Inter-operability and Orchestration in Heterogeneous Cloud/Edge Resources: The ACCORDION Vision

Ioannis Korontanis*
Harokopio University of Athens
Kallithea, Greece

Konstantinos Tserpes
Harokopio University of Athens
Kallithea, Greece

Maria Pateraki
ORamaVR S.A. & National Technical
University of Athens
Heraklion, Crete, Greece

Lorenzo Blasi
Hewlett Packard Italiana s.r.l.
Cernusco su Naviglio, Italy

John Violos
National Technical University of
Athens
Zografou, Greece

Ferran Diego
Telefonica Research
Barcelona, Spain

Eduard Marin
Telefonica Research
Barcelona, Spain

Nicolas Kourtellis
Telefonica Research
Barcelona, Spain

Massimo Coppola
ISTI-CNR
Pisa, Italy

Emanuele Carlini
ISTI-CNR
Pisa, Italy

Zbyszek Ledwoń
Orbital Knight
Warsaw, Poland

Przemysław Tarkowski
Orbital Knight
Warsaw, Poland

Thomas Loven
Plexus Tech
Santiago de Compostela, Spain

Yago González Rozas
Plexus Tech
Santiago de Compostela, Spain

Mike Kentros
ORamaVR S.A.
Heraklion, Crete, Greece

Michael Dodis
ORamaVR S.A.
Heraklion, Crete, Greece

Patrizio Dazzi
ISTI-CNR
Pisa, Italy

ABSTRACT

This paper introduces the ACCORDION framework, a novel framework for the management of the *cloud-edge continuum*, targeting the support of NextGen applications with strong QoE requirements. The framework addresses the need for an ever expanding and heterogeneous pool of edge resources in order to deliver the promise of ubiquitous computing to the NextGen application clients. This endeavor entails two main technical challenges. First, to assure interoperability when incorporating heterogeneous infrastructures in the pool. Second, the management of the largely dynamic pool of edge nodes. The optimization of the delivered QoE stands as the core driver to this work, therefore its monitoring and modelling comprises a core part of the conducted work. The paper discusses the main pillars that support the ACCORDION vision, and provide a description of the three planned use case that are planned to demonstrate ACCORDION capabilities.

CCS CONCEPTS

• **Computer systems organization** → **Cloud computing**.

KEYWORDS

Architecture; Cloud; Edge; Federation

1 INTRODUCTION

Cloud computing has played and it is still playing a key role in the digital revolution. Clouds are the real-world materialization of the long-lasting ambition of achieving utility computing [33]. Utility computing aspires to provision computing, storage and network resources in the same way common utilities are provided to end-users. Thanks to clouds, this has been finally possible. However, a big class of applications is currently being blocked because of their dependency on on-site infrastructures or specialised hardware

but also because they are too latency-sensitive or data-dependent to be moved to the public cloud. Several researches in the past worked on providing QoS guarantees [3, 5] to applications running on Cloud infrastructures, however, there are intrinsic limitations due to the nature of Cloud infrastructures, such as potential high latency and reduced or variable bandwidth availability. Instead, Next Generation (NextGen) applications would benefit from an advanced infrastructure with ubiquitous presence, unblocking them from fixed geographies. However, current edge computing implementations support only specific locations, which means that the scope of local resources and infrastructures are also restricted to the needs of certain edge-enabled applications (e.g., AWS Wavelength [1]). Moreover, existing solutions lead to user lock-in and, overall, have a negative impact on the open diffusion of edge computing. To overcome these limitations, ACCORDION[2] project aspires to provide an open low-latency, privacy-preserving, secure, and robust virtualized infrastructure, encompassing clouds and network infrastructures as well as resources owned by end-users. ACCORDION achieves this goal by opportunistically extending the infrastructure pool with all sorts of available compute and network resources and infrastructures; be it a telco operator, local commodity hardware, and private clouds, or end-devices, leaving the public cloud as a last resort.

