# Efficient Management in Fog Computing

José Santos*, Tim Wauters* and Filip De Turck*
* Ghent University - imec, IDLab, Department of Information Technology
Technologiepark-Zwijnaarde 126, 9052 Gent, Belgium
Email: josepedro.pereiradossantos@UGent.be

***Abstract*— Recent application domains such as the Internet of Things (IoT) and Smart Cities (SCs) have introduced novel challenges to Cloud Computing based on their stringent requirements (e.g., low latency, high bandwidth). With the exponential growth of IoT traffic in the last few years, traditional cloud systems have become inadequate for these applications since requests are made on-demand simultaneously by multiple devices at different locations. The Fog Computing (FC) paradigm has emerged to deal with the limitations of traditional clouds since computational resources are placed at the edges of the network, aiming to decrease the latency expected by IoT devices and reduce the amount of data sent to the cloud. However, research challenges persist in FC since it is not a mature concept yet. This PhD research addresses four challenges in the FC domain focused on providing an efficient resource allocation in these distributed infrastructures. This dissertation includes theoretical formulations as benchmarks for resource allocation, fog-based architectural concepts, anomaly detection practices for IoT, and latency-aware allocation approaches that lead to the implementation of a network-aware framework named *Diktyo*. It optimizes the allocation of container-based service chains by considering latency and bandwidth in the scheduling process of a well-known container orchestration platform, Kubernetes. Experiments showed that *Diktyo* increases throughput by 22% and reduces latency by 45% for microservice benchmark applications.**

***Index Terms*—Fog Computing, Resource Allocation, Service Function Chaining, Orchestration, Microservices, Kubernetes**

## I. Introduction

Over the last few years, Cloud Computing [1] has seen enormous growth since it has become the *defacto* standard for application deployments. These systems reduce expenses for enterprises since computational resources are requested via a cloud provider responsible for maintaining and upgrading the cloud infrastructure. However, recent application domains such as the Internet of Things (IoT) and Smart Cities (SCs) have introduced novel challenges mainly concerning resource allocation [2]. IoT traffic has been increasing exponentially in the last few years (from 6.1 billion in 2018 to 12.5 billion connections in 2022 [3]), making centralized cloud solutions impractical for these applications due to their stringent requirements, including high mobility coverage and low latency. Thus, the Fog Computing (FC) paradigm [4] has emerged as an evolution of Cloud Computing to deal with the stringent requirements of IoT applications by placing computing resources on several locations in the network area, aiming to decrease the expected latency. Fig. 1 illustrates a typical FC architecture, where IoT devices (e.g., mainly sensors and actuators) communicate through wireless gateways via the fog layer, where multiple Fog Nodes (FNs) provide computing resources. Each FN represents a small cloud entity that provides a given set of computational resources. FNs communicate with the cloud layer through Cloud Nodes (CNs), representing the top-level management entities.

FC and Multi-access Edge Computing (MEC) are close concepts [5]. FC focuses on IoT while MEC focuses on the mobile network differing in the considered interactions (i.e., between edges and cloud). MEC aims to deploy services close to end-users to reduce latency and avoid congestion in the network core. MEC follows guidelines established by the European Telecommunications Standards Institute (ETSI) Network Function Virtualization (NFV) Management and Orchestration (MANO) while FC follows architectural principles established by the ETSI Machine-to-Machine (M2M) technical committee [6]. Bi-directional communications between FNs and CNs are important in FC due to the hierarchical architecture. For instance, a service requiring high computational requirements is deployed in the cloud, needing to communicate with another service placed in the fog to reduce data transportation to the cloud. These interactions need to be considered in the allocation process, leading to complex service dependencies that must be guaranteed.

