EasyChair Preprint

№ 15116

Navigating the Dynamic Heterogeneous Computing Sphere: the Role of EdgeHarbor as a Multi-Edge Orchestrator

Francesco D'Andria, Alex Volkov and Josep Martrat

September 27, 2024

Navigating the Dynamic Heterogeneous Computing Sphere: The Role of EdgeHarbor as a Multi-Edge Orchestrator

Francesco D'Andria[1[0000-0002-8464-9450]], Alex Volkov[1[0009-0006-9925-698X]] and Josep Martrat[1[www.researchgate.net/profile/Josep-Martrat]]

[1] Edge R&D Team, BDS, Eviden (an Atos Business), Spain
{francesco.dandria, alex2.volkov, josep.martrat}@eviden.com

Abstract. The term "Dynamic Heterogeneous Computing Sphere" is employed to delineate a computing paradigm that is heterogeneous, volatile, and highly dynamic. This is the consequence of the incorporation of a substantial number of compute, storage, network, etc. resources, which are distributed across a spectrum that encompasses both infrastructure (physical and virtual) and data. These resources are distributed across a range of locations, from the extreme IoT to the edge (far and near) and to the cloud. To deploy and execute a set of innovative vertical applications that are heterogeneous, it is necessary to consider the deployment of a suitable computing paradigm. The envisaged system requires the capacity for both resources and services to be elastic, that is, capable of being continuously shaped and moulded to support the specific needs of highly demanding applications, both in terms of allocation and runtime. Moreover, software modules, which comprise vertical applications, should be partitioned intelligently into virtual elements (i.e., containers) to optimise their placement and subsequent execution. This can be achieved by considering aspects such as performance and eco-efficiency aspects. The necessity for ad hoc resource and service shaping is growing in accordance with the prevailing tendencies towards the softwareisation of systems management. This has been further driven by the concept of disaggregation and the advent of new ultra-real-time services with high performance demands (X-AR, holo, metaverse, etc.). This paper introduces the EdgeHarbor orchestrator service, an open-source, multi-edge management system for a dynamic heterogeneous computing sphere. The work has been supported by the European Union's Horizon research and innovation programme under grant agreement ICOS (www.icos-project.eu), grant number 101070177.

Keywords: Cloud-Edge-IoT continuum, Resource Management, Computing Sphere, Dynamic, Heterogeneous, Open-Source.

1 Introduction and Cloud-Edge-IoT context

The rise of multi-deallocated computing has transformed computing architectures, giving way to cloud-to-edge computing as a promising paradigm for distributed data processing. Edge computing offers numerous advantages, including reduced latency,

decreased network load, and enhanced security measures. However, managing a complex continuum computing system—where centralized cloud systems collaborate with highly distributed edge systems operated by multiple independent providers—necessitates a robust management strategy. This strategy must effectively address the unique characteristics and challenges posed by such a diverse computational landscape. A novel cognitive approach that leverages artificial intelligence (AI), data mining, pattern recognition, natural language processing, sentiment analysis, and context modeling can provide effective solutions to these challenges. This approach focuses on critical aspects such as energy-aware offloading, data sovereignty, and interoperability issues that arise from the varying underlying technologies and systems in use. Furthermore, CEI (Cloud-Edge-Internet) continuum systems are evolving into intricate ecosystems comprising diverse computing infrastructures and applications, collecting vast amounts of data from billions of IoT devices [1]. For instance, an astounding 850 Zettabytes of data were generated by IoT and mobile devices from 2016 to 2021, highlighting the scale and complexity of the data environment today [2].

Challenge 1: Develop innovative strategies that can adapt to the inherent characteristics of hyper-distributed computational scenarios. These scenarios can vary significantly based on factors such as the specific application domain, the providers involved, the services offered, and the unique data processing requirements. Addressing these variations is essential for ensuring efficient management and control of CEI continuum systems.

Challenge 2: The proposed CEI continuum management architecture must be capable of seamlessly and proactively managing heterogeneous physical and logical resources in a dynamic, semi-automated, adaptable, and cost-efficient (green) manner. This entails leveraging autonomous computing paradigms, such as swarm intelligence, supported by AI-assisted orchestration strategies and innovative programming models that enhance adaptability and resource allocation.