As a matter of fact, relying on such a complex, dynamic, heterogeneous and distributed set of resources may be excessively complex and thus, impracticable, if the management of the consequent burden is left to application developers and end-users. To address this issue ACCORDION provides a distributed enabling platform organized over three layers, each served by a specifically devoted framework aimed at a different goal.

- **Edge Infrastructure Pool Framework** within which are provided the tools managing the resources expansion and facilitating the conditions to enable a volatile and "sparse" pool of low-latency, heterogeneous edge resources and infrastructures.
- **Edge/Cloud continuum management framework** designates the action to be enacted to support the NextGen applications in terms of resource assignment, availability, reliability, security, privacy and performance at the various sites where the users will reside. The continuum management framework creates a continuum of resources that virtually "follows" the users leaving a perception of persistent performance and high QoE, whilst mitigating the lack of reliability and availability in the infrastructure as well as providing strong security and privacy guarantees.
- Finally, the **Application Management Framework** will support the development of NextGen applications, via an integrated environment exposing advanced DevOps automation.

An expected byproduct of ACCORDION is in providing application developers and local infrastructure owners the chance to retain some of the revenue that is currently directed towards mainly non-EU public cloud providers. These conditions are explored by ACCORDION project in the context of a techno-economic analysis

for a viable federation of edge and cloud resource and infrastructure owners.

The remaining of this paper is organized as follows. Section 2 details the ACCORDION platform vision, also contextualizing the ACCORDION frameworks in the scientific literature. Section 3 introduces the conceptual architecture of the ACCORDION project, while Section 4 discusses ACCORDION use cases. Finally, Section 5 draws the conclusion and plans future work.

2 ACCORDION PLATFORM VISION

As aforementioned, this section is aimed at providing a viewpoint on the ACCORDION platform, overall. As such, it revolves around three main pillars that are representative of the key activities that are conducted in the context of the ACCORDION Project to pursue the achievement of an enabling platform.

2.1 Maximize the edge resource pool

To ensure low latency between end-user devices and the computational platform, ACCORDION deploys application components (e.g., containers, Unikernels, etc.) on resources that are close to the end-users. These computational resources are the edge nodes. Edge nodes do not match the computational capabilities of traditional clouds. The limited capacity of edge nodes make them expensive and critical, so they need to be properly exploited. The edge resource pool has also to be dynamic, to cope with the unpredictable availability of this kind of resources at run-time. ACCORDION has to overcome these obstacles in order to deploy application components on edge and cloud nodes.

To ease the efficient management of edge resources, ACCORDION introduces the abstraction of the mini-cloud: it is a set of resources that is available at the edge that can host instances of application components. ACCORDION infrastructure consists of a federation of mini-clouds. Infrastructure owners could use the ACCORDION platform to let his/hers edge nodes available within the federation. As such, those edge nodes may become available on an opportunistic fashion.

Edge mini-clouds will form "islands" in the sparse edge infrastructure pool at the respective sites, organizing and providing the underlying resources in a seamless way. Each mini-cloud is managed through the ACCORDION *Virtual Infrastructure Manager* (VIM), regardless of the kind of its underlying resources. The VIM will use K3s for container orchestration. K3s is a lightweight version of Kubernetes, which is designed to run on clusters that can be made up of hosts with limited resources, as it often happens in the IoT scenario. ACCORDION will also offer the possibility to run Virtual Machine images that contain application components; for this purpose the VIM uses the KubeVirt framework, a Kubernetes extension that manages KVM-based VMs.

