The Journal of Systems and Software 210 (2024) 111942

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

New trends and ideas

Edge to cloud tools: A Multivocal Literature Review[☆]

Sergio Moreschini [a,*,1], Elham Younesian [a,b,1], David Hästbacka [a], Michele Albano [c], Jiří Hošek [b], Davide Taibi [a,d]

[a] Tampere University, Tampere, Finland
[b] Brno University of Technology, Brno, Czech Republic
[c] Aalborg University, Aalborg, Denmark
[d] University of Oulu, Oulu, Finland

ARTICLE INFO

Keywords:
Multivocal Literature Review
Systematic mapping studies
Edge
Cloud
Edge-to-cloud-tools

ABSTRACT

Edge-to-cloud computing is an emerging paradigm for distributing computational tasks between edge devices and cloud resources. Different approaches for orchestration, offloading, and many more purposes have been introduced in research. However, it is still not clear what has been implemented in the industry. This work aims to merge this gap by mapping the existing knowledge on edge-to-cloud tools by providing an overview of the current state of research in this area and identifying research gaps and challenges. For this purpose, we conducted a Multivocal Literature Review (MLR) by analyzing 40 tools from 1073 primary studies (220 PS from the white literature and 853 PS from the grey literature). We categorized the tools based on their characteristics and targeted environments. Overall, this systematic mapping study provides a comprehensive overview of edge-to-cloud tools and highlights several opportunities for researchers and practitioners for future research in this area.

Editor's note: Open Science material was validated by the Journal of Systems and Software Open Science Board.

1. Introduction

Edge computing is an emerging computing paradigm where data processing and storage are performed closer to the source rather than on centralized servers in the cloud. This is achieved by placing computing resources at the network edge, which enables the processing of data to be performed in real-time and with low latency.

In particular, the Edge-to-cloud Cloud Continuum enables the extension of the traditional Cloud towards multiple entities (e.g., Edge, Fog, IoT) providing analysis, processing, storage, and data generation capabilities (Moreschini et al., 2022).

In order to support the computation in the Edge-to-cloud continuum, different tools have emerged. With edge-to-cloud tools we refer to the combination of software, and services used to collect, process, and analyze data from edge devices (sensors, machines, etc.) to the cloud.

The characteristics of edge-to-cloud tools vary depending on the specific use case and organization's needs. These characteristics can include processing and analyzing data at the edge of the network, reducing latency, optimizing data transmission, scalability, and flexibility. Additionally, edge-to-cloud tools can provide access to advanced analytics and machine learning algorithms, enabling organizations to gain insights and make data-driven decisions. In this context, understanding the different characteristics of edge-to-cloud tools such as offloading, orchestration, workflow management and other computational tasks in the cloud continuum is crucial to harness their full potential for various applications.

In order to solve the aforementioned issues, we performed a systematic mapping study from 1073 primary studies in academic and industrial (i.e., grey literature) sources to classify the edge-to-cloud tools in the cognitive cloud continuum. Therefore, the goal of this work is to contribute to the state-of-the-art by providing a comparison of edge-to-cloud tools for both researchers and practitioners. First of all, such a comparison provides the reader with a list of all the available tools. Following this, by performing a comparison, the reader can understand the main characteristics of each tool, including its license. Another fundamental aspect that we aim to tackle is the target environment of each tool so that users can understand which architecture a tool is compatible with.

The rest of this paper is structured as follows. Section 2 introduces the background with a particular focus on concepts such as Cloud

☆ Editor: Uwe Zdun.

* Corresponding author.
E-mail addresses: sergio.moreschini@tuni.fi (S. Moreschini), elham.younesian@vut.cz (E. Younesian), david.hastbacka@tuni.fi (D. Hästbacka), mialb@cs.aau.dk (M. Albano), hosek@feec.vutbr.cz (J. Hošek), davide.taibi@oulu.fi (D. Taibi).

[1] The two authors equally contributed to the paper.

https://doi.org/10.1016/j.jss.2023.111942
Received 17 April 2023; Received in revised form 4 October 2023; Accepted 20 December 2023
Available online 27 December 2023

Continuum and Edge-to-Cloud Offloading. Section 3 presents the main goal of the work and the related research questions. Section 4 reports the method, and the multiple steps performed to follow it, used in this paper to answer the 3 research questions proposed. Section 5, illustrates the results obtained by answering the different research questions. The main discussion points, future challenges and possible threats to validity. The work ends with the conclusions presented in Section 7.

2. Background

In this section, we introduce the background of this work in cloud, edge, cloud continuum, and edge-to-cloud technologies.

Cloud continuum is the seamless integration of different cloud services and resources, such as IoT devices, fog, and edge nodes. Edge and cloud technologies have grown significantly during these years. Therefore, investigating offloading between these environments has become important for practitioners and academics. In this Section, we provide an overview of the cloud continuum and edge-to-cloud offloading.

2.1. Cloud continuum

Cloud Computing was defined officially in 2011 by the National Institute of Standard and Technologies (NIST) as *"a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"* (Mell et al., 2011). Among its main characteristics, we have scalability and reliability; the first is granted by the possibility of creating multiple instances which can be easily distributed. Recently the cloud has been used as a platform abstracting underlying infrastructure resources, particularly for Serverless and Functions as a Service (Nupponen and Taibi, 2020; Aslanpour et al., 2021).

Fog computing is the computing layer between the cloud and the edge. The main goal of fog nodes is to minimize the load on the cloud by performing some services closer to the edge, providing a reduction both in streaming loads and response time.

In Edge computing, the computation takes place at the edge of the network where the data is usually generated. Use the Edge is an effective solution whenever there are network problems and very strict response time. However, it is essential to highlight that edge computing does not have the computing and storing capabilities of the cloud or the fog. To solve the problems related to restrictions in timing, storage, or computational power the concepts of Cloud Continuum and Cognitive Cloud have been proposed.

Cloud Continuum has been defined as "*an extension of the traditional Cloud towards multiple entities (e.g., Edge, Fog, IoT) that provide analysis, processing, storage, and data generation capabilities*" (Moreschini et al., 2022).

Cognitive Cloud, instead, is defined as "*a Cloud-based system that is capable of sensing its environment, learning from it, and opportunistically and dynamically adapt its computational load as well as its outcome*" (Moreschini et al., 2023).

The main difference between these two definitions is that the Cloud Continuum is defined as the medium used to perform the computation while the Cognitive Cloud towards the capability of adapting the computational needs such as the load or the outcome.