Challenge 3: To optimize resource utilization and ensure high Quality of Service (QoS) delivery while maintaining resilience, security, and privacy, an elastic load balancing mechanism for computing tasks is essential across the entire CEI continuum—from cloud to IoT. This will be achieved through the implementation of innovative orchestration strategies that adapt to varying workload demands and environmental conditions.

In response to these challenges, this paper introduces a proactive, highly automated, and adaptable CEI continuum management architecture. This architecture is designed to dynamically handle the data, services, and computational resources across the continuum, ensuring alignment with behavioral context requirements linked to users, infrastructure, data, and services. Moreover, it aims to capitalize on the substantial advantages that arise from effectively addressing the interoperability challenges among different underlying technologies, systems, and providers. Finally, recent developments in CEI continuum systems will be discussed, alongside potential limitations and future challenges that warrant further exploration [3].

2 Requirements Elicitation

The requirements elicitation presented in this section is the result of a comprehensive analysis and a set of activities involving the different actors within the value chain of the Cloud-Edge-IoT (CEI) continuum. This process was guided by the MoSCoW method [4], which prioritizes requirements based on whether they are Must-have, Should-have, Could-have, or Won't-have in the project's current scope. This approach helped streamline the scope of requirements elicitation. To formulate the functional requirements, we analyzed the problem from five key perspectives, which represent the core topics of the project:

1. Continuum Creation: This includes the requirements related to onboarding and setting up the CEI continuum ecosystem. It involves service and resource registration and discovery, as well as configuration tasks necessary to establish the ecosystem.
2. Continuum Management: These requirements address the governance and orchestration of resources within the CEI continuum ecosystem. This includes managing both the physical and virtual resources as well as overseeing the software services offered to end users.
3. Data Resiliency and Transformation: This focuses on the internal data management policies and mechanisms, including data access interfaces, caching policies, and optimizations for transforming data within the system to ensure efficiency and resiliency.
4. Smart Security and Trust: Security requirements are critical, covering aspects such as auditing system components, detecting and mitigating anomalies, and ensuring compliance. This includes identifying potential issues and implementing mitigation strategies.
5. Operability and Serviceability: Finally, this perspective focuses on the usability of the system for external users. It considers both graphical interfaces and command-line interfaces to ensure that the system is accessible and user-friendly.

A comprehensive list of functional and non-functional requirements for the CEI continuum can be found in Chapter 7 of the ICOS project deliverable D2.1, titled "ICOS Ecosystem: Technologies, Requirements, and State of the Art" [5]. In this paper, the CEI orchestration component implements only a subset of the requirements previously outlined. Specifically, it fully addresses Requirement 1 (Continuum Creation) and Requirement 2 (Continuum Management), and partially fulfills Requirement 5 (Operability and Serviceability). The remaining requirements, including Requirement 3 (Data Resiliency and Transformation) and Requirement 4 (Smart Security and Trust), are assumed to be covered by other functionalities within the broader system architecture, which have already been implemented and are not the focus of this paper.

3 Continuum Management Underlying Technologies

The proposed CEI architecture operates within a highly dynamic and heterogeneous environment, demanding a specialized software solution capable of managing the

deployment, execution, and orchestration of data, software services, and compute, storage, and network resources across the continuum. This dynamic system must account for the diverse and often incompatible components that comprise the continuum, such as differing operating system kernels, software dependencies, hardware drivers, and other low-level software layers. These discrepancies can introduce significant challenges, complicating the management of distributed resources, and hindering interoperability. To address these challenges, the CEI architecture incorporates a carefully curated set of open-source technologies, many of which are developed and maintained by the Cloud Native Computing Foundation (CNCF) community. The CNCF ecosystem offers a variety of tools that facilitate the compatibility and seamless orchestration of services across the Cloud-Edge continuum. These tools are essential not only for managing resources efficiently but also for ensuring that applications and services remain portable and adaptable across different cloud and edge environments, whether they are public, private, or hybrid.

