Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 7476201, 23 pages
https://doi.org/10.1155/2018/7476201

Review Article

A Taxonomy for Management and Optimization of Multiple Resources in Edge Computing

Klervie Toczé and Simin Nadjm-Tehrani

Department of Computer and Information Science, Linköping University, Linköping, Sweden

Correspondence should be addressed to Klervie Toczé; klervie.tocze@liu.se

Received 16 November 2017; Revised 11 March 2018; Accepted 17 April 2018; Published 4 June 2018

Academic Editor: Anna Kobusinska

Edge computing is promoted to meet increasing performance needs of data-driven services using computational and storage resources close to the end devices at the edge of the current network. To achieve higher performance in this new paradigm, one has to consider how to combine the efficiency of resource usage at all three layers of architecture: end devices, edge devices, and the cloud. While cloud capacity is elastically extendable, end devices and edge devices are to various degrees resource-constrained. Hence, an efficient resource management is essential to make edge computing a reality. In this work, we first present terminology and architectures to characterize current works within the field of edge computing. Then, we review a wide range of recent articles and categorize relevant aspects in terms of 4 perspectives: resource type, resource management objective, resource location, and resource use. This taxonomy and the ensuing analysis are used to identify some gaps in the existing research. Among several research gaps, we found that research is less prevalent on data, storage, and energy as a resource and less extensive towards the estimation, discovery, and sharing objectives. As for resource types, the most well-studied resources are computation and communication resources. Our analysis shows that resource management at the edge requires a deeper understanding of how methods applied at different levels and geared towards different resource types interact. Specifically, the impact of mobility and collaboration schemes requiring incentives are expected to be different in edge architectures compared to the classic cloud solutions. Finally, we find that fewer works are dedicated to the study of nonfunctional properties or to quantifying the footprint of resource management techniques, including edge-specific means of migrating data and services.

1. Introduction

Recently, the edge computing paradigm, which consists in having network nodes with computational and storage resources close to the devices (mobile phones, sensors) at the edge of the current network, has attracted interest from both industry and researchers, carrying the promise of a new communication era in which industry can meet the rising performance needs of future applications.

Indeed, with a forecast of 9 billion mobile subscriptions in the world by 2022, of which 90% will include mobile broadband, coupled to an eightfold increase in mobile traffic and 17.6 billion of Internet of Things (IoT) devices also sending data [1], there will be a considerable strain put on the network. The current network technologies need to undergo a paradigm shift in order to handle this situation [2]. Therefore, the aim is to avoid overwhelming the network up to the cloud and, when possible, move some computing and data analysis closer to the users to enable better scalability [3]. Thus, the main idea of edge (or fog) computing is to have intermediate computing facilities between the end devices and the current cloud. As suggested by Mehta et al. [4], this would also enable the current telecom network operators to reduce their operational costs.

In addition to this, moving computing and storage to the edge of the network has other benefits [3] such as reducing the latency and jitter [5], which is especially important for real-time applications such as self-driving cars. Moreover, it enables more privacy for the users by making it possible to keep private data at the edge and enforce privacy policies for the data sent to the cloud (such as blurring sensitive info on a video [2]). Finally, edge networking makes the applications

more resilient by being able to process requests at the edge even if the central cloud is down.

In order to achieve this and to make edge computing a reality and a success, there is a need for an efficient resource management at the edge. Indeed, mobile devices or IoT devices are resource-constrained devices, whereas the cloud has almost unlimited but far away resources. Providing and/or managing the resources at the edge will enable the end device to spare resources (e.g., stored energy in batteries) and speed up computation and allows using resources it does not possess. Moreover, keeping data close to where it was generated enables better control, especially for privacy-related issues. Finally, being located close to the user, edge computing makes it possible to increase the quality of provided services through the use of profiling within a local context, without compromising the privacy or having to handle a large number of users. This is known as *context adaptation*.

Even though this is still an emerging research area, there is a lot of work ongoing under different denominations including mobile cloud computing [6], fog computing [7], edge computing [3], mobile edge computing [8], path computing [9], mobile edge cloud [10], mobile edge network [4], infinite cloud [11], follow-me cloud [12], mobile follow-me cloud [13], multitier cloud federations [14], small cell cloud [15], fast moving personal cloud [16], CONCERT [17], distributed clouds [18], and femtoclouds [19, 20].

