edge and cloud computing," in *Proc. Int. Workshop Big Data Emergent Distrib. Environ.*, Jun. 2021, pp. 1–8.
- [39] F. A. Zaman, A. Jarray, and A. Karmouch, "Software defined network-based edge cloud resource allocation framework," *IEEE Access*, vol. 7, pp. 10672–10690, 2019.
- [40] G. Castellano, F. Esposito, and F. Risso, "A service-defined approach for orchestration of heterogeneous applications in cloud/edge platforms," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 4, pp. 1404–1418, Dec. 2019.
- [41] J. Tang, J. Nie, Z. Xiong, J. Zhao, Y. Zhang, and D. Niyato, "Slicing-based reliable resource orchestration for secure software-defined edge-cloud computing systems," *IEEE Internet Things J.*, vol. 9, no. 4, pp. 2637–2648, Feb. 2022.
- [42] J. Okwuibe, J. Haavisto, I. Kovacevic, E. Harjula, I. Ahmad, J. Islam, and M. Ylianttila, "SDN-enabled resource orchestration for industrial IoT in collaborative edge-cloud networks," *IEEE Access*, vol. 9, pp. 115839–115854, 2021.
- [43] M. Huang, W. Liu, T. Wang, A. Liu, and S. Zhang, "A cloud-MEC collaborative task offloading scheme with service orchestration," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 5792–5805, Jul. 2020.
- [44] J. Huang, C. Xiao, and W. Wu, "RLSK: A job scheduler for federated kubernetes clusters based on reinforcement learning," in *Proc. IEEE Int. Conf. Cloud Eng. (IC2E)*, Apr. 2020, pp. 116–123.
- [45] K. Kaur, S. Garg, G. Kaddoum, S. H. Ahmed, and M. Atiqzaman, "KEIDS: Kubernetes-based energy and interference driven scheduler for industrial IoT in edge-cloud ecosystem," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4228–4237, May 2020.
- [46] Y. Gong, H. Yao, J. Wang, D. Wu, N. Zhang, and F. R. Yu, "Decentralized edge intelligence-driven network resource orchestration mechanism," *IEEE Netw.*, vol. 37, no. 2, pp. 270–276, Jul. 2023.
- [47] C. Sonmez, A. Ozgovde, and C. Ersoy, "Fuzzy workload orchestration for edge computing," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 2, pp. 769–782, Jun. 2019.
- [48] E. Zeydan, J. Mangues-Bafalluy, and Y. Turk, "Intelligent service orchestration in edge cloud networks," *IEEE Netw.*, vol. 35, no. 6, pp. 126–132, Nov. 2021.
- [49] M. C. Ogbuachi, C. Gore, A. Reale, P. Suskovic, and B. Kovács, "Context-aware K8S scheduler for real time distributed 5G edge computing applications," in *Proc. Int. Conf. Softw., Telecommun. Comput. Netw. (SoftCOM)*, Sep. 2019, pp. 1–6.
- [50] T. Zheng, J. Wan, J. Zhang, and J. Jiang, "Deep reinforcement learning-based workload scheduling for edge computing," *J. Cloud Comput.*, vol. 11, no. 1, p. 3, 2022.
- [51] X. Xiong, K. Zheng, L. Lei, and L. Hou, "Resource allocation based on deep reinforcement learning in IoT edge computing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1133–1146, Jun. 2020.
- [52] Y. Qian, L. Hu, J. Chen, X. Guan, M. M. Hassan, and A. Alelaiwi, "Privacy-aware service placement for mobile edge computing via federated learning," *Inf. Sci.*, vol. 505, pp. 562–570, Dec. 2019.
- [53] K. Senjab, S. Abbas, N. Ahmed, and A. R. Khan, "A survey of Kubernetes scheduling algorithms," *J. Cloud Comput.*, vol. 12, no. 1, 2023, Art. no. 87.
- [54] J. Browne, E. Collins, K. Kepka, S. Ranganath, J. K. Kadavathu, S. Sehgal, K. Muldoon, and M. Kobylinski, "Closed loop automation—Telemetry aware scheduler for service healing and platform resilience," Intel, White Paper, 2019. [Online]. Available: <https://bit.ly/3RgHmEP>
- [55] A. Beltre, P. Saha, and M. Govindaraju, "KubeSphere: An approach to multi-tenant fair scheduling for kubernetes clusters," in *Proc. IEEE Cloud Summit*, Aug. 2019, pp. 14–20.
- [56] Z. Rejiba and J. Chamanara, "Custom scheduling in kubernetes: A survey on common problems and solution approaches," *ACM Comput. Surv.*, vol. 55, no. 7, pp. 1–37, Dec. 2022, doi: [10.1145/3544788](https://doi.org/10.1145/3544788).
- [57] J. Frtunikj, J. Fröhlich, T. Rohlfs, and A. Knoll, "Qualitative evaluation of fault hypotheses with non-intrusive fault injection," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW)*, Nov. 2015, pp. 160–167.
- [58] Y. Lee, J.-H. An, and Y. Kim, "Scheduler for distributed and collaborative container clusters based on multi-resource metric," in *Proc. Int. Conf. Res. Adapt. Convergent Syst. (RACS)*. New York, NY, USA: Association for Computing Machinery, 2020, pp. 279–281, doi: [10.1145/3400286.3418281](https://doi.org/10.1145/3400286.3418281).