The selected technologies enable seamless orchestration, critical for overcoming challenges related to resource management, orchestration, and networking in hyper-distributed systems. They provide a flexible foundation for managing diverse workloads and configurations, ultimately facilitating improved scalability, security, and performance across the Cloud-Edge continuum.

The following table (Table 1) summarizes the primary cloud and resource management technologies relevant to CEI continuum management. These technologies are critical in addressing the complexities and constraints of resource management, orchestration, and networking across distributed environments.

Table 1. Cloud and Resource management technologies.

Technology Name	Software Solution	License
Docker Engine: containerization technology built on top of Linux kernel containing other components such as chroot or namespaces	Docker Compose [8]	Apache License, V2.0
	Docker Swarm [9]	Apache License, V2.0
Cloud Based Operating Systems: designed for joint operation and deployment within cloud computing and virtualization environments, responsible for the management, operation, execution and all related processes of virtual machines, virtual servers, and virtual infrastructure, as well as the back-end hardware and software resources	OpenStack [10]	Apache License, V2.0
	AWS [11]	Proprietary License
Edge Containerization and Orchestration: provide a container orchestration to automatically provision, deploy, scale, and manage containerized applications without worrying about the underlying infrastructure. Developers can implement container orchestration anywhere containers are, allowing them to automate the life cycle management of containers.	Kubernetes (K8S) [12]	Apache License, V2.0
	Lightweight K8S: K3s [13]	Apache License, V2.0
	OKD [14]	Apache License, V2.0
	OpenShift [15]	Proprietary License

	Rancher [16]	Apache License, V2.0
Multi-Cluster Orchestration: orchestration and scheduling capabilities to place and manage workloads across multiple clusters.	Open Cluster Management [17]	Apache License, V2.0
	Rancher Fleet [18]	Apache License, V2.0
	KubeAdmiral [19]	Apache License, V2.0
	Liqo [20]	Apache License, V2.0
Far Edge Device Orchestration and Management: tools for data collection and management when data sources are not in the close proximity of data centers. Depending on the devices' capabilities, far edge processing can be supported, where in this case mobile nodes can collect and process data.	Genie [21]	Apache License, V2.0
	Nuvla [22]	Apache License, V2.0
	AWS IoT Greengrass [23]	Proprietary License
Cluster to Cluster networking and secure communications: direct networking connectivity between across computational resources either on-premises or in the cloud	Submariner [24]	Apache License, V2.0
	ClusterLink [25]	Apache License, V2.0

The technologies listed above are integral to the effective management of the CEI continuum, providing essential functionalities for resource deployment, service orchestration, and operational control across both cloud and edge environments. A key component in this architecture is the Edge-Harbor service orchestrator, a multi-edge orchestration solution responsible for managing the deployment, execution, and continuum oversight of data and software services. It is designed on top of those technologies, to meet specific behavioural and contextual requirements across the Cloud-Edge continuum. By leveraging these technologies, Edge-Harbor dynamically manages the Cloud-Edge continuum, ensuring that data and services are efficiently distributed and governed according to the system's context and operational needs.

This paper specifically focuses on the orchestration component within the CEI architecture, which fully addresses continuum creation (Requirement 1) and continuum management (Requirement 2), while partially fulfilling operability and serviceability (Requirement 5). Other aspects, such as data resiliency (Requirement 3) and smart security (Requirement 4), are assumed to be covered by additional functionalities that have already been implemented within the broader system ecosystem.

4 EdgeHarbor Architecture

The study identified a number of challenges associated with the creation, orchestration and underlying resources management of the CEI continuum and requirements analysis. This highlighted the necessity to establish two management layers:

- **Multi-Cluster Management layer**: responsible for the provision the dynamic heterogeneous computing sphere. This is achieved through the management of dissimilar and heterogeneous underlying clusters and nodes, as well as the physical or/and

virtual resources at their disposition. This includes the management of near and far edge devices.

- **Multi-Agent Management**: to develop a strategy that can be applied to different cluster management tools, regardless of their operational strategies, syntax, location, or provider.

The strategy for the optimal coordination of cluster managers is achieved through the use of an agent-based modelling approach. This approach introduces two software modules that implements the proposed solution, namely the Controller and the Agent.