Cloud computing technology reshaped traditional infrastructure, platform, and software resources into flexible and available virtual components. The effective orchestration of heterogeneous and multi-layer resources is essential to ensure that end-users receive satisfactory quality levels. Commercial cloud providers now offer proprietary cloud orchestration platforms to users. These solutions, while powerful, lack portability due to business considerations. Additionally, even with modern configuration management tools often need to grapple with low-level cloud service APIs and procedural programming constructs to manage intricate resource configurations. Multi-cloud computing compounds orchestration challenges. It is a recent trend in cloud computing, involving the use of services and resources from multiple cloud providers without a prior agreement, often managed by a third party (Tomarchio et al., 2020).

The practical realization of the Computing Continuum vision faces significant challenges due to the intricate nature of deploying applications across highly dispersed and diverse Edge-to-Cloud infrastructures. Achieving this vision involves intricate tasks such as configuring numerous system-specific parameters, as well as harmonizing various demands and limitations concerning interoperability, mobility, communication latency, network efficiency, data privacy, and hardware resource utilization (e.g., GPU memory, CPU power, storage capacity, among others) (Rosendo et al., 2022).

2.2. Edge-to-cloud offloading

As mentioned in the previous section, Edge Computing extends computation facilities towards the edge of a network. Therefore, computation is performed near the end user, resulting in ultra-low latency and high bandwidth. Offloading algorithms allow end devices, edge nodes, and the cloud to work together. Generally, task offloading can be defined as the transfer of resource-intensive computational tasks to an external, resource-rich platform such as the ones used in Cloud, Edge, or Fog Computing.

There are different types of task offloading based on where the tasks are split and where is performed the computation; these are:

- Partial offloading at the edge: In this type, part of the computation is executed locally at the end device, and remained will be offloaded at the edge.
- Full offloading at the edge: In this case, all computation tasks will be offloaded and executed at the edge.
- Collaborative offloading at the edge and at the cloud: Such offloading is for situations where the edge resources cannot execute all the tasks offloaded from the end device. Therefore, edge and cloud collaborate to process all the computational tasks (Saeik et al., 2021).

2.2.1. Targets of task offloading

There are different objectives for task offloading in the cloud continuum based on the different stakeholders, which can be categorized as follows (Saeik et al., 2021):

- Delay: Minimizing the task execution delay is one of the main goals of task offloading. This delay can be split into different parts. It could be the delay related to the task execution at the device, the edge, or the cloud, delay related to the transmission at the various layers of the infrastructure, queuing delay, or task partitioning delay. The goal of reducing the delay by task offloading can be minimizing each of the mentioned delays or the average one (Yang et al., 2013).
- Energy consumption: How to minimize the energy consumption by using task offloading is another meaningful objective of task offloading that typically refers to the end devices (Sardellitti et al., 2014). However, minimizing the energy consumption has to be followed by all the layers of this communication model because this problem is pushed to the edge and/or cloud infrastructure at the full offloading model (Mao et al., 2016; Singh and Awasthi, 2013).
- Bandwidth/spectrum: Allocation of spectrum in IoT and cellular networks plays an important role because of the limited bandwidth availability. Evaluating the spectrum utilization based on the number of offloaded tasks, power transmission, and bandwidth consumption is an efficient metric to deploy the available spectrum optimally (Sahni et al., 2017; Zhao et al., 2017; Mao et al., 2017; Zhang et al., 2017).

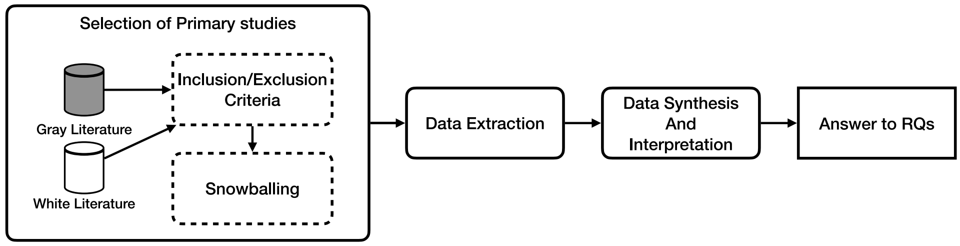

Fig. 1. Overview of the followed MLR process.

3. Research questions

Our goal is to identify edge-to-cloud tools available on the market and classify their characteristics.

To achieve the aforementioned goal, we defined 3 main research questions (RQs).

RQ1. Which edge-to-cloud tools are available on the market?
In this RQ, we aim at finding tools capable of performing offloading, either vertical (edge-to-cloud) or horizontal (edge-only). The available tools are researched in the grey and peer-reviewed literature.

RQ2. What are the characteristics of edge-to-cloud tools?
In this RQ we aim at finding the characteristics of the tools which have been found from different sources. Whether they are just for offloading, capable of orchestrating, or if they are simulators.

RQ3. What are the target environments of edge-to-cloud tools?
In this RQ we aim at finding out the environment where the discovered tools can work. In this context, the target environment classifies the tools based on the specific environments they are tailored to address.

4. Study design

This section outlines the process used for this work, which involved a Multivocal Literature Review (MLR) following the guidelines proposed by Garousi et al. (2019), due to the topic's novelty. The following subsections provide a summary of the MLR process, including the selection of primary studies, quality assessment of the grey literature, data extraction, tool selection, conducting the review, and verifiability and replicability.

4.1. MLR process overview

The MLR process encompasses a variety of sources, including peer-reviewed and grey literature, and acknowledges the diverse perspectives of both practitioners and academic researchers. To categorize contributions, MLR distinguishes between academic literature (peer-reviewed papers) and grey literature (other forms of content such as blog posts, white papers, podcasts, etc.) An overview of the whole process is summarised in Fig. 1.

4.2. MLR motivation

When it comes to exploring a complex and rapidly evolving field like Edge to Cloud Tools, it is crucial to approach the literature with a multivocal perspective. Multivocal literature reviews acknowledge diverse perspectives, opinions, and experiences of researchers and practitioners. By incorporating multiple voices, this approach can help to surface hidden assumptions, biases, and blind spots in the existing literature, as well as open up new avenues for inquiry and innovation.