As it is analyzed by Vaquero [29], edge computing brings additional challenges to those already impacting cloud orchestration [6, 13]. Volatility is inherent to edge resources, it impacts services availability [4], requires dynamic discovery [7, 8, 12, 14–16, 21, 22] to exploit new resources as soon as they are available, and quickly makes obsolete the status information of all resources. Heterogeneity is also a characterizing trait of Edge platforms, which are based on different types of resources, each with different access

---

[1]https://aws.amazon.com/it/wavelength/
[2]https://www.accordion-project.eu/

methods and protocols, potentially managed by different administrative domains. Another challenge results from the scale of the Edge: it is composed of a large number of small resources, and this, coupled with the ever smaller execution units of the applications, makes it difficult for orchestrators that are not properly designed to take optimal decisions. A method proposed by Wen [32], called late calibration, deploys the first solution found, even if sub-optimal, and then improves it with further calculations based on updated information to possibly migrate/redeploy part of the application. Similar approaches are labelled as *Eventually Consistent/Probabilistic Orchestration* and employ similar incremental techniques to progressively take the system closer to the desired state. It appears that such incremental techniques are the best solution to cope with the challenges posed by the Edge. Incremental techniques are typically based on a machine-readable description of the desired state of the system, usually in text-based configuration files, and lead to the so-called "declarative paradigm" that is the same on which many current DevOps tools, such as Kubernetes, are based.

2.2 Maximize robustness of Cloud/Edge compute continuum

As described in the previous section, the resources belonging to the ACCORDION federation participate in the cloud/edge continuum to support the application deployment and execution.

The resource orchestrator is the component that drives the match-making process resulting in the actual assignment of resources to specific instances of application components.In this attempt (matching application components with the appropriate resources) it is driven by the information that developers provide to the ACCORDION Platform by means of a proper usage of the ACCORDION application model, that ease the process of providing descriptions of the hardware and network requirements. Additionally ACCORDION needs a description of its resources also to be able to do the matching between components and resources, Di Cosmo [17], proposed to have three types of files: i) a specification file that contains components and their constraints, ii) a universe file that has a representation of the available services and VMs in JSON format and iii) a configuration file that describes the system level data, number of VMs and actions that are needed for a service, some of those data may have to be retrieved from users or other configuration files. For this case ACCORDION has the characterization system which provides information about the hardware and the location or resources (VMs, RaspberryPis and PCs),the orchestrator should consider both characterization and the application model to do the match-making and the deployment plan targeting the VIM (e.g., structured in a way compliant with K3s).

The provided application model uses an extended TOSCA grammar.TOSCA is an open standard by OASIS (Topology and Orchestration Specification for Cloud Applications) and its purpose is to describe the components of an application and the way they are communicating in the cloud to provide appropriate results. Generally TOSCA is able to describe services and applications hosted on the cloud including their components, relationships, dependencies, requirements, and capabilities. Related to the match-making between components and resources a feature that our extended

TOSCA grammar provides is to indicate if an application component instance needs to be deployed at the Edge or in the cloud. The idea to indicate where an application component should be placed was presented by Petrovic [25] who automated the deployment of containers at the Edge and the Cloud in Docker Swarm, by using an general description XML file which has an execution environment description that described if the container should be deployed at Edge or Cloud and then this XML file was converted to a Docker Swarm script.

In conducting its match-making process, the orchestrator takes into account the applications' QoS/QoE requirements as well as their internal topologies. When a new QoE/QoS violation is detected by ACCORDION services, the resource orchestrator modifies or composes a new service allocation plan and executes the appropriate actions (e.g., migration, scale up/down, etc.).

An additional ACCORDION mechanism, the Resilience Policy, works proactively and predicts a potential deterioration of QoE/QoS [31] by leveraging resource usage metrics such as CPU, RAM and bandwidth to predict when the available resources will not be sufficient. This proactive mechanism provides useful information and triggers the orchestrator for a migration between neighboring hybrid edge mini-clouds and dynamic horizontal scaling. This approach makes an intelligent replication that focuses on the bottlenecks of the infrastructure and outperforms the conventional replication techniques. The "intelligence" is achieved using Recurrent Neural Networks, like LSTM and GRU, that outperform traditional statistical time series forecasting models.