Independently of the terminology chosen, which might follow the current naming trend, a common concept here is an intermediate level between the device and the traditional cloud. It is possible to find in the literature numerous surveys about those paradigms in general [6, 10, 21–25], specific aspects of them such as security [8, 26], or specific techniques such as Software-Defined Networking (SDN) [27]. However, those typically do not consider the resource aspect. The existing surveys about resources either consider it at a high level [28] or consider only resource/service provisioning metrics [29].

One area that is of high importance and is present in many use-cases in edge computing is *offloading*. This is the idea of executing a task on a device other than the current execution target. This other device has typically more powerful computational capacities or fewer energy constraints. Resource management is tightly connected to offloading since in order to take a decision to offload, one needs to have knowledge about system resources. This knowledge is provided by resource management techniques. For example, resource discovery can be used as an input for taking an offloading decision, while resource allocation techniques can be used to perform the offloading decision. To the best of our knowledge, existing surveys about resource management for offloading at the edge focus on an end device perspective [30, 31], on the resource allocation part of resource management [32, 33], or on a single-user/multiuser perspective [34].

We aim to complement those surveys by providing a more comprehensive perspective. That is, (a) we consider allocation as one among five resource management objectives, (b) we consider edge resources in addition to end device or cloud resources, (c) we address multiple types of resources and interrelations amongst them, and (d) we review aspects related to locality and what the resource is intended for.

In selecting the survey papers, works considering direct interactions from a device to a cloud [35] or focusing on cloud performance by offloading to the edge [36] are not considered. However, offloading between edge devices or from the edge to the cloud when edge resources are also considered is included. All included papers consider the notion of edge which we attempt to characterize by defining edge-specific architectural instances. This will be done independently of the terminology the authors chose to use. This paper is a substantial extension of our previous much shorter review [37].

In the remaining parts of this paper, we will first present the terminology used, define edge-specific architectures, and present the proposed taxonomy in Section 2. The taxonomy is then exemplified by an extensive review of papers, which are categorized using the taxonomy elements introduced, namely, resource type (Section 3), resource management objective (Section 4), resource location (Section 5), and resource use (Section 6). We then discuss research challenges in Section 7 and conclude the paper in Section 8.

2. Architectures and Research Taxonomy

Edge computing is an innovative area bringing together diverse business sectors such as telecommunication actors, vehicle vendors, cloud providers, and emerging application or device providers, for example, for augmented reality. Therefore, the terminology used in research works is diverse and is still evolving and multiple architectures are considered.

In this section, we present first the relevant terminology associated with edge computing which will be used in the rest of the paper. Then, we discuss the current architectures used and present an architectural breakdown that will be the basis for classifying existing research. Finally, we present our proposed research taxonomy and use it to classify the surveyed works.

2.1. Terminology. Following the development of the IoT, it is nowadays not only computers or smartphones which can be connected to the network but also a large variety of things such as cars, sensors, drones, robots, or home appliances. In this survey, all those objects located at the user end of the network which produce data or need cloud/edge resources will be called *end devices*.

Devices installed at the edge specifically for edge computing purposes are called *edge devices*. We also include under this term the devices that are already now connecting the end devices to the rest of the network, for example, home routers, gateways, access points, or base stations, which are becoming increasingly powerful [38].

Finally, physical components of the cloud are referred to by the term *cloud devices*.

We use those network device classifications to create different *levels* in the network: the device level, the edge level, and the cloud level. Resources that are managed are used to perform *tasks* at some level of the architecture. These can be composed to provide a *service* to the user.

2.2. Current Status of Edge Architectures. There is currently no standard architecture for edge computing, although industry and research initiatives exist, such as the Open Edge Computing (http://openedgecomputing.org/) community, the Open Fog Consortium (https://www.openfogconsortium.org/), and a European Telecommunications Standards Institute (ETSI) standardization group working on Multiaccess Edge Computing (http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing). Current standardization efforts coming from the ETSI group have been reviewed in detail by Mao et al. [34] and Mach and Becvar [33]. Mao et al. [34] also present edge standardization efforts within the 5G standard.

