EDGELESS Project: On the Road to Serverless Edge AI

Claudio Cicconetti
claudio.cicconetti@iit.cnr.it
National Research Council
Pisa, Italy

Emanuele Carlini
emanuele.carlini@isti.cnr.it
National Research Council
Pisa, Italy

Antonio Paradell
antonio.paradell@worldline.com
Worldline
Barcelona, Spain

ABSTRACT

The EDGELESS project is set to efficiently operate serverless computing in extremely diverse computing environments, from resource-constrained edge devices to highly-virtualized cloud platforms. Automatic deployment and reconfiguration will leverage AI/ML techniques, resulting in a flexible horizontally-scalable computation solution able to fully use heterogeneous edge resources while preserving vertical integration with the cloud and the benefits of serverless and its companion programming model, i.e., Function-as-a-Service (FaaS). The system under design will be environmentally sustainable, as it will dynamically concentrate resources physically (e.g., by temporarily switching off far-edge devices) or logically (e.g., by dispatching tasks towards a specific set of nodes) at the expense of performance-tolerant applications.

CCS CONCEPTS

• **Computing methodologies** → **Distributed algorithms**; *Distributed artificial intelligence*; • **Computer systems organization** → **Cloud computing**; • **Networks** → *Cloud computing*.

KEYWORDS

serverless computing, edge computing, Internet of Things, resource-constrained devices

ACM Reference Format:
Claudio Cicconetti, Emanuele Carlini, and Antonio Paradell. 2023. EDGELESS Project: On the Road to Serverless Edge AI. In *Proceedings of the 3rd Workshop on Flexible Resource and Application Management on the Edge (FRAME '23), June 20, 2023, Orlando, FL, USA*. ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3589010.3594890

1 INTRODUCTION

Cognitive edge-cloud with serverless computing (EDGELESS) is a collaborative project funded by the European Commission under the Horizon Europe program, which started on January 1st, 2023, with a target duration of 36 months. The project includes 12 partners from six European countries and is coordinated by Worldline (Spain).

EDGELESS aims to leverage the serverless concept [2] in all the layers in the edge-cloud continuum to fully benefit from diverse and decentralised computational resources available on demand close to where data are produced or consumed. In particular, we aim at realising an efficient and transparent *horizontal* pooling of the resources on edge nodes with constrained capabilities or specialised hardware, smoothly integrated with cloud resources, which is a giant leap forward compared to state-of-the-art *vertical* offloading solutions where the edge is a mere supplement of the cloud.

FRAME '23, June 20, 2023, Orlando, FL, USA

ACM ISBN 979-8-4007-0164-1/23/06…$15.00
https://doi.org/10.1145/3589010.3594890

2 OBJECTIVES

During the course of the project, the consortium plans to define an innovative approach to the execution of edge applications based on the dynamic orchestration of serverless functions running on heterogeneous edge devices, which will achieve the following objectives:

(1) Enabling efficient operation of data-intensive applications with a dynamic behaviour for the realisation of a cognitive framework spanning across the edge-cloud continuum, considering resource-constrained and heterogeneous edge computing resources under fast-changing conditions. Achieving this objective will require the implementation of algorithms for the efficient run-time composition of applications as a graph of lambda functions and ancillary services (e.g., persistence to implement stateful services). In particular, the choice, placement, and instantiation of these lambda functions and their interactions will be realised without relying on a centralised entity, but taking distributed, albeit coordinated, decisions under uncertainty, yet addressing global performance objectives under system stability. Such a composition will be the basis of a smooth interaction with the underlying orchestration system for finding the available executor instances (or creating new ones) and making them interoperate.
(2) Develop cognitive tools and techniques based on Machine Learning (ML) and Artificial Intelligence (AI) tools, for efficient use of resources in networks of constrained and specialised edge nodes. Such tools will consider computation needs and performance, always ensuring the most efficient implementation of function-oriented execution, according to a Function-as-a-Service (FaaS) paradigm [4].
(3) Enabling trusted access to lambda functions executed on edge nodes, including devices with limited computational capabilities, to attain a decentralised exchange of trusted data and computations, leveraging certified hardware security [3].
(4) Defining interfaces and models to deploy edge applications in a continuum multi-provider environment [5] according to specific functional and non-functional requirements while ensuring the highest level of Quality of Service (QoS).