A part of the resource orchestrator deals with network specific aspects. It is the one who assures that there are no violations of the network requirements of an application. It provides a multi-domain AI-based orchestration framework of the network elements by ensuring reliability and latency. It provides automated orchestration and intelligent management operations and facilitates the life cycle management of network slices with the aim of rapid slice creation and activation.Network resource orchestrator is able to perform SDN operations to reconfigure the network and achieve better QoE/QoS. To enable self-configuration and self-optimization capabilities of network resources, this component considers the exploitation of machine learning techniques and their integration. By studying several ML techniques, the results showed that Distributed Artificial Intelligence and Federated Learning [18, 27, 28] promise to solve the problems related to coordination, concurrency, and decision making to obtain AI-based orchestration. ACCORDION adopts hyper-parameter optimization techniques to Deep Reinforcement Learning [23] to find close to optimal solutions and will investigate the application of multi-agent reinforcement learning [34] when the decision should be taken by multiple actors.

Before deploying a given application within the ACCORDION infrastructure, several tools are leveraged to ensure security and privacy within the container ecosystem in each mini-cloud and public cloud. These tools aim to significantly reduce the container security weaknessOne of such tools is intended to identify the minimum set of capabilities containers need to provide for executing their applications correctly while minimising their interactions with the OS kernel. In fact, according to the NIST container security guidelines [1], reducing the attack surface of the host OS kernel is a very promising way to mitigate the fragile isolation

guarantees of containers. In addition to this, another tool studies the feasibility of uniquely identifying containerised applications solely based on their syscall patterns. This is particularly relevant for cloud providers that offer confidential cloud computing services to their customers, as in such a case it should be impossible for cloud providers to infer any information about the applications running inside containers. In case a containerised application exhibits a distinctive syscall pattern, obfuscation techniques that rely on carefully adding dummy syscalls to reduce the container's syscall pattern uniqueness are used.

A further collection of tools enables privacy preservation. The goal of these tools is to optimize Machine Learning (ML) models such that they converge well and offer high ML performance, while preserving the privacy of the data owners. To this end, ACCORDION employs hardware-based Trusted Execution Environments (TEEs) to preclude well-known ML-based attacks such as membership of attribute inference attacks and data reconstruction attacks. TEEs allow to securely store data and execute arbitrary code on an untrusted device almost at native speed through secure memory compartments. However, to keep the Trusted Computing Base (TCB) as small as possible, current TEEs have limited memory, which makes it impossible to simultaneously place all ML layers inside the TEE while training them. Therefore, ACCORDION plans to leverage a recently proposed solution: PPFL [24], a TEE-based framework for Privacy-Preserving Federated Learning that protects against privacy attacks while considering the current limitations of TEEs. In addition to this, ACCORDION envisions the exploitation of FLaaS [20], a Federated Learning as a Service platform, which enables different scenarios of 3rd-party application collaborative model building which addresses the challenges of permission and privacy management, usability, and hierarchical model training.

2.3 Minimize overheads in migrating applications to cloud/edge federation

ACCORDION attempts to support developers in tailoring their applications to its platform. In doing this, it provides a properly defined application model. The application model primarily aims at supporting the ACCORDION platform in the resource provisioning process, but secondarily it drives the decisions about the actual deployment. Furthermore, it is designed to enable the declaration of application QoS/QoE requirements. The application workflows can be also described by properly defined means, as well as application life-cycle management. There is an instance of the application model for each scenario of the Use Case that describes perform deployment, run-time adaptation, scaling, etc.

The application providers leveraging ACCORDION do not have to know how ACCORDION components interact to deploy, start, and manage the application, they only have to describe the hardware requirements, relationships, dependencies with Edge or Cloud, workflows of their application components within the application model that expresses these features through the extended TOSCA grammar. As it is mentioned by Binz [10] the goals of TOSCA are i) the automation of application deployment, ii) to represent the application in a cloud agnostic way, and iii) the inter-operability and re-usability of components.