In the context of edge-to-cloud tools, a multivocal literature review could of particular value. From the perspectives of software developers, hardware manufacturers, cloud providers, policymakers, end-users, and other stakeholders, the benefits, risks, and trade-offs of Edge to Cloud Tools can look very different. By engaging with multiple voices, this multivocal literature review can help to paint a more nuanced and comprehensive picture of Edge to Cloud Tools. It can reveal the divergent interests, values, and priorities that underpin different perspectives. This approach can also help to highlight gaps and contradictions in the existing literature and identify opportunities for further research and collaboration.

Ultimately, this work can help to enrich the understanding of Edge to Cloud Tools and inform more effective and inclusive approaches to their development and deployment. It can help us to embrace the complexity and diversity of this dynamic field.

4.3. Selection of primary studies

The first step for selecting the Primary Studies (PS) is the search string identification that will be adopted in the academic bibliographic sources and in the grey literature source engines. We define the search string as follows:

("edge cloud" OR "edge-to-cloud")
AND
(offloading OR cognitive OR orchestration)
AND
tool

To maximize the number of retrieved works, the search terms were used across all fields (i.e. title, abstract, and keywords). The same search terms were utilized for both grey literature from online sources and white literature from academic bibliographic sources, with both searches being performed in November 2022.

Peer-reviewed literature search. We considered the papers indexed by five bibliographic sources:

- IEEEXplore digital library (Anon, 2023b)
- Scopus (Anon, 2023f)
- ACM digital library (Anon, 2023a)
- Science Direct (Anon, 2023e)
- ISI Web of science (Anon, 2023g)

Grey literature search. We performed the search using two search engines:

- Google Search[2]
- Medium[3]

The search results consisted of books, blog posts, forums, websites, videos, white-paper, frameworks, and podcasts. Specifically, for Google Search we limited our search by forcing all the 6 different combinations of the search string.

Application of inclusion and exclusion criteria. Based on guidelines for Systematic Literature Reviews (Keele et al., 2007), we defined *inclusion and exclusion* criteria (Table 1). We considered less restrictive

[2] https://www.google.com/
[3] https://medium.com

Table 1
Inclusion/Exclusion criteria for primary studies selection.

Primary Study (PS)	Criteria
Inclusion	Research papers or search results that are commercial and open source tools performing computational tasks along the Cloud Continuum (Edge-to-Cloud)
Exclusion	Not in English Duplicated (post summarizing other websites) Out of topic (using the terms for other purposes) Non peer-reviewed papers Research Plans, roadmaps, vision papers

Table 2
Data extraction.

RQs	Info	Description	Step
RQ_1 RQ_2	Tool Name Tool Url	Name of the tool	PE
RQ_1	*Where* to move	Categorizing the tools based on where they can move the computational tasks	TE
RQ_2	Characteristics	Identify the main characteristics of each tools	
RQ_3	Environment	Main environment for the tool	

Table 3
Initial search result from bibliographic sources.

Bibliographic source	#non-duplicated papers
IEEEXplore	5
Scopus	144
ACM Digital Library	69
Science Direct	1
ISI Web of Science	1
Total	**220**

Table 4
Search results from grey literature search engines.

Search engines	#non-duplicated search result
Google search	495
Medium search	358
Total	**853**

Table 5
Inclusion/Exclusion criteria for tool selection.

Tools Selection	Criteria
Inclusion	Commercial and Open Source Tools
Exclusion	Not downloadable
Exclusion	Open Source Tools with less than 100 stars in GitHub

inclusion criteria to enable the inclusion of a more comprehensive set of tools.

To ensure the effectiveness of the inclusion and exclusion criteria, a subset of 10 randomly selected primary studies (PSs) from those retrieved were tested before their application. The resulting inclusion and exclusion criteria are outlined in Table 1.

To screen each entry, two researchers were tasked with independently fairly reviewing them. The entry assignments were mixed up, and each researcher was given a similar number of entries to review, along with other team members. Cohen's kappa coefficient was calculated to assess inter-rater agreement following the guidelines provided in Emam (1999). Cohen's coefficient kappa (k) is therefore calculated as:

$$k = \frac{P_0 - P_e}{1 - P_e} \quad (1)$$

where P_0 is the sum of the proportion of the agreements (i.e. sum of the proportion of included by both and excluded by both) and P_e is the chance agreement (i.e. the observed data is used to calculate the probabilities of each observer randomly seeing each category).

4.4. Data extraction

As our goal is to characterize information from edge-to-cloud offloading tools, we need to get the information directly from the tools' websites. Therefore, the data extraction process is composed of two steps:

(PE) *Extraction of the list of tools from the primary studies* (PSs) that satisfied the quality assessment criteria.
(TE) *Extraction of the information from the tools list*. In this case, we extracted the information directly from the official website portals.

We utilized a review spreadsheet to manually extract information following our research questions (RQs). A summary of the data extraction form can be found in Table 2, as well as the mapping of the necessary information for addressing each RQ.

The data extraction process adhered to the qualitative analysis guidelines proposed by Wohlin et al. (2012) as the extraction form is separated into two parts for quality data and for the study of the data. All information was extracted by two researchers. If there was disagreement, a third author was consulted, and a discussion was held until the disagreement was resolved.

4.5. Tool selection

To identify the final set of tools required to answer our RQs, a similar process to the one employed in the paper selection phase (Section 4.3) was applied, which involved filtering the tools based on a set of inclusion and exclusion criteria.

As with the PS selection process, we tested the applicability of the inclusion and exclusion criteria on a randomly selected subset of 10% of the retrieved tools before applying them. The final set of inclusion and exclusion criteria is presented in Table 1.

The tool selection was carried out by two researchers who independently reviewed them, as was the case with the PS selection. Any discrepancies, a third author was brought in to reach a consensus. The inter-rater agreement was also assessed in this case by calculating Cohen's kappa coefficient Emam (1999). These results are documented in the replication package.[4]

4.6. Conducting the review

From the Search process, conducted in November 2022, we retrieved a total of 1073 unique PS (after the exclusion of 137 duplicated): 220 PS from the white literature (Table 3) and 853 PS from the grey literature (Table 4).

Out of the 1073 works we retrieved and after applying inclusion and exclusion criteria, with an almost perfect agreement (Cohen's kappa = 0.765), the two authors agreed on excluding 771 works (157 from white literature and 614 from grey literature), resulting in **302** PS (63 from white literature, 239 from grey literature).

From the data extraction process, 338 tools were obtained.

The application of inclusion and exclusion criteria for tools resulted in a fair agreement (Cohen's kappa = 0.523) and a final set of 68 tools, as reported in Table 7.