Therefore, current research on edge computing is using several different architectures and there is ongoing work for defining edge computing architectures. Recent surveys focus on presenting these architectures. For example, Liu et al. [10] review different architectures for mobile edge cloud servers and networks, and Mach and Becvar [33] present an overview of proposed solutions enabling computation to be brought close to the end device within the field of mobile edge computing. The approach chosen by Mouradian et al. [39] is to classify the architectures depending on whether they are application-specific or not. They also elaborate on architectural challenges according to 6 criteria including scalability and heterogeneity. Our classification of the device types above is consistent with all the surveys on architecture so far.

2.3. Used Breakdown of Architectures. In this survey, we choose to classify the different architectures into three main categories inspired by the work of Mtibaa et al. [40] and presented in Figure 1. Those categories are technology-independent and aim at visualizing three high-level variants of the edge computing concept that the current works are using.

The first category, named *edge server* and depicted in Figure 1(a), is a generic architecture, where devices are connected to an edge server, which itself is connected to the rest of the network, including the cloud. In this type of architecture, the edge server is at a fixed physical location and has relatively high computational power, though it remains less powerful than a conventional data center used in the cloud computing paradigm. Moreover, there is a clear separation between the device level and the edge level. In the literature, such edge servers are named, for example, cloudlets [41, 42], micro data centers [43, 44], nano data centers [45], or local cloud [46]. They can be located, for example, in shops and enterprises or colocated with the base stations of the telecom access network. Indeed, in the ongoing work on what the fifth generation (5G) of telecommunication networks will look like, a cloud radio access network (C-RAN) is envisaged [47, 48], with connections to other edge computing areas such as mobile cloud computing [49].

The second category, named *coordinator device* and depicted in Figure 1(b), is an architecture, where one end device acts as a coordinator between the other end devices. It also acts as a proxy towards an edge device and/or the cloud if such connectivity is needed. The difference between a coordinator device and an edge server is that the coordinator device can be mobile and has less computational power and bandwidth than an edge server. In this architecture category, the border between the device level and the edge level is not a sharp one, as the coordinator level providing edge functionality is actually an end device. Solutions using this category of architecture are named, for example, fog colonies with a control node [50], vehicular clouds with a cluster head [51], and local clouds with a local resource coordinator [52]. It is interesting to note here that the term *local cloud*, which was already used for describing a part of the edge server architecture category described in the previous paragraph, is used to describe various architectural solutions, illustrating well the fact that the terminology used in edge computing is not yet set.

The last category, named *device cloud* and depicted in Figure 1(c), is an architecture, where the end devices communicate with each other to find needed resources and deliver the wanted services. The devices might communicate with an edge device connected to the cloud if needed but this is not necessary. In this architecture category, the device level and the edge level are thus merged. Research works considering this category of architecture call it opportunistic computing [53], cooperation-based mobile cloud computing [54, 55], or transient clouds [56].

While all these architectures need to be populated with dedicated resource management elements, there is no general agreement about where to place the needed policies. A recent proposal for a generic software architecture that encompasses the edge server version in Figure 1(a) is an enabler for evaluation of multiple resource management policies within common testbeds [57].

2.4. Taxonomy of Edge Resource Management. In addition to classifying the reviewed papers according to the architecture category they consider, we also present a taxonomy of resource management at the edge. This taxonomy, illustrated in Figure 2, aims at getting an overview of state-of-the-art research in this area and presents four main aspects: resource type, objective of resource management, resource location, and resource use.

The two first aspects were constructed by reviewing the current *type* of resources used and the *objective* for which they are used in the literature. The two last aspects are based on mutually exclusive pairs for describing the resource *location* and the *use* of the resource.

In the coming sections, we will describe the different parts of the taxonomy and how the surveyed works can be placed in the four above contexts, as well as the architectural models described.

3. Resource Type

The first step in evaluating the benefit of an edge solution is to decide what are the resource types that can be managed in a better way compared to a centralized system.