ACCORDION aims to benefit from these goals and use TOSCA as an application model which contains a generic description which then has to be converted to a deployment plan, a similar approach has been introduced before in BASMATI [2, 30], CloudCAMP [9] and Tosker [11]. TOSCA has been used before as a general description which was then converted to Docker or Kubernetes deployment plans. A synergy of TOSCA and Docker is Tosker that parses an extended TOSCA YAML description file of a multi-component application to deploy it in Docker. For the case of Kubernetes and Kim [19] used TOSCA to define application components along with their deployment and run-time adaptation on Kubernetes clusters across different countries, which is a case very similar with the one of ACCORDION. As we mentioned before, ACCORDION enable the description of workflows with an extended TOSCA grammar, this idea is inspired by the work of Qasha [26], which used Cloudify's extended TOSCA grammar to describe workflows.

3 ARCHITECTURE

A conceptual architecture is drawn in order to facilitate the vision outlined in the previous section. The architecture (Fig. 1) is organized in three layers, that are established upon the physical layer. The latter is comprised of public cloud and edge resources and infrastructures. The motivation is to host NextGen applications on top of the edge infrastructure pool and only use the public cloud as a capacitor, filling up the shortcomings – in type and volume of resources – of the edge part. However, this interplay requires the elevation of the edge infrastructure pool to a state of abstraction equivalent to that of the public cloud (Layer 1). In this layer the provided mechanisms are VIM, resource monitoring, resource indexing, edge storage and lightweight virtualization. Then, intelligent algorithms can ensure that the compute and network edge/cloud continuum will be robust, i.e., secure and privacy-preserving, orchestrator, resilience policies and mechanisms (Layer 2). On top of those layers, there are tools that are meant to facilitate NextGen application migration and deployment (Layer 3). In this layer, ACCORDION provides mechanisms and tools like the application model and data services, dynamic QoE assessment, DevOps to support application deployment and to orchestrate network paths at the edge.

A concrete version of this conceptual architecture is under development in the ACCORDION project. As it clearly results from both Figure 1 as well as from the description provided in Sec. 2, ACCORDION Platform is composed by a complex and articulated set of components and artifacts whose detailed description goes beyond the aim of this paper.

4 ACCORDION USE CASES

The ACCORDION Platform is planned to be the materialization of the ACCORDION Vision, as such it is aimed to be as much general as possible and able to satisfy a large set of different types of applications. Despite this fact, its actual realization is taking into account the specific needs of the three applications that will serve as project use cases. As a matter of fact, the developers of these applications are behaving as early adopters of the ACCORDION Platform and also playing a relevant role in the effective design and tailoring of the platform to the needs of their applications.
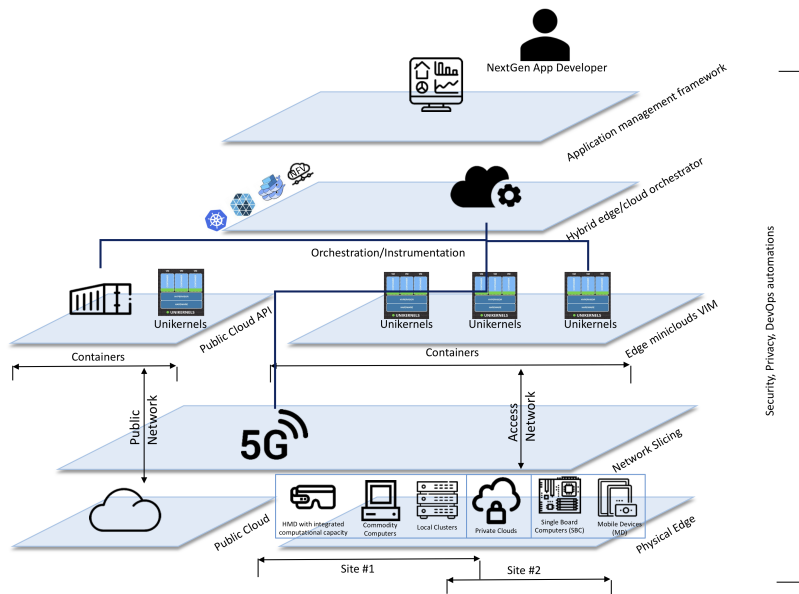
Figure 1: ACCORDION conceptual architecture

These applications are called to "demonstrate" the viability of the ACCORDION Vision from the perspective of real world scenarios.