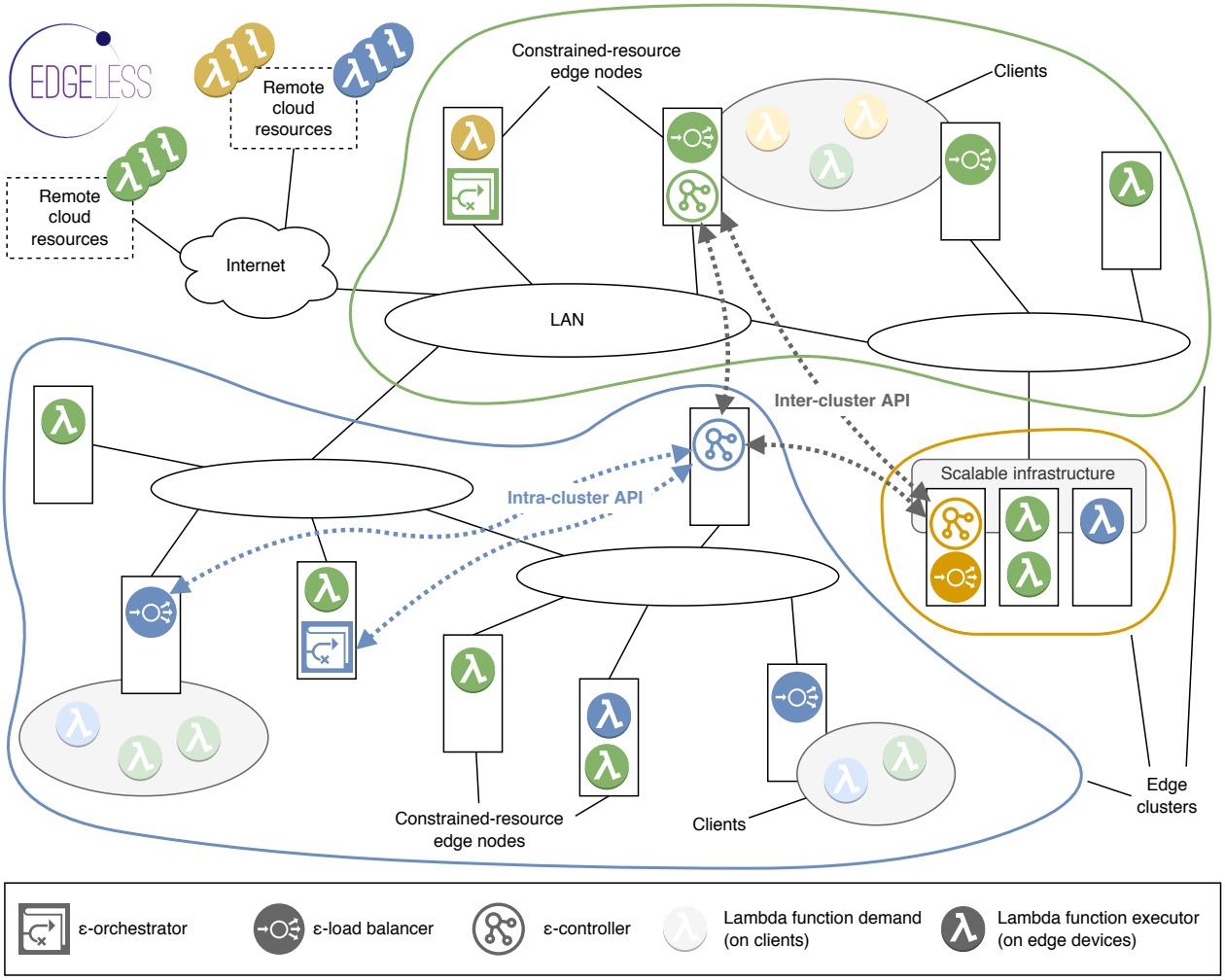

Figure 1: EDGELESS reference architecture.

(5) Evaluating the solution in three realistic use cases with heterogeneous requirements: i) Autonomous Smart City Surveillance, ii) Internet of Robotic Things, and iii) HealthCare Assistant.

3 INITIAL ARCHITECTURE

The reference architecture of EDGELESS is shown in Fig. 1. In the project, we define an edge domain as a set of clusters, each made of a collection of edge nodes that can be controlled as if it were a single serverless platform. For instance, we have three clusters in the figure, identified by different colours. Within each cluster, we have *cognitive and local* versions of the key components of any serverless platform [1], i.e., the orchestrator, the controller, and the load balancer, which we distinguish from their state-of-the-art equivalents by pre-pending the name with ε.