- [59] S. V. Gogouitis, H. Mueller, S. Premnadh, A. Seitz, and B. Bruegge, "Seamless computing in industrial systems using container orchestration," *Future Gener. Comput. Syst.*, vol. 109, pp. 678–688, Aug. 2020.
- [60] J. Santos, C. Wang, T. Wauters, and F. D. Turck, "Diktyo: Network-aware scheduling in container-based clouds," *IEEE Trans. Netw. Service Manage.*, Apr. 2023.
- [61] D. Silva and R. C. Sofia, "A discussion on context-awareness to better support the IoT cloud/edge continuum," *IEEE Access*, vol. 8, pp. 193686–193694, 2020.
- [62] P. Chavhan, G. Chavhan, P. Shafi, and P. Mahalle, "Context awareness computing system—A survey," in *Proc. Int. Conf. Commun. Artif. Intell. (ICCAI)*. Singapore: Springer, 2020, pp. 423–429.
- [63] A. Atieh, P. Nanda, and M. Mohanty, "Context-aware fog computing implementation for Industrial Internet of Things," in *Proc. Int. Wireless Commun. Mobile Comput. (IWCMC)*, Jun. 2021, pp. 598–603.
- [64] Y. Wu, "Cloud-edge orchestration for the Internet of Things: Architecture and AI-powered data processing," *IEEE Internet Things J.*, vol. 8, no. 16, pp. 12792–12805, Aug. 2021.
- [65] E. Paraskevoulakou, J.-D. T. Tom-Ata, C. Symvoulidis, and D. Kyriazis, "Enhancing cloud-based application component placement with AI-driven operations," in *Proc. IEEE 14th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2024, pp. 687–694.
- [66] P. Gkonis, A. Giannopoulos P. Trakadas, X. Masip-Bruin, and F. D'Andria, "A survey on IoT-edge-cloud continuum systems: Status, challenges, use cases, and open issues," *Future Internet*, vol. 15, no. 12, 2023, Art. no. 383.
- [67] Y. Park, H. Yang, and Y. Kim, "Performance analysis of CNi (container networking interface) based container network," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2018, pp. 248–250.
- [68] D. Lachos, C. Rothenberg, Q. Xiang, Y. Yang, B. Ohlman, S. Randriamasy, and K. Gao, "Multi-domain information exposure using ALTO: The good, the bad and the solution," in *Proc. ACM Appl. Netw. Res. Workshop*, 2020, pp. 52–54.
- [69] L. F. Gonzalez, I. Vidal, F. Valera, and D. R. Lopez, "Link layer connectivity as a service for ad-hoc microservice platforms," *IEEE Netw.*, vol. 36, no. 1, pp. 10–17, Jan. 2022.
- [70] Z. Dehghani. *How to Move Beyond a Monolithic Data Lake to a Distributed Data Mesh*. Accessed: Nov. 5, 2023. [Online]. Available: <https://Martinfowler.com/articles/data-monolith-to-mesh.html>
- [71] V. Theodorou, I. Gerostathopoulos, I. Alshabani, A. Abelló, and D. Breitgand, "MEDAL: An AI-driven data fabric concept for elastic cloud-to-edge intelligence," in *Advanced Information Networking and Applications (Lecture Notes in Networks and Systems)*, vol. 227. Cham, Switzerland: Springer, 2021.
- [72] A. Ghiran and R. Buchmann, "The model-driven enterprise data fabric: A proposal based on conceptual modelling and knowledge graphs," in *Knowledge Science, Engineering and Management (Lecture Notes in Computer Science)*, vol. 11775. Cham, Switzerland: Springer, 2019.
- [73] R. Alimi, R. Penno, Y. Yang, S. Kiesel, S. Previdi, W. Roome, S. Shalunov, and R. Woundy, *Application-Layer Traffic Optimization (ALTO) Protocol*, document IETF RFC 9274, 2014.