The three ACCORDION use cases deal with: *(i)* Virtual Reality for Education purposes (Sec. 4.1, *(ii)* Online Video Games (Sec. 4.2) and *(iii)* Optimized Content Delivery (Sec. 4.3)

4.1 Collaborative Virtual Reality Training

The use case aims at supporting collaborative Virtual Reality training applications[3] specially formulated for untethered Head-Mounted Displays (HMDs) and the adaptation of the networking layer to edge computing, in order to optimize the cooperative mode, to ensure lower latency and higher performance on average network conditions, to minimize host dependency and ultimately to reach a higher number of Concurrent Users. In order to exploit the functionalities of the ACCORDION Platform, the application is redesigned to enable offloading of certain computations across the edge mini-cloud and/or in public clouds, eliminating dependencies on proprietary API's that are linked to specific HMD vendors.

4.2 Multiplayer mobile Games

The use case addresses multiplayer mobile gaming[4]. The game servers will be deployed on top of the ACCORDION Infrastructure to meet the requirements of NextGen mobile online gaming, which aims to lower the latency between servers and clients to highly improve the user experience. It will also take advantage of the AI-based orchestration to dynamically and automatically deploy new servers based on performance metrics and the player's geographical location. The game system will consist of two elements: the game server and a mobile application. Mobile application instances will be run by the end-users – players on their own mobile devices. The

game system must be able to handle up to 100 players and a large number of in-game events while performing full simulation of the game world and generating responses with minimal possible delay. The ACCORDION Platform must optimally scale its resources in order to match the number of current players.

4.3 Optimized Content Delivery

This use case consists of the client identification for adapted digital signage and mobile augmented reality game[5] sub use cases.

For the adapted digital signage, information is collected from the clients' mobile devices located in an defined space (like a mall) and the related application component derives client identification information from the mobile devices to dynamically adapt the content displayed on screen. For instance, in case there are mostly young players, the actual content will be adapted to show something that is most likely of their interest.

The game is an augmented reality customer loyalty game where points can be won based on clients playing against each other. Depending on the interaction between the user and actual coordinates, the game scenario dynamically changes.

5 CONCLUSION AND FUTURE WORK

This paper introduces the vision at the basis of the ACCORDION Project, that advocates the approach of extending and managing the resource pool of a Cloud/Edge federation with devices placed at the edge of the network, opportunistically recruited. The actual exploitation of such resources enable the execution of NextGen applications (characterized by ultra low-latency requirements, specialized hardware and strict constraints on data) by means of an utility-like computing paradigm. In fact the consequent heterogeneity and dynamic behavior of resources has to be properly managed

---

[3]https://oramavr.com/
[4]https://www.orbitalknight.com/

[5]https://www.plexus.es/

to be efficiently exploited. ACCORDION Platform is structured over a three-layers architecture, enabling resource pooling, management, and ease of usage, respectively.

In the near future we plan to release a detailed concrete architecture of the system, a first integrated version of the ACCORDION Platform, that will be validated and evaluated against the three use cases that have been introduced in Section 4. The ultimate goal will be to demonstrate the features and capabilities of the ACCORDION platform under very different perspectives.

ACKNOWLEDGMENTS

REFERENCES

[1] 2017. NIST Special Publication 800-190. Application Container Security Guide. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-190.pdf.

[2] Jörn Altmann, Baseem Al-Athwari, Emanuele Carlini, Massimo Coppola, Patrizio Dazzi, Ana Juan Ferrer, Netsanet Haile, Young-Woo Jung, Jamie Marshall, Enric Pages, Evangelos Psomakelis, Ganis Zulfa Santoso, Konstantinos Tserpes, and John Violos. 2017. BASMATI: An Architecture for Managing Cloud and Edge Resources for Mobile Users. In *Economics of Grids, Clouds, Systems, and Services*, Congduc Pham, Jörn Altmann, and José Ángel Bañares (Eds.). Springer International Publishing, Cham, 56–66.

