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 and Immersive Media 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 used to describe a computing paradigm that is heterogeneous, volatile and highly dynamic. This results from the integration of a large set of resources, which are distributed across a continuum that includes both infrastructure (real and virtual) and data. These resources are distributed from the extreme IoT to the edge (far and near) and to the cloud. When the objective is to deploy and execute a set of innovative vertical applications that are heterogeneous in technology and in requirements. The envisioned scenario necessitates the capacity for both resources and services to be elastic, that is, to be capable of being continuously shaped and moulded to support the specific needs of those highly demanding applications, both in the allocation and runtime windows. Furthermore, software modules (those that compose vertical applications) should be intelligently partitioned into virtual elements (i.e., containers) to optimise their placement and consequent execution. This can be achieved by considering aspects such as performance and green aspects. The demand for ad hoc resource and service shaping is increasing in line with the current trends towards softwareisation of systems management. This has been further fuelled by the disaggregation concept and the development of new extremely demanding ultra-real-time services (X-AR, holo, metaverse, etc.). This paper introduces the architecture of the EdgeHarbor orchestrator as an open-source, multi-edge management system for a Dynamic Heterogeneous Computing Sphere. This work has been almost entirely 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 advent of multi-deallocated computing has led to a redefinition of computing architectures, with a cloud to edge computing emerging as a promising paradigm for distributed data processing. Edge computing offers a number of advantages, including reduced latency, limited network load, and enhanced security guarantees.

It is evident that such a complex continuum computing system, where centralized clouds systems should operate in conjunction with highly distributed edge systems, managed by multiple and usually completely independent providers, necessitates the implementation of a well-designed management strategy to accommodate the inherent characteristics of such a diverse computational scenario. Indeed, a novel cognitive approach that leverages artificial intelligence (AI), data mining, pattern recognition, natural language processing, sentiment analysis, and context modelling may provide a successful and accurate solution to address the challenges posed by the continuum. This solution addresses key aspects such as energy-aware offloading, data sovereignty and the interoperability challenges that arise when considering different underlying technologies, systems and providers. Likewise, Cloud-Edge-IoT (CEI) continuum systems are evolving into complex ecosystems of computing infrastructures and distributed computing applications, with data collected from multiple tens of billions of diverse IoT devices in today’s Internet [1]. In addition to this, multiple independent cloud providers, connected through heterogeneous networking technologies, and across distinct network zones and types, are only some of the additional characteristics of such systems. The data and AI-based systems embedded in the continuum not only add to the complexity of the system, but also to the scale of the data volumes. As of today, 850 Zettabytes of data were generated by IoT and mobile devices within the period 2016–2021 [2].

Challenge 1: develop innovative approaches that can accommodate the inherent characteristics of hyper-distributed computational scenarios, which may vary considerably depending on the application domain, the provider, the services and the data processing requirements. This is necessary in order to ensure an efficient management and control of Cloud-Edge-IoT (CEI) continuum systems.

Challenge 2: The proposed CEI continuum management architecture must be capable of seamlessly and pro-actively managing heterogeneous physical and logical resources in a dynamic, semi-automated, adaptable, cost-efficient (green) manner, while leveraging autonomous computing (i.e., swarm) through the support of AI-assisted orchestration strategies and innovative programming models.

Challenge 3: To optimise the utilisation of resources and the delivery of Quality of Service (QoS), while ensuring resilience, security and privacy, an elastic load balancing of computing tasks is required across the entire CEI continuum, from cloud to Internet of Things (IoT). This should be achieved through the implementation of innovative orchestration strategies.

In light of the aforementioned challenges, this paper presents a proactive, highly automated, and extremely adaptable CEI continuum management architecture [3]. This architecture is responsible for dynamically handling the data, the services, and the computational resources of the overall continuum in order to meet behavioural context requirements, including those specifically linked to users, infrastructure, data, and services. Furthermore, the architecture will benefit from the substantial advantages that arise from properly addressing the challenges associated with the interoperability of different underlying technologies, systems and providers. Finally, [5] presents all recent developments in the field of CEI continuum systems, with respect to the aforementioned challenges. In the same context, potential limitations and future challenges are highlighted as well.

2 Requirements Elicitation

The requirements elicitation is the result of analysis and a set of activities implemented within the different actors that constitute the value chain of the CEI continuum. This output is produced by following the MoSCoW [4] method to reduce the requirements elicitation extension. To produce the functional requirements, the problem is analysed from five different perspectives that are considered the main topics of the project. These are:

1. Continuum Creation: requirements related to the onboarding and setting up of the cloud-edge-iot continuum ecosystem, including service and resource discovery and configuration.
2. Continuum Management: requirements regarding the governance and orchestration of the resources that compose the proposed continuum ecosystem as well as the software services provided for the end users of this infrastructure.
3. Data Resiliency and Transformation: requirements related to the internal data management policies and mechanisms, including data access interfaces, caching policies, and data transformation optimisations.
4. Smart Security and Trust: requirements related to the security and audit when using the components, detection and mitigation of anomalies as well as detection of compliance issues and their mitigation.
5. Operability Serviceability: requirements related the system usability for external users in terms of graphical or command line interfaces.