RUTE C. SOFIA (Senior Member, IEEE) received the Ph.D. degree (summa cum laude), in 2004.

She is currently the Head of Industrial IoT with the Fortiss Research Institute and an Invited Associate Professor with the University Lusfona de Humanidades e Tecnologias and an Associate Researcher with ISTAR, Instituto Universitário de Lisboa. Her research background has been developed in industry (Grupo Forum, Lisbon; and Siemens AG, Nokia Networks, Munich) and academia (FCCN, Lisbon; INESC TEC, Porto; ULHT, Lisbon; and Universität der Bundeswehr, Munich). She was the Scientific Director of Portuguese Research Unit COPELABS (2013–2017) and the Co-Founder of COPELABS and Portuguese startup Senception Lda (2013–2019). Her current research interests include network architectures and protocols, the IoT, edge computing, edge AI, in-network computing, and 6G. She has over 80 peer-reviewed publications and nine patents in her areas of interest. She is the IEEE ComSoc WICE Deputy Liaison for Industry and an ACM Europe Councilor. She leads the 6G CONASENSE platform. She was the Co-Chair of the IEEE ComSoc N2Women Awards Board (2020–2021). She is also an Associate Editor for several venues, including IEEE Access and *IEEE Network*. She is the Coordinator of the Horizon Europe CODECO Project.



JOSH SALOMON received the degree in mathematics and in computer science from Hebrew University, Jerusalem. He is currently a Senior Principal Software Engineer with RedHat Israel. He has more than 30 years of experience in software development, more than 20 years in architecture positions, and seven years of experience as an Architect in the storage development and cloud industries. In CODECO, he leads the component ACM.



SIMONE FERLIN-REITER (Member, IEEE) received the Dipl.-Ing. degree in telecommunications from Friedrich-Alexander Erlangen-Nürnberg University, Germany, in 2010, and the Ph.D. degree in computer science from the University of Oslo, Norway, in 2017. She is currently a Senior Performance Engineer with Red Hat AB and a Senior Adjunct Lecturer with Karlstad University, Sweden. Her research interests include cellular networks and the internet,

computer networks, transport protocols, congestion control, network and system performance, security, and measurement. Her Ph.D. thesis focused on improving robustness in multipath transport for heterogeneous networks using MPTCP, resulting in several upstream contributions. She is actively involved in the technical committees of major conferences and journals in these areas.