[3] Gaetano F Anastasi, Emanuele Carlini, Massimo Coppola, and Patrizio Dazzi. 2017. QoS-aware genetic cloud brokering. *Future Generation Computer Systems* 75 (2017), 1–13.

[4] Gaetano F Anastasi, Emanuele Carlini, and Patrizio Dazzi. 2013. Smart cloud federation simulations with cloudsim. In *Proceedings of the first ACM workshop on Optimization techniques for resources management in clouds*. 9–16.

[5] Gaetano F Anastasi, Massimo Coppola, Patrizio Dazzi, and Marco Distefano. 2016. QoS guarantees for network bandwidth in private clouds. *Procedia Computer Science* 97 (2016), 4–13.

[6] Ranieri Baraglia, Gabriele Capannini, Patrizio Dazzi, and Giancarlo Pagano. 2013. A multi-criteria job scheduling framework for large computing farms. *J. Comput. System Sci.* 79, 2 (2013), 230–244.

[7] Ranieri Baraglia, Patrizio Dazzi, Barbara Guidi, and Laura Ricci. 2012. GoDel: Delaunay Overlays in P2P Networks via Gossip. In *IEEE 12th International Conference on Peer-to-Peer Computing (P2P)*. IEEE, 1–12.

[8] Massimiliano Bertolucci, Emanuele Carlini, Patrizio Dazzi, Alessandro Lulli, and Laura Ricci. 2015. Static and dynamic big data partitioning on apache spark.. In *PARCO*. 489–498.

[9] Anirban Bhattacharjee, Yogesh Barve, Aniruddha Gokhale, and Takayuki Kuroda. 2017. Cloudcamp: A model-driven generative approach for automating cloud application deployment and management. *Vanderbilt University, Nashville, TN, USA, Tech. Rep. ISIS-17-105* (2017).

[10] Tobias Binz, Uwe Breitenbücher, Oliver Kopp, and Frank Leymann. 2014. TOSCA: portable automated deployment and management of cloud applications. In *Advanced Web Services*. Springer, 527–549.

[11] Antonio Brogi, Luca Rinaldi, and Jacopo Soldani. 2018. TosKer: a synergy between TOSCA and Docker for orchestrating multicomponent applications. *Software: Practice and Experience* 48, 11 (2018), 2061–2079.

[12] Emanuele Carlini, Massimo Coppola, Patrizio Dazzi, Domenico Laforenza, Susanna Martinelli, and Laura Ricci. 2009. Service and resource discovery supports over p2p overlays. In *2009 International Conference on Ultra Modern Telecommunications & Workshops*. IEEE, 1–8.

[13] Emiliano Casalicchio. 2016. Autonomic Orchestration of Containers: Problem Definition and Research Challenges.. In *VALUETOOLS*.

[14] Massimo Coppola, Patrizio Dazzi, Aliaksandr Lazouski, Fabio Martinelli, Paolo Mori, Jens Jensen, Ian Johnson, and Philip Kershaw. 2012. The Contrail approach to cloud federations. In *Proceedings of the International Symposium on Grids and Clouds (ISGC'12)*, Vol. 2. 1.

[15] Marco Danelutto, Patrizio Dazzi, et al. 2005. A Java/Jini Framework Supporting Stream Parallel Computations.. In *PARCO*. 681–688.

[16] Patrizio Dazzi and Matteo Mordacchini. 2020. Scalable Decentralized Indexing and Querying of Multi-Streams in the Fog. *Journal of Grid Computing* 18, 3 (2020), 395–418.

[17] Roberto Di Cosmo, Antoine Eiche, Jacopo Mauro, Stefano Zacchiroli, Gianluigi Zavattaro, and Jakub Zwolakowski. 2015. Automatic Deployment of Services in the Cloud with Aeolus Blender. In *Service-Oriented Computing*, Alistair Barros, Daniela Grigori, Nanjangud C. Narendra, and Hoa Khanh Dam (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 397–411.

[18] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948* (2020).