The heterogeneity of Low Power Wide Area Network (LPWAN) technologies introduces a novel set of challenges coming from the wireless domain [7]. LPWANs are typically used by low-powered devices with low bandwidth capacity. The main advantage offered by these technologies is the long communication range, usually of a few kilometers. LPWANs operate at a lower cost with higher energy efficiency than traditional mobile networks. These solutions can support massive numbers of connected devices over a large area, making them suitable for M2M and IoT use cases. However, to decide which LPWAN technology is the most adequate for a given use case is not a trivial task. Selecting an LPWAN depends on the specific application and its requirements, including minimum communication range, minimum data rate, downlink capacity, and additional security layer. Previous works on resource allocation only addressed constraints coming from the cloud domain. Little attention has been given to requirements stemming from the characteristics of wireless networks. When deploying IoT applications, cloud operators need to know which LPWAN technology will be available for IoT devices to access the deployed services. A smart metering use case or a latency-sensitive air quality application pose different challenges to the cloud system and LPWAN technology. The

current LPWAN ecosystem includes a plethora of technologies with diverse characteristics far from mature [7]. The performance and scalability of these technologies are still uncertain since these LPWAN technologies have only been assessed through small-scale experiments and simulations. Furthermore, Service Function Chaining (SFC) placement [8] has been studied in the network management domain during the last few years. An application is decomposed into several services connected in a specific order forming a service chain. SFC allows mobile operators to benefit from the high flexibility and low operational costs introduced by network softwarization, offering a reliable alternative to today's static network environment. SFC has been studied for Software-Defined Networking (SDN) and NFV use cases, mainly for MEC deployments [9]. However, SFC is still unexplored in container placement and in the IoT domain. Most efforts focus on virtual network embedding and Virtual Machine (VM) placement [10].

This dissertation addresses four challenges in FC concerning resource allocation detailed in Sec. II. The first challenge relates to studying theoretical formulations based on Integer Linear Programming (ILP) for a benchmark in resource allocation research in FC considering both cloud and wireless characteristics. Then, a fog-based system for SC applications has been designed, followed by a distributed data monitoring and analysis approach aiming to provide efficient anomaly detection in FC environments. Finally, a network-aware framework named *Diktyo* has been implemented for the inclusion of latency and bandwidth in the scheduling process of a popular container orchestration platform named Kubernetes (K8s) to address container-based SFC. This PhD research[1] [11] resulted in several publications (15 as a first author) and it has been conducted in partnership with leading industry partners such as IBM and within Antwerp's City of Things (CoT) project [12]. The main contributions of this dissertation are the following:

- **Publications in International Journals**: Five journal articles [13]–[17] have been published and one [18] is currently under review.
- **Publications in International Conferences**: Nine conference papers [19]–[27] resulted from the PhD research.
- **Publications in Book Chapters**: One book chapter [28] has been published.
- **Code Repositories**: During the dissertation, we advocated for open and reproducible research, and several prototypes have been made publicly available[2].

The remainder of this paper is structured as follows. Sec. II details the four challenges addressed during the PhD research. Sec. III details the proposed theoretical formulations for the IoT service placement, while Sec. IV presents the FC-based architecture for SC applications. Sec. V addresses anomaly detection in an FC environment, followed by a network-aware scheduling approach for microservice-based applications in Sec. VI. Lastly, Sec. VII concludes the paper and highlights future research directions that can leverage the PhD work presented in this paper.

[1]https://imec-publications.be/handle/20.500.12860/39650

[2]https://github.com/jpedro1992/

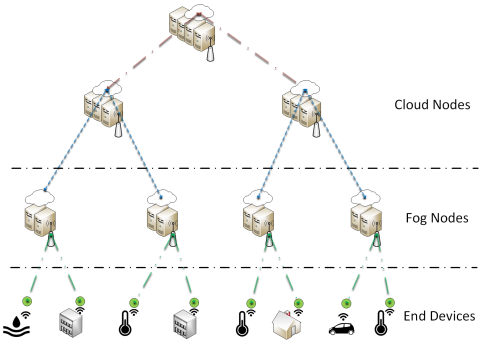



Fig. 1: Overview of a Fog Computing (FC) architecture [11].