LUIS GARCÉS-ERICE (Member, IEEE) received the M.Sc. degree in telecommunications engineering (EE+CS) from the Public University of Navarra, Spain, in 2001, and the Ph.D. degree from the TCOM University of Paris, France, in 2004. His Ph.D. thesis on peer-to-peer systems. He is currently a Senior Research Scientist with the Accelerated Discovery & AI Group, IBM Research, Zürich, Switzerland. He has published extensively in international conferences and journals

and received outstanding recognition for his work on products and projects for IBM Corporation. His research interests include distributed systems, network protocols, and middleware, with his current work focusing on large-scale computing systems for AI, such as data lakes and how to scale and govern them between the enterprise and the cloud. He is a Professional Member of the IEEE Computer Society and the ACM.



PETER URBANETZ is currently a member of the AI for Data Integration Team, IBM Research, Zürich, Switzerland. His recent projects have focused on data management in large organizations. The team created the Cognitive Enterprise Data Platform (CEDP) for IBM's Chief Data Office to provide a state-of-the-art platform for analyzing Hadoop-based data. From the CEDP experience, the team recognized the constraints associated with copying data to a common data

analytics platform, and therefore developed the idea of managing data based on decentralized data stores—the data remains with the organization that manages it—with centralized metadata to make it possible to simplify the job of managing and analyzing data while removing any practical scalability limits.



HARALD MUELLER received the Ph.D. degree in electrical engineering from the Technical University of Munich. He is currently a Principal Key Expert with Siemens Technology, Munich, Germany. He joined Siemens AG as a System Architect for VoIP products, in 1996. Since then, he has been working in various positions in several Siemens and Atos businesses, including communication networks, IT solutions and services, and smart grid. He joined Siemens Technology,

in 2014, where a major part of his work is research and technology transfer for seamless computing, which aims to enable the deployment and operation of industrial systems in the cloud-to-edge continuum.



RIZKALLAH TOUMA received the Ph.D. degree, in 2019. He is currently a Senior Researcher and the Technical Lead of the Distributed Data Architectures Research Line, i2CAT Foundation, Barcelona. He has more than eight years of hybrid experience in research organizations (Barcelona Supercomputing Centre, UPC) and the private sector (NTT Data). He was an Assistant Professor with UPC (2016–2021). He has had a leading technical role in different capacities in

multiple regional research and development and H2020/Horizon Europe projects (BigStorage, SoCaTel, WELCOME, BOOST 4.0, CODECO, and TEADAL). He has authored several scientific publications and acted as a reviewer in international conferences. His research interests include distributed data architectures, AI applications for distributed data management, data spaces, and semantic data management.



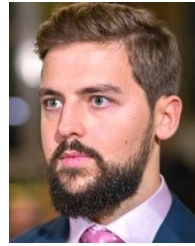
ALEJANDRO ESPINOSA received the Graduate degree in computer science from FIB UPC with a specialization in AI. He is currently a Computer Engineer specializing in AI research. His academic journey led him to explore the intricacies of AI, as shown in his bachelor's thesis with i2CAT, where he worked on the detection of data poisoning in AI models. Part of his thesis contributed to the emerging research area of applied AI in cybersecurity. He is also actively

contributing his skills to the Horizon Project CODECO, developing PDLC-DL, and is involved in ongoing research with i2CAT. Prior to his current roles, he gained practical experience as a Data Analyst with SDG. In this role, he played a crucial role in automating processes, including the development of a web crawler to retrieve information on individuals from the civil war.



LUIS M. CONTRERAS received the degree in telecommunications engineering from the Universidad Politécnica de Madrid, in 1997, the joint M.Sc. degree in telematics from the Universidad Carlos III de Madrid and the Universitat Politècnica de Catalunya, in 2010, and the Ph.D. degree in 2021. Since August 2011, he has been part of a Telefonía I+D/Telefónica Global CTO Unit, working on 5G, SDN, virtualization, transport networks and their interaction with cloud and distributed services,

and on interconnection topics. He is also a part-time Lecturer with the Universidad Carlos III de Madrid. Before the position with Telefonía, he was with Alcatel and Orange. He is actively involved in research and innovation activities, with more than 70 papers published in relevant journals, magazines, and conferences, being regular speaker at reputed academic and industrial events. He has participated on several projects funded by European Commission and the ESA. Currently, he is working on the EU H2020 projects 5GROWTH and 5G-EVE, the EU-TW 5G-DIVE, and the EU GNSS Agency project ROOT. On the standardization arena, he is an active Contributor to IETF (authoring five RFCs), ETSI, O-RAN, ITU-T, and ONF.