Due to the small scale of edge resources, and expected high load variability in some relevant scenarios, once a function is invoked by a user application, its dispatching is performed by a **ε-balancer**, which operates at the smallest possible time scale with the aim of maximising Quality of Experience (QoE) of the incoming lambda execution requests, also depending on the instantaneous location and context of the user invoking the function. Instead, the **ε-orchestrator** is responsible for managing the life-cycle of lambda function executors and ancillary services required, e.g., by loading/unloading the containers/unikernels and ensuring their connectivity, and exposing their entry points and capabilities. Finally, the **ε-controller** manages the lifecycle of functions (as opposed to the function executors), at an abstract level: it receives requests for the addition of new functions (either provided as part of the request or to be downloaded from a local/cloud repository), and it composes at complex run-time workflows, which require the invocation of multiple lambda functions. Furthermore, it mediates trust in both directions: on the one hand, it makes sure clients

are adequately authorised for the given operations they request to perform (e.g., lambda function invocation or creation); on the other hand, it enables clients to request the execution of certain lambda functions (as policy-based annotations) in a trusted environment, if required by the application. The ε-controller also interacts with the ε-orchestrator and ε-balancer in the same cluster with fine-grained details and monitoring information, as well as with peer ε-controllers in the other clusters of the edge domain (with coarse-grained detailed and aggregate monitoring information).

4 USE CASES

In the project, we focus on three use cases. The first use case is **Autonomous Smart City Surveillance**, which aims to increase citizens' safety by monitoring strategic geographical places through a city-wide distribution of CCTV cameras. Data processing at the edge of the network has several advantages, particularly low latency and backhaul traffic reduction (OPEX bandwidth reduction). However, deploying applications/services on low-powered and widely distributed devices, with rather limited computing capabilities, raises specific challenges which do not occur in a centralised cloud computing solution.

The second use case is **Internet of Robotic Things**, which originates from the increasing demand for complex distributed systems capable of handling large-scale factory automation. The movement to customer-driven production and personalization has generated a need for flexible manufacturing systems that can respond rapidly to production process variations. The outcomes of the project will allow the designing of a decentralised manufacturing system architecture that can support a distributed application logic, through self-contained and lightweight computation units across a serverless computing framework. Such FaaS-based approach will allow easy AI-supported reconfiguration of manufacturing systems, through cognitive services, and will enable a balanced computational load of intensive tasks across the EDGELESS framework.

Finally, the third use case is **HealthCare Assistant**: taking advantage of ML capabilities, it is focused on providing people with special needs living at home (seniors, pre/post-surgery or chronic disease patients) with a personalised assistant that will help them in their daily life. This includes monitoring their health status and activities and assisting them in difficult situations. For this purpose, the assistant will be capable of detecting by itself situations or moods that may require making decisions, e.g., making suggestions of activities, making a shopping list, contacting neighbours, friends or relatives, notifying caregivers, or even making emergency calls.

ACKNOWLEDGMENTS

This project has received funding from the HADEA program under Grant Agreement No 101092950 (EDGELESS project). This document reflects only the authors' view and the European Commission is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] Juan José López Escobar, Felipe Gil-Castiñeira, and Rebeca P. Díaz Redondo. 2023. Decentralized Serverless IoT Dataflow Architecture for the Cloud-to-Edge Continuum. *2023 26th Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN)* (3 2023), 42–49. https://doi.org/10.1109/ICIN56760.2023.10073502

[2] Yongkang Li, Yanying Lin, Yang Wang, Kejiang Ye, and Cheng-Zhong Xu. 2022. Serverless Computing: State-of-the-Art, Challenges and Opportunities. *IEEE Transactions on Services Computing* 1374 (2022), 1–1. Issue c. https://doi.org/10.1109/tsc.2022.3166553

[3] Panagiotis Papadopoulos, Giorgos Vasiliadis, Giorgos Christou, Evangelos Markatos, and Sotiris Ioannidis. 2017. No Sugar but All the Taste! Memory Encryption Without Architectural Support. In *Computer Security - ESORICS 2017 (Lecture Notes in Computer Science)*, Simon N. Foley, Dieter Gollmann, and Einar Snekkenes (Eds.). Springer International Publishing, 362–380.

[4] Ali Raza, Ibrahim Matta, Nabeel Akhtar, Vasiliki Kalavri, and Vatche Isahagian. 2021. SoK: Function-As-A-Service: From An Application Developer's Perspective. *Journal of Systems Research* 1 (9 2021), 1–20. Issue 1. https://doi.org/10.5070/SR31154815

[5] H. Zhao, Z. Benomar, T. Pfandzelter, and N. Georgantas. 2022. Supporting Multi-Cloud in Serverless Computing. In *2022 IEEE/ACM 15th International Conference on Utility and Cloud Computing (UCC)*. IEEE Computer Society, Los Alamitos, CA, USA, 285–290. https://doi.org/10.1109/UCC56403.2022.00051