4.1 Controller

Controller represents the top-tier entity that manages a set of heterogeneous agents. This entity enables the continuum creation from high-level perspective by abstracting the agent's underlying orchestration technologies, operating system and containerization technology. Proposed architecture reference is displayed Figure 1. EdgeHarbor reference architecture.. Aiming to address the mentioned challenges, fulfill the CEI continuum, and enable multi-agent and multi-cluster orchestration, controller entity provides the following sub-modules:

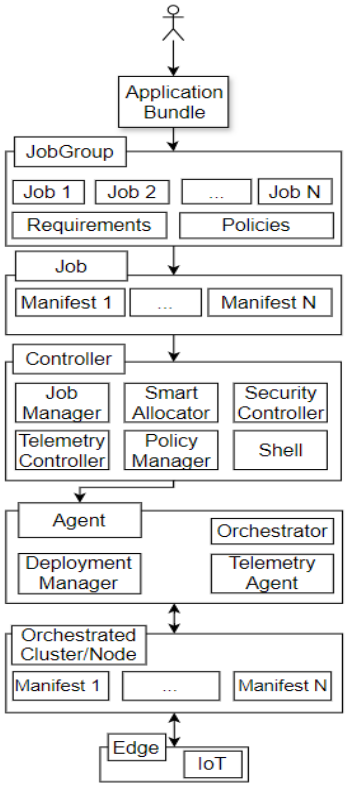

Figure 1. EdgeHarbor reference architecture.

- **Job Manager:** Stateful sub-module responsible for maintaining the state of the applications, related services and resources used by the latter. Operator submits the application in form of job group, which in fact is a set of jobs to be executed. These jobs are later-on handed over to a managed agent following the established deployment strategy without regards to underlying technology-specific requirements.
- **Job**: as mentioned above, represents a piece of service and/or deployment to be executed by managed agent, e.g., deploy an application component**.** Job is entirely agnostic as the agents are heterogeneous in their nature and shape However, mentioned actions can be described following the abstract development model, furthermore, being eligible for execution by any orchestration tool(agent). Development models are further defined in the following section 4.2. Additionally, the job keeps track of the state of the resources in use during the entire lifecycle of the latter. Finally, a job can be continuously reshaped or modified, thus, becoming "moldable" from the controller perspective.
- **Smart Allocator**: module responsible for selecting the appropriate agent where specific job must be executed. The selection consists of intelligent mapping between existing taxonomy of the resources and the application requirements. This sub-

module implements the Abstract Application Development Model, furthermore, can reshape the application for optimal allocation.

- **Security Controller**: Intelligent service responsible for smart security assurance within the piece of continuum that the controller manages. This sub-module guarantees the secure communication between controller and agent, as well as the internal communication among the different sub-modules.
- **Telemetry Module**: Responsible for providing intelligent observability features and continuously recollecting the existing taxonomy for supporting other components such as Policy Manager or Smart Allocator. This sub-module generates traces about the underlying resources and application metrics for further analysis.
- **Policy Manager**: Responsible for configuring and enforcing the service level agreement expressed in form of policies as part of the application description, periodically analyzing the taxonomy and the generated KPIs.

4.2 Agent

The Agent represents any kind of orchestration tool and the corresponding piece of continuum it manages (clusters, nodes and IoT) providing multi-cluster-management.

Furthermore, the agent is responsible for reaching the state of resources the controller demands for specific job. Therefore, agent provides elasticity to mentioned resources and devices as part of the management. The following sub-modules comprise the agent, to ensure the desired orchestration is possible, despite that an orchestrator can provide its own specific application development model and different management strategies. The agent provides the following sub-modules:

- **Deployment Manager:** The module responsible for handling the job transformation from abstract development model into specific application development model and pass it to the orchestrator for further execution, as shown in Figure 2. Development Model Abstraction Reference.. Consequently, this sub-module acts as the interface between the agent and the orchestrator in place. Each orchestrator must provide a piece of code in form of **driver** to be compliant with the continuum and the management strategy it offers.
- **Telemetry Agent:** Responsible for pushing the underlying taxonomy to the telemetry controller module. Also reports resources and devices usage and availability.