After applying the inclusion and exclusion criteria, we focused on one of the characteristics extracted during the previous stage: *Tool Type*. In the previous stage, we had 3 possible outcomes for the Tool Type characteristic, namely Commercial, Open Source, and Research Prototype. As a quality check, we included two more exclusion criteria as reported in Table 5. The reason behind this choice is to favor only projects that can be categorized as popular (Han et al., 2019; Du et al., 2020; Xiao et al., 2022). This would result in checking every single tool marked as a Research Prototype and categorizing it as Commercial or Open Source. As a result, we included 40 tools.

Table 6
Edge to cloud tools and their characteristics.

RQ1				RQ2										RQ3				RQ2
Tool code	Tool name	edge-to-cloud	edge-only	Offloading	Orchestration	Workflow Management	Proprietary Container	Deployment	Kubernetes Distribution	Kubernetes Extension	Simulator	E2E Service	Platform	Agnostic	Cloud	Edge Node	Far Edge	License
T1	Amazon EKS		x							x				x				Com
T2	Ambassador Edge Stack		x							x				x				Com
T3	Apache Airflow	x				x								x				Com
T4	APEX	x										x		x				Com
T5	Aruba ESP	x										x			x			Com
T6	Avassa		x					x						x				Com
T7	AWS IoT GreenGrass	x										x				x		Com
T8	AWS Wavelength	x										x				x		Com
T9	Azure stack Edge	x										x			x			Com
T10	Baetyl		x							x				x				OSS
T11	Cloudify		x					x						x				Com
T12	Docker Swarm	x			x									x				Com
T13	Eclipse ioFog 2.0	x			x			x						x				Com
T14	EdgeCloudSim	x									x				x			OSS
T15	eKuiper	x				x											x	OSS
T16	FogFlow	x			x									x				OSS
T17	Home edge orchesterator	x											x	x				OSS
T18	Intel Smart Edge	x											x		x			OSS
T19	k0s	x			x				x					x				OSS
T20	K3s	x			x				x					x				OSS
T21	KubeEdge	x			x				x					x				OSS
T22	KubeFed	x			x				x					x				OSS
T23	Kubernetes	x			x				x					x				OSS
T24	MicroK8S	x		x										x				OSS
T25	Microsoft's Azure IoT Edge	x										x		x				Com
T26	Nearby One	x			x									x				Com
T27	Nomad	x			x		x						x	x				OSS
T28	NEBULA	x			x		x						x	x				OSS
T29	Nuvlabox (NuvlaEdge)	x											x	x				Com
T30	ONAP	x			x										x			OSS
T31	Open horizon	x											x			x		Com
T32	Open Stack Starlingx	x						x							x			OSS
T33	OpenNebula	x											x	x				Com
T34	OpenShift	x											x	x				Com
T35	Openstack	x											x	x				Com
T36	OpenYurt	x											x	x				OSS
T37	Ormuco		x									x	x	x				Com
T38	Saguna	x											x			x		Com
T39	Windriver Studio	x											x	x				Com
T40	Zededa	x											x	x				Com

4.7. Verifiability and replicability

To allow our study to be replicated, we have published the complete raw data in the replication package.[4]

5. Study results

In this section, we present the results of our MLR study, guided by the research questions stated in Section 3. Table 6 summarize the results achieved among the different RQs. Such a Table can be subdivided into 4 main areas: the first one includes the information related to RQ1, the second is related to RQ2, the third to RQ3, and the last area is specifically reserved for the license.

5.1. Edge-to-cloud tools (RQ1)

Description. To answer RQ1, we identified the tools that can perform offload, orchestration, or other computational tasks in the cloud continuum.

Results. The main objective of performing task offloading, orchestration, or workflow management is to optimize the computational tasks from an end-user device to a remote site under specific constraints. This process consists of three main parts that are (i) various hardware components, such as end-user devices and Edge/Cloud devices, (ii) multiple computing processes, including task splitting and computational processing either locally or remotely and (iii) networking components for transferring data between the hardware components involved (Saeik et al., 2021).

As discussed in Section 2.2, the different types of computational tasks are categorized based on where they are executed. According to the final list of selected tools, we identify two types; vertical and horizontal, i.e. *edge-to-cloud* and *edge-only*. As illustrated in Table 6, from the 40 tools, 6 of them are capable of moving the tasks only among edge devices, while 34 can move to the cloud.

5.2. Tools characteristics (RQ2)

Description. To answer RQ2, we extracted the characteristics of the tools.

Results. We extracted characteristics of the identified tools and their alternatives, and grouped them into 11 categories:

- **Offloading:** tools used to perform different kinds of offloading as introduced in Section 2.2.
- **Orchestration:** tools performing Orchestration with a focus on the edge, i.e., tools that are capable of "managing, automating and coordinating the flow of resources between multiple types of devices, infrastructure, and network domains at the edge of a network" (Stackpath, 2023).
- **Workflow Management:** tools used to manage the workflow and tasks assignable. In contrast with orchestration, workflows are usually not scalable and only allow for the creation of task queues that can be executed sequentially by a limited number of resources. In many cases, high-level processes are connected through monolithic applications in a workflow. However, orchestration goes beyond this by managing not only high-level processes but also low-level services and virtual infrastructure. It adapts to changing demands and scales accordingly.

[4] https://figshare.com/articles/dataset/Replication_package/22567708

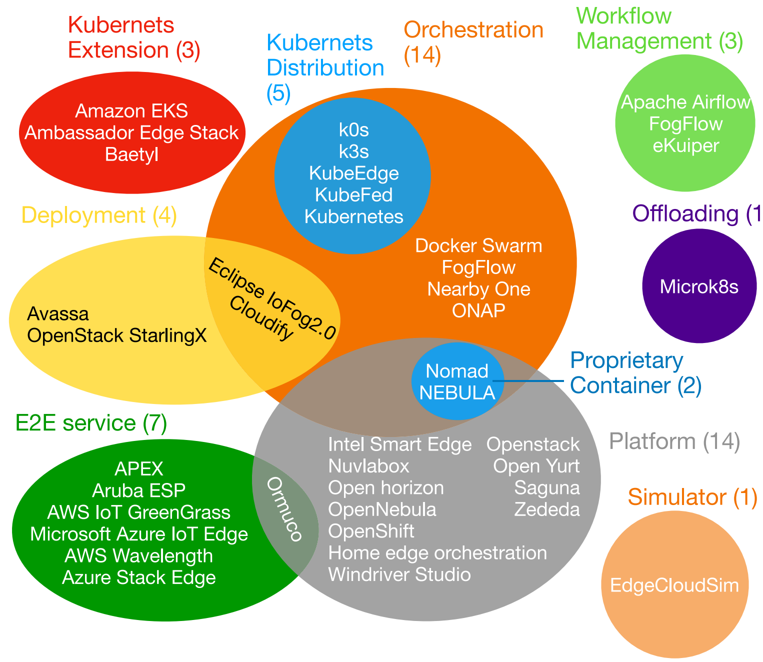

Fig. 2. Tools classification based on their characteristics.