VASILEIOS THEODOROU received the Diploma (M.Sc. equivalent) degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 2010, the M.A. degree in information systems and technology from York University, Canada, in 2013, the Ph.D. degree from the Polytechnic University of Catalonia, Spain, in 2017, and the Dr.-Ing. Degree from the Technical University of Dresden, Germany, on the automation of user-centered design of data-

intensive processes. He is currently a Master Research Engineer with Intracom Telecom, Greece, working in the areas of multi-access edge computing (MEC), data-driven systems, and network function virtualization (NFV) for 5G/6G networks. He has been a Research Associate with the Database Technologies and Information Management Group, Polytechnic University of Catalonia (UPC), Spain, and a Research Assistant with the Adaptive Systems Research Laboratory, York University, Toronto, Canada. He has published in renowned international journals, conferences, and workshops; and has received numerous awards and grants for academic excellence.



NIKOS PSAROMANOLAKIS received the M.Eng. degree in electrical and computer engineering from the Technical University of Crete, in 2015, and the M.Sc. degree in electrical and computer engineering from the Technical University of Crete, in 2017, on wireless (5G) network systems. He is currently a Senior Research and Development Engineer with Intracom Telecom, involved in several projects in the area of cloud/edge computing, software defined

networking (SDN), and network function virtualization (NFV) for 5G/6G networks.



LEFTERIS MAMATRAS (Member, IEEE) is currently an Associate Professor with the Department of Applied Informatics, University of Macedonia, and a Scientific Associate with the ATHENA Research Center, Greece. In the past, he was a Researcher with University College London, U.K., the Democritus University of Thrace, Greece, and DoCoMo Eurolabs, Germany. He has also been involved in numerous international research projects, including CODECO (Horizon Europe),

NEPHELE (Horizon Europe), NECOS (H2020), FED4FIRE+ OC4 (H2020), WISHFUL OC2 (H2020), MONROE OC2 (H2020), and Extending Internet into Space (ESA). He has authored over 80 papers in international journals and conferences, focusing on areas such as software-defined networks, the Internet of Things, and 5G/6G networks.



VASSILIS TSAOUSSIDIS received the degree in applied mathematics and in computer science from Aristotle University and the Ph.D. degree in computer science from Humboldt University, Berlin. He is currently a Professor with the Democritus University of Thrace, Greece, and an Adjunct Professor with Athena RC. He has held faculty positions with the Department of Computer Science, SUNY Stony Brook, NY, USA, and the College of Computer Science, Northeastern

University, Boston, MA, USA. He has served as an Associate Editor for IEEE TRANSACTIONS ON MOBILE COMPUTING, *Computer Networks*, and *Ad Hoc Networks*. He was a member of the Board of the Hellenic Space Agency and a member of the Board of the Democritus University. He has coordinated several EU projects, such as FP-7 SPICE, FP-7 Space Data Routers, HORIZON UMOBILE, and NGI projects, such as a Trusted Content Networking and ESA Projects, such as Extending Internet into SPACE.



XIAOMING FU (Fellow, IEEE) received the Ph.D. degree from Tsinghua University, China, in 2000. He was a Research Associate with the Technical University of Berlin, until he joined as a Faculty Member of the University of Göttingen, in 2002, where he has been a Full Professor in computer science and the Head of the Computer Networks Group, since April 2007. He has held research positions with the Universities of Cambridge, Columbia, UCLA, Tsinghua, Nanjing, Fudan,

Uppsala, UPMC, and Sydney. His research interests include network architectures, protocols, and applications. He currently serves on the editorial boards for *IEEE Network* and *IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT* and is an Associate Editor of the *Journal of Social Computing*. He has served on conference committees for leading conferences, such as SIGCOMM, CoNEXT, MOBICOM, MOBIHOC, INFOCOM, ICNP, and ICDCS. He has been a PI in the EU FP6 projects ENABLE, VIDIOS, Daidalos-II, MING-T, H2020 COSAFE, HE CODECO, and COVER; and a Coordinator of the EU FP7 Projects MobileCloud, GreenICN and CleanSky, and H2020 ICN2020 Project.