## II. Research Challenges

***Challenge #1****: provide a benchmark for resource allocation research in FC.* The distribution of computing resources in FC adds further complexity to the service placement since resources are available across the network area, and service requests can come from multiple locations. Deploying applications far from devices results in high communication latency since devices and gateways lack processing power, storage capacity, and memory [29]. The heterogeneity of the hardware resources also adds further complexity to the problem since an FN has lower computing capacities than a CN. Thus, efficient service placement strategies need to consider the hosts' computing capacity, application requirements, and hosts' locations. Different factors should be analyzed, including latency, energy efficiency, and network bandwidth. In recent years, several works have addressed resource allocation in FC (e.g., [30], [31]), but none of them addressed the real-time requirements of IoT applications. Also, the implications of LPWANs in the allocation process should be studied to establish a relationship between the cloud and the wireless domain.

***Challenge #2****: design an FC framework supporting autonomous management and orchestration functionalities.* In recent years, the ETSI oneM2M has been working towards an end-to-end (E2E) high-level architecture for M2M communication. The aim is to establish for M2M what the 3rd Generation Partnership Project (3GPP) realized for mobile networks. Several aspects are currently being addressed, including security, data management, device authentication, and M2M service subscription. Designing an FC framework addressing all these aspects would help to standardize FC architectures. Several works have proposed IoT or FC architectures [32], [33] with several functionalities. However, without a clear path to standardization, it is hard to integrate and combine the efforts of academia and industry.

***Challenge #3****: implement efficient distributed data monitoring and analysis in FC.* Monitoring IoT applications is especially important for those focused on personal health monitoring or emergency response services since delays can impact their performance and produce severe consequences.

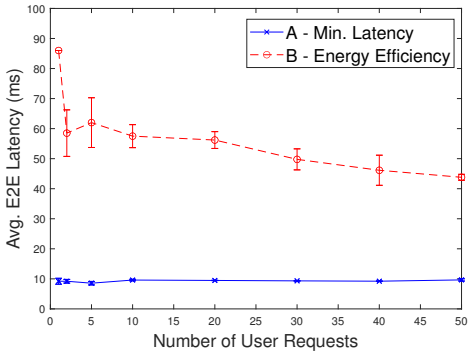



Fig. 2: The expected E2E latency for both objectives [16].

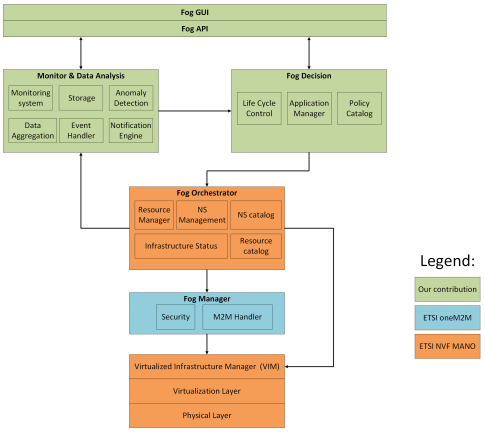



Fig. 3: Detailed overview of the FN architecture [13].

Detecting unusual events or malfunctions beforehand is thus an important matter. Traditional anomaly detection approaches are inadequate for latency-sensitive IoT applications since these solutions would require sending all data to a centralized location, resulting in high latency. Also, data is not transported fast enough due to the low bandwidth capacity of IoT devices. FC helps to reduce the latency since services can be deployed close to these devices. By identifying unusual events at the fog layer, malfunctions are quickly detected in IoT devices, increasing the overall Quality of Service (QoS) of IoT applications [34]. Thus, an anomaly detection approach for IoT should be based on the advantages of FC.

***Challenge #4****: consider latency and bandwidth in the scheduling process in a container orchestration system.* Containers have revolutionized application deployment and lifecycle management in current cloud platforms [35]. Applications have evolved from single monoliths to complex graphs of loosely-coupled microservices. However, the efficient allocation of microservice-based applications is challenging due to their complex inter-dependencies. Recent applications such as IoT and video streaming services are becoming even more delay-sensitive, demanding lower latency between dependent microservices. Scheduling policies in popular container orchestration platforms mainly aim to increase the resource efficiency of the infrastructure insufficient for latency-sensitive applications. FC provides computing resources at the edge and fog, helping to meet stringent latency and bandwidth requirements. Thus, efficient allocation strategies need to consider bandwidth and latency in the scheduling process. Previous works [36], [37] focus mainly on VM allocation and migration, and little attention has been given to containerized applications or container-based service chaining concepts. The main challenge is to design efficient scheduling strategies that assess latency and available bandwidth while deploying container-based service chains in FC infrastructures.