- **Proprietary Container:** tools that allow the creation of containerized applications.
- **Deployment:** tools used to deploy containerized and non-containerized applications across the cloud continuum. These tools create an environment where the containerized applications can be managed after the deployment.
- **Kubernetes Distribution:** complete Kubernetes platforms designed to provide an out-of-the-box solution for deploying Kubernetes and managing containerized workloads with pre-configured solutions.
  - **k0s:** minimalist distribution designed to reduce complexity packaged into a single binary
  - **k3s:** lightweight distribution similar to k0s. It also includes some features by default, such as an ingress controller and load balancer.
  - **KubeEdge:** k8s distribution proposed to be extended for edge devices. It includes device management and MQTT integration.
  - **KubeFed:** k0s is designed to unify the lifecycle management of a multi-cluster workload. Therefore it is a unified server that distributes the objects towards multiple clusters.
  - **Kubernetes:** classic distribution also known as k8s.
- **Kubernetes Extension:** additional components that extend the core functionality of Kubernetes. These extensions are developed and added by third-parties. To enhance the functionality of Kubernetes and provide additional capabilities for managing complex workloads. Kubernetes extensions are custom additions to a Kubernetes distribution, enhancing its capabilities with features like custom controllers or resources.
- **Simulator:** tools used to simulate specific environments for testing purposes. Such environments are useful when performing tests among different simulated entities and devices without deploying a complete hardware infrastructure.
- **End2End (E2E) Service:** fully managed services providing a set of functionalities to automate the full software lifecycle along the cloud continuum. They are usually based on a set of integrated tools aimed at providing an end-to-end solution.
- **Platform:** complete software environment that provides a set of services and tools for developing, deploying, and managing distributed applications. The goal of a platform is to provide a high-level abstraction of the underlying infrastructure, making it easier for developers to build and deploy distributed applications.
- **License:** the license adopted by the different tools. A tool can be based on a Commercial (Com) or released as an Open Source Software (OSS).

Fig. 2 illustrates the proposed tools classification based on their characteristics. From our extraction process, we discovered that only one tool is used to perform pure offloading: MicroK8s while 14 other tools are mostly used for orchestration purposes, and 2 for workflow management. On the same page, we discovered that the use of containers is of tremendous importance in this environment as 2 tools have been categorized as proprietary containers, and 4 tools are used to perform deployment. Most importantly, the high use of Kubernetes allowed us to label two specific categories: the Kubernetes distributions composed of 5 tools and the Kubernetes Extensions composed of 3 tools. While only one tool can be categorized as a Simulator, multiple End-to-End Services (i.e. 7) and Platforms (i.e. 14) have been discovered, with Ormuco being categorized as both of the latter. Out of the 40 retrieved tools, 23 have a commercial license, and 17 have an Open Source License. The mapping of each different tool to the different categories is depicted in the second part of Table 6. A complete list of tools and their URL is reported in Table 7.

5.3. Tools environment (RQ3)

Description. To answer RQ3, we identified the target environment for each tool.

Results. We categorized the different tools based on the main environment they target. We identified 4 different alternatives:

- **Agnostic:** tools created to run seamlessly on different entities among the cloud continuum. We identified 29 agnostic tools.
- **Cloud Infrastructure:** tools that target the Cloud as the main environment to be run. Out of 40 final tools, 6 of them could be run at cloud infrastructures.

- **Edge Node:** tools made to run on Edge Devices. We found 5 tools whose main environment is edge nodes.
- **Far Edge:** tools targeting the deployment on those devices which reside in the far edge and therefore have lower computation capabilities. Examples of far edge can be weather or remote sensors. Based on our results, only 1 tool was made to run on the far edge.

Among the tools analyzed, the vast majority can be cataloged as Agnostic (29 out of 40), the following 6 tools have been labeled as targeted for the Cloud, 4 for the edge nodes, while only eKuiper targets a deployment in the far edge.

The second part of Table 6 maps each different tool to its targeted environment.

6. Discussion

The results achieved in this work provide insight view of the state of the market for what concerns the ability to move the computational burden of software among different parts of the Cloud Continuum. Given the hype that has targeted cloud computing for more than a decade, as expected, most tools target the execution of tasks on the cloud. However, given the strict requirement that nowadays are valid for edge and far-edge devices, this cannot always be a solution, and therefore the necessity of orchestrators and offloaders becomes essential.

It is therefore surprising to find only a single tool that targets as its primary task the offloading of tasks being executed, which is clearly a consequence of the inherent technical hurdles that are involved in such a process. While only 2 tools are able to perform as workflow managers, many more tools are capable of performing orchestration. This shows that it is not always easy to dynamically adapt the computational tasks but is usually preferred to move the tasks before they actually start and plan in advance the execution and computation.

Predictably, a significant observation stemming from this MLR is the prominent role played by Kubernetes among the chosen tools. Kubernetes is primarily renowned for its orchestration capabilities, and its adoption varies according to user needs and the architectural requirements of the environment. In conjunction with this, numerous other tools leverage its capabilities to execute valuable orchestration tasks

For what concerns container technologies, from our analysis we can see that all of the agnostic and cloud infrastructure-based technologies are compliant with the Open Container Initiative (OCI) (Anon, 2023c) except for Nomad and NEBULA which have proprietary container structures.

On the design of the tools, we can say that the difficulties of creating an end-to-end tool are reflected by the producer of the tool as those are mostly owned by very big companies. Ormuco is the only exception to this rule, being a company employing 40 people only (2022 data) (Anon, 2023d).

Regarding target platforms, platform-agnostic ones are the most common. This is reflected by the fact that the closer you move to hardware the more difficult it gets to support heterogeneous hardware/software platforms. This is shown also by only a few platforms targeting mainly edge nodes, and only one for the far edge.