[19] Dongmin Kim, Hanif Muhammad, Eunsam Kim, Sumi Helal, and Choonhwa Lee. 2019. TOSCA-based and federation-aware cloud orchestration for Kubernetes container platform. *Applied Sciences* 9, 1 (2019), 191.

[20] Nicolas Kourtellis, Kleomenis Katevas, and Diego Perino. 2020. FLaaS: Federated Learning as a Service. In *ACM DistributedML*.

[21] Alessandro Lulli Lucchese, Laura Ricci, Emanuele Carlini, Patrizio Dazzi, and Claudio. 2015. Cracker: Crumbling Large Graphs Into Connected Components. In *20th IEEE Symposium on Computers and Communication (ISCC) (ISCC2015)*. IEEE.

[22] Alessandro Lulli, Emanuele Carlini, Patrizio Dazzi, Claudio Lucchese, and Laura Ricci. 2016. Fast connected components computation in large graphs by vertex pruning. *IEEE Transactions on Parallel and Distributed systems* 28, 3 (2016), 760–773.

[23] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature* 518, 7540 (2015), 529–533.

[24] Fan Mo, Hamed Haddadi, Kleomenis Katevas, Eduard Marin, Diego Perino, and Nicolas Kourtellis. 2021. PPFL: Privacy-preserving Federated Learning with Trusted Execution Environments. In *ACM MobiSys*.

[25] Nenad Petrović. 2018. Model-driven Approach for Deployment of Container-based Applications in Fog Computing. In *Proceedings of the Fifth International Conference on Electrical, Electronic and Computing Engineering (IcETRAN 2018)*. 1084–1089.

[26] Rawaa Qasha, Jacek Cala, and Paul Watson. 2015. Towards automated workflow deployment in the cloud using tosca. In *2015 IEEE 8th International Conference on Cloud Computing*. IEEE, 1037–1040.

[27] Anit Kumar Sahu, Tian Li, Maziar Sanjabi, Manzil Zaheer, Ameet Talwalkar, and Virginia Smith. 2018. On the convergence of federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127* 3 (2018).

[28] Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet Talwalkar. 2017. Federated multi-task learning. *arXiv preprint arXiv:1705.10467* (2017).

[29] Luis M Vaquero, Felix Cuadrado, Yehia Elkhatib, Jorge Bernal-Bernabe, Satish N Srirama, and Mohamed Faten Zhani. 2019. Research challenges in nextgen service orchestration. *Future Generation Computer Systems* 90 (2019), 20–38.

[30] John Violos, Vinicius Monteiro de Lira, Patrizio Dazzi, Jörn Altmann, Baseem Al-Athwari, Antonia Schwichtenberg, Young-Woo Jung, Theodora Varvarigou, and Konstantinos Tserpes. 2017. User behavior and application modeling in decentralized edge cloud infrastructures. In *International Conference on the Economics of Grids, Clouds, Systems, and Services*. Springer, Cham, 193–203.

[31] John Violos, Evangelos Psomakelis, Dimitrios Danopoulos, Stylianos Tsanakas, and Theodora Varvarigou. 2020. Using LSTM Neural Networks as Resource Utilization Predictors: The Case of Training Deep Learning Models on the Edge. In *Economics of Grids, Clouds, Systems, and Services (Lecture Notes in Computer Science)*, Karim Djemame, Jörn Altmann, José Ángel Bañares, Orna Agmon Ben-Yehuda, Vlado Stankovski, and Bruno Tuffin (Eds.). Springer International Publishing, Cham, 67–74. https://doi.org/10.1007/978-3-030-63058-4_6

[32] Zhenyu Wen, Renyu Yang, Peter Garraghan, Tao Lin, Jie Xu, and Michael Rovatsos. 2017. Fog orchestration for internet of things services. *IEEE Internet Computing* 21, 2 (2017), 16–24.

[33] Chan-Hyun Youn, Min Chen, and Patrizio Dazzi. 2017. Cloud Broker and Cloudlet for Workflow Scheduling.

[34] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. 2019. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *arXiv preprint arXiv:1911.10635* (2019).