## III. Integer Linear Programming for the IoT Service Placement

Through ILP, the trade-offs between different allocation strategies have been assessed, such as minimizing latency and maximizing energy efficiency. Latency is related to the network latency in the communication between cluster nodes

TABLE I: Performance evaluation of the infrastructure [13].

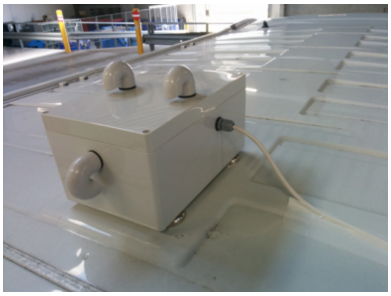
| Data Samples | Network Bandwidth | Transport Time | Alert Delay |
|---|---|---|---|
| 70000 | 31.36 Mbps (100%) | 6.27 s | 0.46 ms |
| | 313.6 Kbps (1% of samples) | 62.72 ms | 0.33 ms |

running services belonging to an IoT application and the end devices sending requests. Energy efficiency is related to the number of cluster nodes used for the service deployment. The model [19] considers several constraints, including hardware capacities, network bandwidth, and path loss. The model addresses cloud requirements and characteristics stemming from the wireless domain, which have not yet been explored in-depth in literature. The cloud model is based on previous work proposed by Moens et al. [36] for network-aware placement of service-oriented applications in clouds. The ILP formulation can serve as a benchmark for resource allocation research in FC since the model has been validated for IoT applications in fog-cloud environments without loss of generality. This work has been extended in [16] by presenting a Mixed Integer Linear Programming (MILP) model for IoT service placement that considers SFC, different LPWAN technologies, service replication, and multiple optimization objectives. The model provides several insights into the complete E2E resource provisioning in FC environments. Results have shown clear trade-offs between the evaluated allocation strategies (Fig. 2). The main contribution of this work is the ILP formulation that can serve as a benchmark in future placement research in FC.

## IV. A Fog-based Architecture for Smart Cities

A novel architectural paradigm for SC applications has been studied in [13]. The approach follows the guidelines of the ETSI NFV MANO architecture extending it with additional software components. Also, a novel Peer-to-Peer (P2P) fog protocol has been designed to enable the exchange of application service information between FNs for fast provisioning decisions. Each FN decides where and when it is more suitable to deploy and instantiate each microservice instance. Fig. 3 presents the detailed architecture of the

(a) Inside view of the multi-radio sensor.



(b) Air Quality sensor mounted on a Bpost car.

Fig. 4: As part of the Antwerp's CoT project, multi-radio sensors have been mounted on Belgian postal service cars.

FN management system. The FN must set up and manage its infrastructure and associated devices in an autonomous manner. Each FN manages a set of computational resources by using a virtualization layer residing over a physical layer, offering virtualization of the main network functionalities. The Virtualized Infrastructure Manager (VIM) performs the life-cycle management of the deployed network functions. The Fog Manager (FM) is responsible for managing the attached IoT devices through a Fog Agent (FA). The FM addresses the device management and M2M security guidelines defined by ETSI oneM2M. The FM module is mainly responsible for device discovery operations, keeping track of the devices' mobility, and ensuring M2M secured communications. Each FN has its own instance of a Fog Orchestrator (FO) component. The FO module is mainly responsible for the life-cycle management of microservices, interface with the monitoring and data analysis system, and interface with the Fog Decision (FD) module. The FD component hosts the main intelligence components responsible for applying the network behavior desired by network administrators, and providing autonomous responses to unknown situations detected by the monitoring and data analysis module. The performance of the FC infrastructure has been evaluated and compared with a traditional centralized cloud, achieving lower delays and lower network bandwidth usage (Table I). The main contribution of this work is the design of a fully integrated FN management system alongside the foreseen application layer P2P fog protocol for the exchange of application service provisioning information between FNs.