Chapter 7 of the ICOS project deliverable D2.1 “ICOS ecosystem: Technologies, requirements and state of the art” [5] provides an exhaustive list of functional and non-functional requirements for the CEI continuum.

3 Continuum Management Underlying Technologies

The proposed CEI architecture is a dynamic system that necessitates the implementation of a highly specialised software solution to facilitate the deployment, execution, and continuum management of data, software services and computational resources. Currently, the continuum is constrained by incompatibilities between multiple components, including the operating system kernel, dependencies, drivers, and other low-level software layers.

This section provides a concise overview of the principal (most of them) open-source technologies that the proposed architecture has decided to be compatible with, with a particular focus on those developed by the Cloud Native Computing Foundation (CNCF) community.

In **Table 1.** Cloud and Resource management technologies. are presented the main Cloud and Resource management technologies. These technologies are deemed pertinent to the CEI continuum management.

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

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 Cloud-Edge-IoT continuum, and enable multi-agent and multi-cluster orchestration, controller entity provides the following sub-modules:

- **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 re-shaped 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.

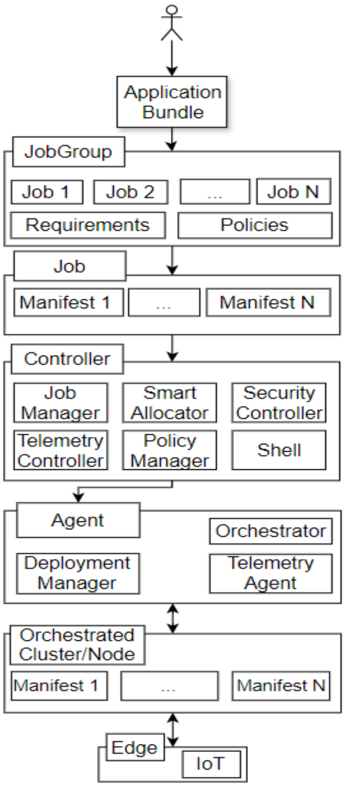

Figure 1. EdgeHarbor reference architecture.

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.

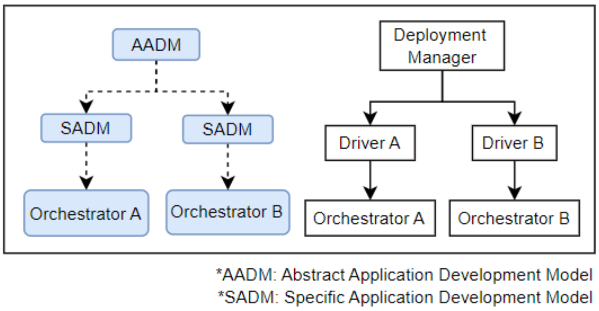

Figure 2. Development Model Abstraction Reference.

5 Conclusion and Future Work

The paper presents the EdgeHarbor reference architecture, which was developed in the context of the EU-funded ICOS project. It is based on user stories, requirements and specifications gathered from stakeholders, end-user feedback and industry best practices, standards and state of the art.

The work details a collaborative effort to define the architecture using principles such as functional suitability and privacy, in accordance with the ISO/IEC/IEEE 42010:2022 standard and Kruchten’s 4+1 Views framework. At this juncture, the ICOS consortium has implemented the initial draft version of its toolkits, which includes the EdgeHarbor orchestrator service. In the future, several avenues for further work will emerge. These include:

- Refinement and Optimization: As the architecture evolves, the project will continue to refine and optimize its components. This will involve addressing performance bottlenecks, enhancing security mechanisms, and streamlining resource allocation.
- Validation and Testing: Rigorous validation and testing are essential. We plan to conduct extensive evaluations, including scalability tests, fault tolerance assessments, and real-world deployment scenarios. We are investigating the possibility of integrating EdgeHarbor with emerging technologies such as 5G networks, quantum computing, and edge AI. These advancements will shape the architecture’s evolution.

In summary, the EdgeHarbor reference architecture represents a significant milestone in edge computing. By combining user-centric design, adherence to standards, and a comprehensive view, we lay the groundwork for efficient, secure, and scalable edge systems. As we move forward, we remain committed to advancing this field and contributing to the broader scientific community.

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/ >