6.1. Feature challenges

The analysis of the literature shows that the design of systems capable of performing task offloading is still in its infancy. The tools are either targeted at a specific use case or are extensions of mainstream tools such as Kubernetes or containers. Moreover, the lack of interest in simulation hints that the scenarios under consideration are limited in size.

Most tools either provide coverage for end-to-end service or are full software platforms. This approach works against standardization and interoperability efforts, which would grow if it was possible to use different tools for the various actions related to task offloading and orchestration.

Moreover, as stated in the previous section, the difficulties in creating a system capable of dynamically adapting the computational tasks (i.e. Cognitive), is nowadays reflected by the presence of multiple works researching how to perform advanced orchestration mechanisms aiming at achieving offloading but not so many tools released on the matter. We believe that a future challenge in the field will be related to the creation of tools leveraging AI-based algorithms for offloading purposes.

6.2. Threats to validity

We are aware that our work is subject to threats to validity, since we got through only tools that are available either commercially or as OSS, while other tools could be used internally by large companies, or more advanced techniques could be developed and close to reaching the maturity level required to be part of a tool such as the ones included in this analysis.

One more issue, that we tried to solve with a thorough analysis, was that some tools could have been abandoned, and slowly growing obsolescent.

With regards to the tools' features, we based our analysis on the material available as a bibliography, and on the tools' website, not considering experimental features. This approach can be too defensive and produce several false negatives, nevertheless, we preferred this issue instead of accepting characteristics and capabilities not mature enough for tools that can be used in a production environment.

To improve the reliability of this work, we defined search terms and apply procedures that can be replicated by others. Since this is a mapping study and no systematic review, the inclusion/exclusion criteria are only related to whether the topic of Cognitive Cloud is present in a paper or not, as suggested by Petersen et al. (2008).

As for the analysis procedure, since our analysis only uses descriptive statistics, the threats are minimal. However, we are aware that the synthesis of the definition might be subjective. To mitigate this threat, the analysis was done collaboratively, using a collecting coding method, and discussing with all the authors about inconsistencies. The Kohen K index about our disagreement also confirms the quality of the qualitative analysis performed.

7. Conclusion

This paper presents the results of a systematic mapping study to classify the edge-to-cloud tools in the cognitive cloud continuum. We conducted a multivocal literature review considering 40 tools from 1073 primary studies (220 from the white literature and 853 from the grey literature), as presented in Section 5.

One of our main findings is that 85% of the tools can perform offloading, orchestration, or other computational tasks from edge-to-cloud while the rest can execute on the edge only.

This work provides a valuable comparison of edge-to-cloud tools and their characteristics that can be used by researchers and practitioners to select the proper tool for their purpose. Such characteristics include the nature of the tools, but also the environment target for the deployment and its license.

Future works include and empirical evaluation of pros and cons of each tool by running an industrial survey and a set of industrial use-cases.

Table 7
Tools main reference and their license.

Tool	URL
Amazon EKS	https://aws.amazon.com/eks/features/
Ambassador Edge Stack	https://www.getambassador.io
Apache Airflow	https://airflow.apache.org
APEX (for Dell Technologies)	https://www.dell.com/en-us/dt/apex/
Aruba Edge Services Platform (ESP)	https://www.arubanetworks.com/solutions/aruba-esp/
Avassa	https://avassa.io
AWS IoT GreenGrass	https://aws.amazon.com/greengrass/
AWS Wavelength	https://aws.amazon.com/wavelength/
Azure stack Edge	https://azure.microsoft.com/en-us/products/azure-stack/edge
Baetyl	https://baetyl.io/en/
Cloudify	https://cloudify.co/
Docker Swarm	https://docs.docker.com/engine/swarm/
Eclipse ioFog 2.0	https://iofog.org
EdgeCloudSim	https://github.com/Cagatay Sonmez/EdgeCloudSim
eKuiper	https://ekuiper.org/
FogFlow	https://github.com/smartfog/fogflow
Home edge orchesterator	https://www.lfedge.org/projects/homeedge/
Intel Smart Edge	https://smart-edge-open.github.io
k0s	https://k0sproject.io/
K3s	https://k3s.io/
KubeEdge	https://kubeedge.io/en/
KubeFed	https://github.com/kubernetes-sigs/kubefed
Kubernetes	https://kubernetes.io/
MicroK8S	https://microk8s.io/
Microsoft's Azure IoT Edge	https://azure.microsoft.com/en-us/products/iot-edge
Nearby One	https://www.nearbycomputing.com/nearbyone/
NEBULA	https://nebula-orchestrator.github.io/
Nomad	https://www.nomadproject.io/
Nuvlabox (NuvlaEdge)	https://nuvla.io/ui/sign-in
ONAP (Open Network Automation Platform)	https://www.onap.org/
Open horizon	https://www.lfedge.org/projects/openhorizon/
Open Stack Starlingx	https://www.starlingx.io/
OpenNebula Edgify	https://opennebula.io/edgify-opennebula-as-a-service/
OpenShift	https://www.redhat.com/en/technologies/cloud-computing/openshift
Openstack	https://www.openstack.org/
OpenYurt	https://openyurt.io/
Ormuco	https://ormuco.com
Saguna	https://www.saguna.net/product/saguna-edge-to-cloud/
StudioGA (WINDRIVER STUDIO)	https://www.windriver.com/studio
Zededa	https://zededa.com/

CRediT authorship contribution statement

Sergio Moreschini: Conceptualization, Methodology, Writing – original draft.
Elham Younesian: Conceptualization, Methodology, Writing – original draft.
David Hästbacka: Conceptualization, Writing – review & editing, Supervision, Funding acquisition.
Michele Albano: Conceptualization, Supervision reviewing and editing.
Jiří Hošek: Conceptualization, Supervision reviewing and editing.
Davide Taibi: Conceptualization, Supervision reviewing and editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

link to the replication package is provided in the paper.

Acknowledgment

This work is supported by projects IndustryX and 6GSoft both funded by Business Finland.