## V. A Fog-based anomaly detection approach

In [13], [20], a Fog-based distributed anomaly detection approach has been designed for Antwerp's CoT testbed [12] focused on low-power FC deployments and evaluated based on an air quality monitoring application. The aim is to detect high amounts of organic compounds in the atmosphere and then alert citizens of air pollution in near real-time. As an initial proof of concept (Fig. 4), air quality sensors have been mounted on the roofs of the Belgian postal services delivery cars (Bpost) to collect measurements of typical gases and

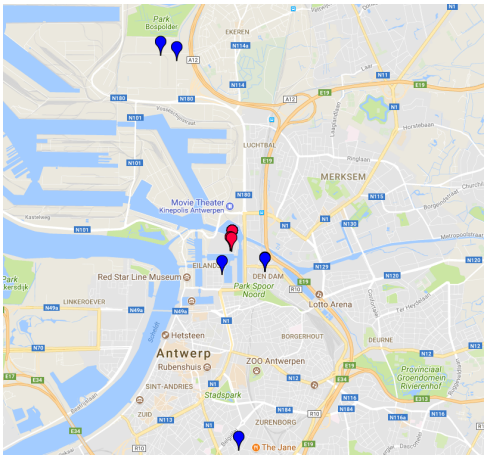



Fig. 5: GPS locations (Bpost car 1 - red / Bpost car 2 - blue) considered as outliers in the evaluation.

TABLE II: Comparison between different LPWANs [20].

| LPWAN Technology | Comm. Range | Upload Data Rate | Upload Time (Packet of 53 bytes) |
|---|---|---|---|
| LoRaWAN | 5 km | 50 kbps | 25.15 ms |
| Sigfox | 10 km | 300 bps | 1.45 s |
| LTE-M | 5 km | 1 Mbps | 17.09 ms |
| DASH7 | 5km | 166.57 kbps | 19.22 ms |
| IEEE 802.11ah | 1 km | 346.66 Mbps | 3.33 ms |

climate data, such as temperature and humidity, which are then annotated with GPS locations (Fig. 5). These sensors allow the gathering of real-time air quality information with broad city coverage since each car is continuously driving around in the city. Popular LPWANs have been studied based on multiple criteria (e.g., communication range, available bandwidth) for the presented anomaly detection approach. The most appropriate ones have been selected based on the evaluated air quality monitoring case. A suitable set of LPWANs, including IEEE 802.11ah, DASH7, and LTE-M, can be applied as wireless communication enablers for the evaluated scenario (Table II). Results have shown that clustering and outlier detection mechanisms can be performed by fog resources close to IoT sensors and, thus, send timely alerts in case unusual events are detected. Also, by distributing anomaly detection algorithms in a fog-cloud infrastructure, the network bandwidth usage and latency are significantly reduced compared to a centralized cloud infrastructure. The main contributions of this work are the distributed anomaly detection approach, including its design and implementation, and the LPWAN evaluation.

## VI. Network-Aware Scheduling for Container-Based Applications

In [15], a FC architecture based on the K8s architectural model has been proposed alongside latency-aware deployment strategies for IoT applications focused on container-based service chaining in [22], [23]. Results have shown that these latency-aware approaches could significantly reduce the

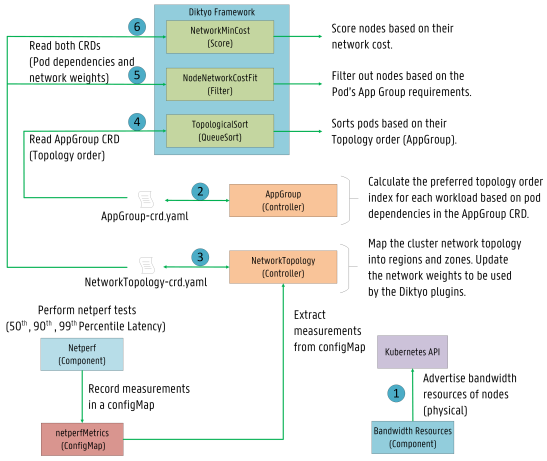



Fig. 6: Illustration of the *Diktyo* framework [18].