TINGTING YUAN received the Ph.D. degree from Beijing University of Posts and Telecommunications (BUPT), China, in 2018. She held post-doctoral positions with INRIA, Sophia Antipolis, France, and the University of Göttingen, from 2018 to 2023. Since 2023, she has been a Junior Professor with the University of Göttingen.



ALBERTO DEL RIO received the degree in mobile and space communications engineering from the Universidad Carlos III de Madrid (UC3M) and the master's degree in signal processing and machine learning for big data from the Universidad Politécnica de Madrid (UPM), where he is currently pursuing the Ph.D. degree. He was with Deutsche Telekom, Berlin, on the specification of the 5G telecommunications system standard, Release 16. In particular, in the development of a system

framework concept focused on CLOUD services. He is also with the Group de Aplicación de Telecomunicaciones Visuales (GATV), UPM, in dedicated projects on 5G communication networks with the help of use-cases focused on AI.



DAVID JIMÉNEZ (Member, IEEE) received the Graduate (Hons.) and Doctor (cum laude) degrees in telecommunications engineering from the E. T. S. de Ingenieros de Telecomunicación, Universidad Politécnica de Madrid, in 2004 and 2012, respectively. He is currently a Lecturer in electrical engineering with the Department of Physical Electronics, Electrical Engineering and Applied Physics. Since 2004, he has been a member of the Grupo de Aplicación de Telecomunicaciones

Visuales (G@TV). His professional activity focuses on the field of smart energy grids (smart grids), the electric vehicle, 5G communications systems, and the application of IoT in the energy field. Lines of work that, he shares with audiovisual technologies, video processing and treatment, and video quality analysis and the average quality of the user experience. He is a member of the Parity Commission of the Chair of RTVE with UPM. He participates in various large-scale national and European projects on topics, such as the evolution of 5G and beyond networks (NEMO and CODECO), the development of energy communities in Europe (eNEURON), and the analysis of hybrid power plants comprising hydropower, Li-ion batteries, and supercapacitors (HybridHydro).



ANDRIES STAM received the Ph.D. degree in computer science from the University of Leiden, in 2009, on modeling the interaction in evolving distributed software systems. He is currently the CEO of Almende BV. For six years, he was a Consultant for various Dutch and international companies. He has been involved in more than 80 national and international research projects in various roles, ranging from principal investigation to coordination. He has published in the areas of

software engineering, coordination languages, and enterprise architecture. His research interests include model-driven software development, distributed systems, and coordination principles for evolving software systems.



EFTERPI PARASKEVOULAKOU received the bachelor's degree in economics from the University of Piraeus, in 2017, and the Diploma (master's) degree in big data and analytics from the Department of Digital Systems, University of Piraeus, in 2020, where she is currently pursuing the Ph.D. degree. She is also a Research Assistant with the University of Piraeus, where she is also a Dedicated Member of the Data & Cloud Laboratory Team, working under the esteemed guidance

of Dr. Dimosthenis Kyriazis. Her research interests include orchestration and optimal management of cloud-edge continuum applications and utilizing advanced technologies, such as artificial intelligence, machine learning, and data analytics.



PANAGIOTIS KARAMOLEGKOS received the bachelor's and master's degrees from the University of Piraeus. He is currently a Researcher and a Technical Programmer with the Department of Digital Systems, University of Piraeus. His main focus is on his studies, the research projects he participates in and the publication of significant results in international conferences and journals. In his spare time, he works as a Secondary School Teacher with Greece's Largest Tutoring Group (DIAKROTIMA). In terms of achievements, he was a member of the

Winning Team of the Second Coding Bootcamp of Athens University of Economics and Business (AUEB). He is one of the few Brown Belts in the Grepper community and has helped over 70 thousand developers find solutions to over 120 thousand problems. His research interests include IT education, the development of advanced information systems, and research on cloud environments.