Appendix A. The selected papers

[SP1] Georgakopoulos, A. et al. (2012). Cognitive cloud-oriented wireless networks for the Future Internet. In 2012 WCNCW (pp. 431–435). IEEE.
[SP2] Cai, W. et al. (2014). Environment Perception for Cognitive Cloud Gaming. In International Conference on Cloud Computing (pp. 3–13). Springer, Cham.
[SP3] Cai, W. et al. (2014). Resource management for cognitive cloud gaming. In 2014 ICC (pp. 3456–3461). IEEE.
[SP4] Cordeschi, N. et al. (2015). Reliable adaptive resource management for cognitive cloud vehicular networks. IEEE Trans. on Vehicular Tech., 64(6), 2528-2537.
[SP5] Shi, W. (2015). QoE guarantee scheme based on cooperative cognitive cloud and opportunistic weight particle swarm. Electrical and Computer Engineering, 2015.
[SP6] Baughman, A. K. et al. (2015). Disruptive innovation: Large scale multimedia data mining. In Multimedia Data Mining and Analytics (pp. 3–28). Springer, Cham.
[SP7] Mahmoodi, S. E. et al. (2016). A time-adaptive heuristic for cognitive cloud offloading in multi-RAT enabled wireless devices. IEEE Transactions on Cognitive Communications and Networking, 2(2), 194-207.
[SP8] Chiang, M., et al. (2016). Fog and IoT: An overview of research opportunities. IEEE Internet of things journal, 3(6), 854-864.
[SP9] Mahmoodi, S. E. et al. (2018). Cognitive Cloud Offloading Using Multiple Radios. In Spectrum-Aware Mobile Computing (pp. 23–33). Springer, Cham.
[SP10] Wu, X. et al. (2018). Phase-compensation-based cooperative spectrum sensing algorithm for cognitive cloud networks. In 2018 ICOIN (pp. 755–759). IEEE.
[SP11] Wang, L. et al. (2018). Cooperative Spectrum Sensing Algorithm Based on Phase Compensation in Cognitive Cloud Networks. ICUFN (pp. 143–147).
[SP12] Huang, H. et al. (2018). On the Performance of Cognitive Cloud Radio Access Networks in the Presence of Hardware Impairment. In APSIPA ASC (427-431).
[SP13] Marshall, T. E. et al. (2018). Cloud-based intelligent accounting applications: accounting task automation using IBM watson cognitive computing. Journal of Emerging Technologies in Accounting, 15(1), 199-215.
[SP14] Jann, J. et al. (2018). IBM POWER9 system software. IBM Journal of Research and Development, 62(4/5), 6-1.

[SP15] Kloeckner, K. et al. (2018). Building a cognitive platform for the managed IT services lifecycle. IBM Journal of Research and Development, 62(1), 8-1.

[SP16] Mahmoodi, S. E. et al. (2019). Time-Adaptive and Cognitive Cloud Offloading Using Multiple Radios. In Spectrum-Aware Mobile Computing (pp. 49–66).

[SP17] Garai, Á. et al. (2019). Revolutionizing healthcare with IoT and cognitive, cloud-based telemedicine. Acta Polytechnica Hungarica, 16(2), 163-181.

[SP18] Amato, F. et al. (2019). A federation of cognitive cloud services for trusting data sources. In CISIS (pp. 1022–1031).

[SP19] Ferrer, A. J. et al. (2021). Towards a Cognitive Compute Continuum: An Architecture for Ad-Hoc Self-Managed Swarms. In CCGrid (pp. 634–641).

[SP20] Campolo, C. et al. (2021). Virtualizing AI at the distributed edge towards intelligent IoT applications. Journal of Sensor and Actuator Networks, 10(1), 13.

[SP21] Vermesan, O. et al. (2021). Internet of Vehicles–System of Systems Distributed Intelligence for Mobility Applications. In Intelligent Technologies for Internet of Vehicles (pp. 93–147). Springer, Cham.

[SP22] Kretsis, A. et al. (2021). SERRANO: Transparent Application Deployment in a Secure, Accelerated and Cognitive Cloud Continuum. In MeditCom (pp. 55–60)

[SP23] Bacciu, D. et al. (2021). TEACHING-Trustworthy autonomous cyber–physical applications through human-centered intelligence. In 2021 COINS (pp. 1–6).

[SP24] Zhang, H. et al. (2021). Knowledge-based systems for blockchain-based cognitive cloud computing model for security purposes. International Journal of Modeling, Simulation, and Scientific Computing, 2241002.

Appendix B. Tools list

See Table 7.

References

Anon, 2023a. ACM. https://dl.acm.org/.

Anon, 2023b. IEEE. https://ieeexplore.ieee.org/Xplore/home.jsp.

Anon, 2023c. Open container initiative. https://opencontainers.org.

Anon, 2023d. Ormuco. https://ormuco.com.

Anon, 2023e. Sciencedirect. https://www.sciencedirect.com/.

Anon, 2023f. Scopus. https://www.scopus.com/search/form.uri?display=basic#basic.

Anon, 2023g. Web of science. https://www.webofscience.com/wos/woscc/basic-search.

Aslanpour, M.S., Toosi, A.N., Cicconetti, C., Javadi, B., Sbarski, P., Taibi, D., Assuncao, M., Gill, S.S., Gaire, R., Dustdar, S., 2021. Serverless edge computing: Vision and challenges. In: 2021 Australasian Computer Science Week Multiconference. ACSW '21, Association for Computing Machinery, New York, NY, USA.

Du, K., Yang, H., Zhang, Y., Duan, H., Wang, H., Hao, S., Li, Z., Yang, M., 2020. Understanding promotion-as-a-service on GitHub. In: Annual Computer Security Applications Conference. ACSAC '20, Association for Computing Machinery, New York, NY, USA, pp. 597–610.

Emam, K.E., 1999. Benchmarking Kappa: Interrater agreement in software process assessments. Empir. Softw. Eng. 4 (2), 113–133.

Garousi, V., Felderer, M., Mäntylä, M.V., 2019. Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. Inf. Softw. Technol. 106, 101–121.

Han, J., Deng, S., Xia, X., Wang, D., Yin, J., 2019. Characterization and prediction of popular projects on GitHub. In: 2019 IEEE 43rd Annual Computer Software and Applications Conference, Vol. 1. COMPSAC, pp. 21–26.

Keele, S., et al., 2007. Guidelines For Performing Systematic Literature Reviews in Software Engineering. Technical Report, Ver. 2.3 Ebse Technical Report Ebse.

Mao, Y., Zhang, J., Letaief, K.B., 2016. Dynamic computation offloading for mobile-edge computing with energy harvesting devices. IEEE J. Sel. Areas Commun. 34 (12), 3590–3605.