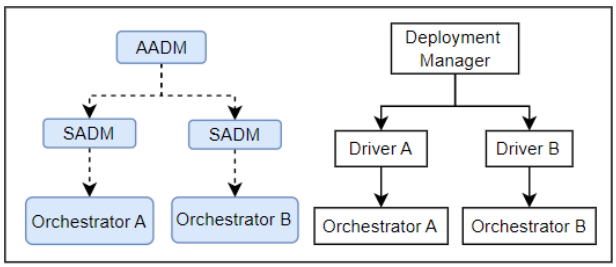

Figure 2. Development Model Abstraction Reference.

5 Conclusion and Future Work

This paper has introduced the EdgeHarbor reference architecture, designed to address the inherent challenges associated with managing the complex and evolving Cloud-Edge-IoT (CEI) continuum. Developed within the scope of the ICOS project, EdgeHarbor provides a comprehensive solution for orchestrating resources, data, network and services across distributed and heterogeneous environments. Through this work, we have laid the groundwork for a highly adaptive, scalable, and secure architecture that effectively bridges the gap between cloud and edge infrastructures.

The architecture's strength lies in its modularity and adaptability, allowing seamless integration of diverse clusters, nodes, and IoT devices through the combination of multi-cluster management and agent-based orchestration. Key modules such as ***the Job Manager, Smart Allocator, and Security Controller*** offer a flexible, intelligent framework for managing computational tasks and resource allocation in dynamic environments. Moreover, EdgeHarbor's ability to abstract underlying technologies enables it to function agnostically across different infrastructures, ensuring compatibility with a wide range of orchestration tools.

In terms of requirements fulfillment, EdgeHarbor fully addresses Continuum Creation and Continuum Management, providing a solid foundation for the deployment and management of services across the continuum. While other aspects such as Data Resiliency and Smart Security are partially covered, ongoing work in these areas will further enhance the architecture's robustness and reliability. However, challenges remain. As the CEI continuum continues to evolve, so too must EdgeHarbor. Future advancements will focus on optimizing performance, improving interoperability, and addressing the ever-growing demands of data sovereignty, security, and privacy.

Future Work

Several avenues of research and development are anticipated to further enhance EdgeHarbor's capabilities and address emerging challenges in edge computing:

1. Refinement and Optimization:

- The current architecture, while functional, will undergo further refinement to improve system performance and resource efficiency. This includes optimizing the Smart Allocator for more precise resource allocation, reducing latency across edge devices, and enhancing energy-aware strategies for offloading tasks to edge or cloud environments.
- Security mechanisms will be continuously improved, particularly in response to evolving threats within the distributed CEI continuum. This involves enhancing the Security Controller for better anomaly detection, automated threat mitigation, and compliance with privacy regulations such as GDPR.

2. Advanced Integration:

- As technological advancements such as 5G networks or edge AI continue to develop, we plan to integrate these emerging technologies into EdgeHarbor. For instance, 5G will provide ultra-low latency and enhanced bandwidth, which could significantly improve the orchestration and management of edge nodes, particularly in real-time applications.
- **Edge AI** will be a crucial component in the evolution of EdgeHarbor, enabling real-time, on-device decision-making that minimizes reliance on centralized cloud resources. This will be particularly beneficial for latency-sensitive applications such as autonomous vehicles, smart grids, and real-time video analytics.

3. Validation and Scalability Testing:

- As the architecture matures, rigorous testing and validation in real-world environments will be crucial to ensure scalability, fault tolerance, and resilience. The project will conduct extensive **scalability assessments**, where EdgeHarbor will be tested under varying conditions, from small-scale edge deployments to large-scale, multi-cluster environments.
- **Fault tolerance** will also be a focus, with comprehensive tests simulating node failures, network disruptions, and resource contention to ensure that the architecture remains resilient and maintains high availability.

4. Expanding Interoperability:

- Interoperability across heterogeneous edge and cloud platforms is a critical goal. Future work will involve improving **cross-domain orchestration** capabilities, enabling EdgeHarbor to seamlessly operate in highly diverse environments with varying infrastructures and service providers. This will also include advancing the **multi-agent framework**, ensuring compatibility with a wider range of orchestration tools and management strategies.
- As part of this effort, EdgeHarbor will continue to evolve its support for **multi-cloud and hybrid cloud deployments**, enabling organizations to manage resources efficiently across different cloud providers while maintaining a high degree of control over their data and services.