VITOR VIEIRA received the M.Sc. degree in electrotechnical engineering from the University of Porto, Portugal, in 2004, and the Ph.D. degree in medical robotics from the University of Heidelberg, Germany, in 2010. He has background in image processing, technical, and software development in particular in user interfaces and robotics. After the Ph.D. degree, he accumulated almost six years of experience as the Software Development Manager of the Medical Device

Manufacturer Precisis AG, before co-founding Inova DE, in 2014, which later merged with INOVA + Portugal. In 2017, he won the award of entrepreneur of the year in "Services and Knowledge" category by the Fundation AEP and the high commissary for Portuguese immigrants ("Prmio Elevar o seu Negcio 4.0") and is also a mentor at different startup accelerator initiatives. More recently, he has been engaged in several communication and data management projects, including with some with AI focus.



JOSEP MARTRAT received the degree in telecommunication engineering from the Polytechnic University of Catalonia (UPC). He is currently the Market Manager of Telecom, Media and Edge Technology, Atos Research and Innovation. He works on the establishment of the innovation strategy for Atos' portfolio around telecoms. His research interests include study of cloud technologies, network virtualization (SDN/NFV), and new vertical services. He coordinated research projects, such as BonFIRE initiative about designing, building, and operating a multiCloud facility for experimentation with hybrid Clouds (public and private). He was an elected member of the Steering Board of 5G Industry Association (2013–2016). He has coordinated 5G SONATA/5GTANGO projects about agile network service development and deployment on software networks that released a NFV-based SDK, a validation and verification (V&V) NFV catalogue, and MANO service platform. He is currently coordinating Affordable5G initiative about creating an affordable solution targeting specific needs of private network roll-outs.



IGNACIO MARISCAL PRUSIEL received the degree in electrical engineering from the Universidad Politecnica de Burgos and the master's degree in computer science and automation from the Universidad Politecnica de Valencia. He has a background as a Software Developer. He is currently a Research and Development Engineer with Atos Research and Innovation, participating in projects focusing in the orchestration of the edge-cloud continuum.



DORINE MATZAKOU received the Ph.D. degree in nonlinear physics from Inria Sophia-Antipolis, France, in 2018. She was also a Postdoctoral Researcher with the University of Cote d'Azur, working on the mathematical modeling of complex systems. She is currently a Research and Development Project Manager of Netcompany-Intrasoft involved in several research projects.



JOHN SOLDATOS received the Ph.D. degree, in 2000. He is currently a Senior Innovation Delivery Consultant with Netcompany-Intrasoft (Luxembourg City/Luxembourg and Athens/Greece), a Honorary Research Fellow with the University of Glasgow, U.K., and a Scientific Advisor to high tech startups. He is also an Expert in Internet-of-Things (IoT) and artificial intelligence (AI), including AI/IoT applications in smart cities, industry (industry 4.0/5.0), and digital finance. He has played a leading role in the successful delivery of more than 70 (commercial-industrial, research, and business consulting) projects, for both private and public sector organizations. He is also a Co-Founder of the open-source platform OpenIoT. He has published more than 200 articles in international journals, books, and conference proceedings. He has recently co-edited and coauthored eight edited volumes (books) on artificial intelligence, the Internet of Things, and cybersecurity related themes. He is the author of the book *A 360-Degree View of IoT Technologies* (Artech House, December 2020).



DAVID REMON is currently the Research Project Manager of Eclipse Foundation. He joined Eclipse Foundation, in January 2023, bringing experience in industry and research projects in IoT, smart cities, or environment. He enjoys spending time in the mountains, traveling, and practicing any kind of sport.



MARCO JAHN is currently the Senior Research Project Manager of Eclipse Foundation. He is part of the Eclipse Foundation Research Team, where he helps to turn innovations into successful open-source projects.

...