Mao, Y., Zhang, J., Letaief, K.B., 2017. Joint task offloading scheduling and transmit power allocation for mobile-edge computing systems. In: 2017 IEEE Wireless Communications and Networking Conference. WCNC, IEEE, pp. 1–6.

Mell, P., Grance, T., et al., 2011. The NIST Definition of Cloud Computing. Computer Security Division, Information Technology Laboratory, National ....

Moreschini, S., Pecorelli, F., Li, X., Naz, S., Albano, M., Hästbacka, D., Taibi, D., 2023. Cognitive cloud: The definition. In: Omatu, S., Mehmood, R., Sitek, P., Cicerone, S., Rodríguez, S. (Eds.), Distributed Computing and Artificial Intelligence, 19th International Conference. Springer International Publishing, Cham, pp. 219–229.

Moreschini, S., Pecorelli, F., Li, X., Naz, S., Hästbacka, D., Taibi, D., 2022. Cloud continuum: The definition. IEEE Access 10, 131876–131886.

Nupponen, J., Taibi, D., 2020. Serverless: What it is, what to do and what not to do. In: 2020 IEEE International Conference on Software Architecture Companion. ICSA-C, pp. 49–50.

Petersen, K., Feldt, R., Mujtaba, S., Mattsson, M., 2008. Systematic mapping studies in software engineering. In: 12th International Conference on Evaluation and Assessment in Software Engineering. EASE 12, pp. 1–10.

Rosendo, D., Costan, A., Valduriez, P., Antoniu, G., 2022. Distributed intelligence on the edge-to-cloud continuum: A systematic literature review. J. Parallel Distrib. Comput. 166, 71–94.

Saeik, F., Avgeris, M., Spatharakis, D., Santi, N., Dechouniotis, D., Violos, J., Leivadeas, A., Athanasopoulos, N., Mitton, N., Papavassiliou, S., 2021. Task offloading in edge and cloud computing: A survey on mathematical, artificial intelligence and control theory solutions. Comput. Netw. 195, 108177.

Sahni, Y., Cao, J., Zhang, S., Yang, L., 2017. Edge mesh: A new paradigm to enable distributed intelligence in Internet of Things. IEEE Access 5, 16441–16458.

Sardellitti, S., Barbarossa, S., Scutari, G., 2014. Distributed mobile cloud computing: Joint optimization of radio and computational resources. In: 2014 IEEE Globecom Workshops. GC Wkshps, IEEE, pp. 1505–1510.

Singh, K., Awasthi, A., 2013. Quality, reliability, security and robustness in heterogeneous networks. In: 9th International Conference. Qshine, Springer, p. 303.

Stackpath, 2023. What is edge orchestration? https://www.stackpath.com/edge-academy/what-is-edge-orchestration/.

Tomarchio, O., Calcaterra, D., Modica, G.D., 2020. Cloud resource orchestration in the multi-cloud landscape: a systematic review of existing frameworks. J. Cloud Comput. 9, 1–24.

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M.C., Regnell, B., Wesslén, A., 2012. Experimentation in Software Engineering. Springer Science & Business Media.

Xiao, W., He, H., Xu, W., Tan, X., Dong, J., Zhou, M., 2022. Recommending good first issues in GitHub OSS projects. In: Proceedings of the 44th International Conference on Software Engineering. ICSE '22, Association for Computing Machinery, New York, NY, USA, pp. 1830–1842.

Yang, L., Cao, J., Yuan, Y., Li, T., Han, A., Chan, A., 2013. A framework for partitioning and execution of data stream applications in mobile cloud computing. ACM SIGMETRICS Perform. Eval. Rev. 40 (4), 23–32.

Zhang, J., Hu, X., Ning, Z., Ngai, E.C.-H., Zhou, L., Wei, J., Cheng, J., Hu, B., 2017. Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks. IEEE Internet Things J. 5 (4), 2633–2645.

Zhao, T., Zhou, S., Guo, X., Niu, Z., 2017. Tasks scheduling and resource allocation in heterogeneous cloud for delay-bounded mobile edge computing. In: 2017 IEEE International Conference on Communications. ICC, IEEE, pp. 1–7.

Sergio Moreschini is a Postdoctoral Researcher in the Faculty of Information Technology and Communication Sciences, Tampere University, Tampere, 33720, Finland, where he is a researcher in the Cloud Software Evolution and Assessment group. His research interests include MLOps, Cloud Continuum. Contact him at sergio.moreschini@tuni.fi.

Elham Younesian is a Double Degree Ph.D. candidate in the Faculty of Electrical Engineering and Communication, Brno University of Technology, Czech Republic, and Faculty of Information Technology and Communication Sciences, Tampere University, Finland. She received a master's degree in Electrical Engineering from the Polytechnic University of Turin, Italy. She is also a researcher at WISLAB Research Group. Her research interests include enhancing network functionalities through AI/ML, in particular for the 5G and beyond networks and non-terrestrial networks (NTN). She also contributes actively to the domain of AIenabled edge offloading. Contact her at Elham.younesian@vut.cz

David Hästbacka is an Associate Professor (tenure track) in Software Engineering at Tampere University, Finland. His research interests include system and software architectures and interoperability of software systems in production and energy systems applications. He received his D.Sc. (Tech.) degree from Tampere University of Technology in 2013. Contact him at david.hastbacka@tuni.fi.

Michele Albano is an associate professor for the Department of Computer Science at Aalborg University, Denmark. His research focuses on distributed systems and embedded systems, in particular for the application areas of industrial informatics and the smart grid. He received his Ph.D. from University of Pisa, Italy. Contact him at mialb@cs.aau.dk

Jiri Hosek (hosek@vut.cz) received his M.S. and Ph.D. degrees in electrical engineering from the Faculty of Electrical Engineering and Communication at BUT in 2007 and 2011, respectively. He is currently an associate professor (2016) and Deputy Vice-Head for R&D and International Relations in the Department of Telecommunications (2018), BUT. He is also coordinating the WISLAB research group (2012), where he deals mostly with industry-oriented projects in the area of IoT and home automation services. He has (co-)authored more than 130 research works on networking technologies, wireless communications, quality of service, quality of experience, and IoT applications.

Davide Taibi is full professor on Cloud Software engineering at Oulu and Tampere university, Finland. His research interests include empirical software engineering applied to cloud-native systems. Taibi received his Ph.D. in computer science from University of Insubria. He is a member of the International Software Engineering Network. Contact him at davide.taibi@tuni.fi.