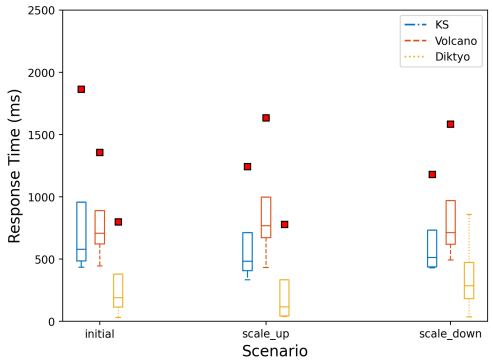



Fig. 7: Diktyo reduces the response time on average by 45% for *POST /setCurrency* requests [18].

network latency compared to the default K8s scheduling mechanism. These works have been the foundation for the network-aware framework named *Diktyo* proposed in [18]. *Diktyo* proposes additional scheduling plugins aiming to determine low-latency deployment schemes for applications scheduled on K8s clusters by considering the application's microservice dependencies and the underlying infrastructure topology. The framework minimizes the application's E2E latency by selecting nodes with low network costs for dependent microservices. Also, it filters out nodes without enough bandwidth to run the placed microservices based on previous deployments. Fig. 6 shows an overview of the *Diktyo* framework, including its main components. Bandwidth resources are advertised to the K8s API to consider the node available bandwidth in the scheduling process ①. Then, the framework introduces two Custom Resources (CRs) [38]: **AppGroup** and **NetworkTopology** to consider both the application dependency information ② and the infrastructure network topology ③ when scheduling pods in K8s. The **NetworkTopology** controller updates network weights between cluster nodes across regions and zones based on a netperf component. *Diktyo* provides network-aware algorithms implemented as three scheduling plugins based on the K8s scheduling framework [39]:

- **TopologicalSort**: pods sorted based on their established dependencies ④.
- **NodeNetworkCostFit**: nodes filtered out based on the pod's AppGroup requirements ⑤
- **NetworkMinCost**: nodes are scored based on network costs ensuring a low cost between dependent pods ⑥.

Experiments in a K8s cluster consisted of typical and often used microservice benchmark applications requiring high bandwidth (Redis Cluster [40]) or low latency (Online Boutique [41]). Results have shown that *Diktyo* increases the throughput by 22% for Redis Cluster and reduces the latency by up to 45% for Online Boutique by deploying dependent microservices close to each other (Fig. 7). The main contribution of this work is the design and development of the *Diktyo* framework for the K8s platform and its inclusion in the K8s scheduling plugins project [39]. This work represents a collaboration with IBM TJ Watson (NYC, USA). To the best of our knowledge, *Diktyo* goes beyond the current state-of-the-art since it is an important step towards scalable network-aware placement of dependent microservices in future cloud-native architectures.

## VII. Conclusions and Future Directions

This dissertation presented several methods for efficient resource allocation in an FC environment. Theoretical formulations, architectural paradigms, and practical implementations are among the various proposed methods for efficient IoT service placement in FC. Nevertheless, novel challenges are emerging, which will need to be addressed by the network management community. The advent of recent architectural paradigms as FC enabled the deployment of service chains on computational resources from the cloud up to the edge, creating a continuum of virtual resources. Machine learning and artificial intelligence will continue to evolve and position themselves as crucial enablers of autonomous networks, which will strongly impact the performance of emerging use cases, such as immersive video streaming and autonomous vehicles. Extended reality applications will require throughput above 1 Tbps, and their interactive experiences will need sub-millisecond latency. Autonomous cars will demand high mobility and at least seven levels of reliability without necessarily requiring higher throughput. In addition, distributed and decentralized allocation strategies will become even more important in the next few years to efficiently manage service deployments across several clusters and domains.

In conclusion, this PhD research focused on efficient orchestration practices for FC infrastructures. A collaboration with IBM TJ Watson has been established to bring network awareness to K8s scheduling, the most popular cloud-native platform. This dissertation resulted in about 16 scientific publications and attracted a lot of attention from the research community, resulting in 566 citations up to date reported by Google Scholar.