5. EdgeHarbor as a Service:

- A long-term vision for the project involves transforming EdgeHarbor into an **"EdgeHarbor-as-a-Service"** offering. This would allow organizations to adopt the architecture on-demand, leveraging its orchestration and management capabilities without the need for heavy investment in infrastructure. This service model would provide flexibility, scalability, and cost-efficiency, particularly for small to medium enterprises (SMEs) looking to integrate edge computing solutions into their operations.

In conclusion, the EdgeHarbor architecture represents a significant step forward in the orchestration and management of the Cloud-Edge-IoT continuum. By focusing on modularity, adaptability, and scalability, the architecture addresses the pressing needs of modern distributed computing environments. Looking ahead, we remain committed

to refining and expanding EdgeHarbor's capabilities, ensuring that it continues to meet the evolving demands of the edge computing landscape.

Acknowledgments. This work was supported by the European Union's HORIZON research and innovation programme under grant agreement ICOS (www.icos-project.eu), No. 101070177.

References

1. Gartner Research: A Guidance Framework for Architecting the Internet of Things Edge, Accessed 15 May 2024, <https://www.gartner.com/en/documents/3783144>
2. CISCO Research: The impact of connected devices, Accessed 15 May 2024, <https://www.cisco.com/c/en/us/solutions/service-provider/a-network-to-support-iot.html>
3. Jasenka Dizdarevic, Marc Michalke, Admela Jukan, Xavi Masip-Bruin, Francesco D'Andria, Engineering a functional IoT-edge-cloud continuum with open-source, CCGRID'2024
4. Agile Business Consortium, "Chapter 10 MoSCoW Prioritization," Jan 2014. [Online]. [Accessed 10 10 2021]. Available: https://www.agilebusiness.org/page/ProjectFramework_10_MoSCoWPrioritisation.
5. 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, [online] Available: https://www.mdpi.com/1999-5903/15/12/383.
6. ICOS deliverable D2.1 - ICOS ecosystem: Technologies, requirements, and state of the art, Accessed 15 May 2024, <https://www.icos-project.eu/deliverables>
7. Cloud Native Computing Foundation (CNCF) community, Accessed 15 May 2024, <https://community.cncf.io/>
8. Docker Compose, Accessed 15 May 2024, <https://docs.docker.com/compose/>
9. Docker Swarm, Accessed 15 May 2024, <https://docs.docker.com/engine/swarm/>
10. OpenStack, Accessed 15 May 2024, <https://www.openstack.org/>
11. AWS, Accessed 15 May 2024, <https://aws.amazon.com/>
12. Kubernetes, Accessed 15 May 2024, <https://github.com/kubernetes/kubernetes>
13. K3S, Accessed 15 May 2024, <https://github.com/k3s-io/k3s>
14. ORK, Accessed 15 May 2024, <https://github.com/okd-project>
15. OpenShift, Accessed 15 May 2024, <https://www.redhat.com/en/technologies/cloud-computing/openshift>
16. Rancher, Accessed 15 May 2024, <https://www.rancher.com/>
17. Open Cluster Management, Accessed 15 May 2024, < https://open-cluster-management.io/>
18. Rancher Feet, Accessed 15 May 2024, <https://fleet.rancher.io/>
19. Kube Admiral, Accessed 15 May 2024, <https://github.com/kubewharf/kubeadmiral>
20. Liqo, Accessed 15 May 2024, <https://liqo.io/>
21. Genie, Accessed 15 May 2024, <https://github.com/Netflix/genie>
22. Nuvla, Accessed 15 May 2024, <https://docs.nuvla.io/l>
23. AWS IoT Greengrass, Accessed 15 May 2024, <https://aws.amazon.com/es/greengrass/>
24. Submariner, Accessed 15 May 2024, < https://submariner.io/>
25. ClusterLink, Accessed 15 May 2024, < https://clusterlink.net/ >