## REFERENCES


[1] A. Sunyaev, "Cloud computing," in *Internet computing*. Springer, 2020, pp. 195–236.

[2] H. Arasteh, V. Hosseinnezhad, V. Loia, A. Tommasetti, O. Troisi, M. Shafie-khah, and P. Siano, "Iot-based smart cities: A survey," in *2016 IEEE 16th international conference on environment and electrical engineering (EEEIC)*. IEEE, 2016, pp. 1–6.

[3] Cisco, "Annual internet report (2018–2023)," accessed on 22 September 2022. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.pdf

[4] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 workshop on mobile big data*, 2015, pp. 37–42.

[5] P. Bellavista, J. Berrocal, A. Corradi, S. K. Das, L. Foschini, and A. Zanni, "A survey on fog computing for the internet of things," *Pervasive and mobile computing*, vol. 52, pp. 71–99, 2019.

[6] M. S. De Brito, S. Hoque, T. Magedanz, R. Steinke, A. Willner, D. Nehls, O. Keils, and F. Schreiner, "A service orchestration architecture for fog-enabled infrastructures," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*. IEEE, 2017, pp. 127–132.

[7] K. Mekki, E. Bajic, F. Chaxel, and F. Meyer, "A comparative study of lpwan technologies for large-scale iot deployment," *ICT express*, vol. 5, no. 1, pp. 1–7, 2019.

[8] D. Bhamare, R. Jain, M. Samaka, and A. Erbad, "A survey on service function chaining," *Journal of Network and Computer Applications*, vol. 75, pp. 138–155, 2016.

[9] Y. Nam, S. Song, and J.-M. Chung, "Clustered nfv service chaining optimization in mobile edge clouds," *IEEE Communications Letters*, vol. 21, no. 2, pp. 350–353, 2016.

[10] H. Hawilo, M. Jammal, and A. Shami, "Network function virtualization-aware orchestrator for service function chaining placement in the cloud," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 3, pp. 643–655, 2019.

[11] J. Santos, "Efficient resource allocation in a fog computing environment," Ph.D. dissertation, Ghent University, 2022, accessed on 22 September 2022. [Online]. Available: https://imec-publications.be/handle/20.500.12860/39650

[12] S. Latre, P. Leroux, T. Coenen, B. Braem, P. Ballon, and P. Demeester, "City of things: An integrated and multi-technology testbed for iot smart city experiments," in *2016 IEEE international smart cities conference (ISC2)*. IEEE, 2016, pp. 1–8.

[13] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Fog computing: Enabling the management and orchestration of smart city applications in 5g networks," *Entropy*, vol. 20, no. 1, p. 4, 2017.

[14] J. Santos, T. Vanhove, M. Sebrechts, T. Dupont, W. Kerckhove, B. Braem, G. Van Seghbroeck, T. Wauters, P. Leroux, S. Latre *et al.*, "City of things: Enabling resource provisioning in smart cities," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 177–183, 2018.

[15] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Resource provisioning in fog computing: From theory to practice," *Sensors*, vol. 19, no. 10, p. 2238, 2019.

[16] ——, "Towards end-to-end resource provisioning in fog computing over low power wide area networks," *Journal of Network and Computer Applications*, vol. 175, p. 102915, 2021.

[17] ——, "Towards low-latency service delivery in a continuum of virtual resources: State-of-the-art and research directions," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 4, pp. 2557–2589, 2021.

[18] J. Santos, C. Wang, T. Wauters, and F. De Turck, "Diktyo: Network-aware scheduling in container-based clouds," *Submitted to IEEE Transactions on Network and Service Management*, September 2022.

[19] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Resource provisioning for iot application services in smart cities," in *2017 13th International Conference on Network and Service Management (CNSM)*. IEEE, 2017, pp. 1–9.

[20] J. Santos, P. Leroux, T. Wauters, B. Volckaert, and F. De Turck, "Anomaly detection for smart city applications over 5g low power wide area networks," in *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2018, pp. 1–9.

[21] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Towards dynamic fog resource provisioning for smart city applications," in *2018 14th International Conference on Network and Service Management (CNSM)*. IEEE, 2018, pp. 290–294.

[22] ——, "Towards network-aware resource provisioning in kubernetes for fog computing applications," in *2019 IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2019, pp. 351–359.

[23] ——, "Towards delay-aware container-based service function chaining in fog computing," in *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020, pp. 1–9.

[24] L. N. Vijouyeh, M. Sabaei, J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Efficient application deployment in fog-enabled infrastructures," in *2020 16th International Conference on Network and Service Management (CNSM)*. IEEE, 2020, pp. 1–9.

[25] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Live demonstration of service function chaining allocation in fog computing," in *2020 6th IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2020, pp. 362–364.

[26] J. Santos, J. van der Hooft, M. T. Vega, T. Wauters, B. Volckaert, and F. De Turck, "Srfog: A flexible architecture for virtual reality content delivery through fog computing and segment routing," in *2021 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. IEEE, 2021, pp. 1038–1043.

[27] J. Santos, T. Wauters, B. Volckaert, and F. De Turck, "Resource provisioning in fog computing through deep reinforcement learning," in *IM2021, the IFIP/IEEE Symposium on Integrated Network and Service Management*, 2021, pp. 1–7.

[28] ——, "Reinforcement learning for service function chain allocation in fog computing," *Communication Networks and Service Management in the Era of Artificial Intelligence and Machine Learning*, pp. 147–173, 2021.

[29] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE access*, vol. 5, pp. 1872–1899, 2017.

[30] K. Velasquez, D. P. Abreu, M. Curado, and E. Monteiro, "Service placement for latency reduction in the internet of things," *Annals of Telecommunications*, pp. 1–11, 2016.

[31] C. Tang, M. Steinder, M. Spreitzer, and G. Pacifici, "A scalable application placement controller for enterprise data centers," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 331–340.

[32] L. Sánchez, V. Gutiérrez, J. A. Galache, P. Sotres, J. R. Santana, J. Casanueva, and L. Muñoz, "Smartsantander: Experimentation and service provision in the smart city," in *Wireless Personal Multimedia Communications (WPMC), 2013 16th International Symposium on*. IEEE, 2013, pp. 1–6.

[33] D. P. Abreu, K. Velasquez, M. Curado, and E. Monteiro, "A resilient internet of things architecture for smart cities," *Annals of Telecommunications*, pp. 1–12, 2016.

[34] L. Lyu, J. Jin, S. Rajasegarar, X. He, and M. Palaniswami, "Fog-empowered anomaly detection in iot using hyperellipsoidal clustering," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1174–1184, 2017.

[35] N. Dragoni, S. Giallorenzo, A. L. Lafuente, M. Mazzara, F. Montesi, R. Mustafin, and L. Safina, "Microservices: yesterday, today, and tomorrow," *Present and ulterior software engineering*, pp. 195–216, 2017.

[36] H. Moens, B. Hanssens, B. Dhoedt, and F. De Turck, "Hierarchical network-aware placement of service oriented applications in clouds," in *Network Operations and Management Symposium (NOMS)*. IEEE, 2014, pp. 1–8.

[37] S. Ahvar, H. P. Phyu, S. M. Buddhacharya, E. Ahvar, N. Crespi, and R. Glitho, "Ccvp: Cost-efficient centrality-based vnf placement and chaining algorithm for network service provisioning," in *2017 IEEE Conference on Network Softwarization (NetSoft)*. IEEE, 2017, pp. 1–9.

[38] Kubernetes, "Custom resources," accessed on 28 March 2022. [Online]. Available: https://kubernetes.io/docs/concepts/extend-kubernetes/api-extension/custom-resources/.

[39] Kubernetes Scheduler Plugins, "Repository for out-of-tree scheduler plugins based on the scheduler framework." accessed on 28 March 2022. [Online]. Available: https://github.com/kubernetes-sigs/scheduler-plugins.

[40] Redis, "Redis, an open source in-memory data structure store." accessed on 22 September 2021. [Online]. Available: https://redis.io/

[41] Online Boutique, "Online boutique, a cloud-native microservices demo application." accessed on 22 September 2021. [Online]. Available: https://github.com/GoogleCloudPlatform/microservices-demo