An obvious justification for using edge architectures is reducing the response time, which can be done if computation and communication resources are provided and

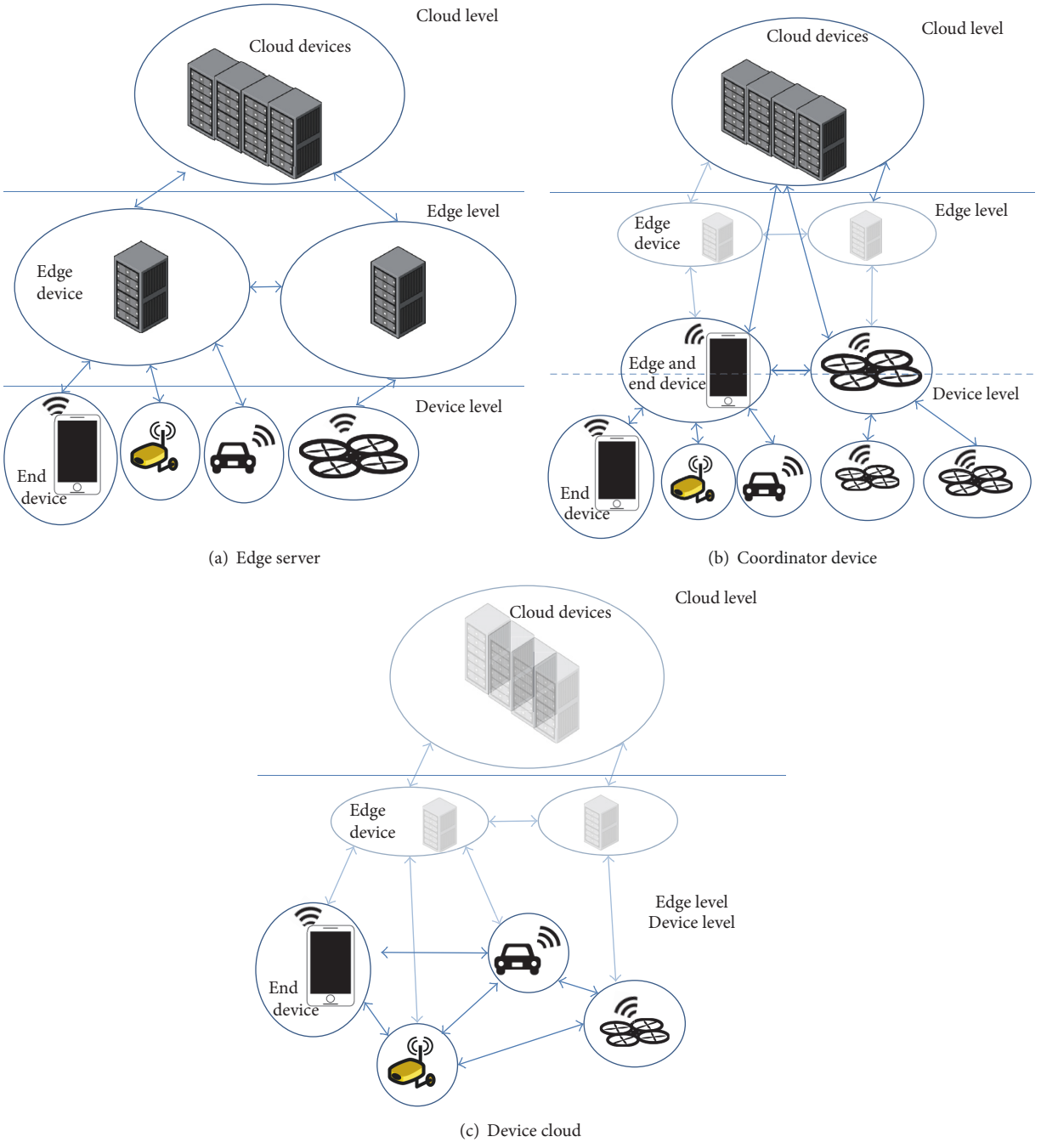

Figure 1: Categories of architectures used in edge computing.

used adequately. Storage as a resource is also a concern, since local storage may benefit security or timeliness due to customized fetching and secure storing mechanisms. A less obvious type of resource is having access to a special type of data (e.g., availability of sensors) that provides local benefits in an application. Examples are the use of cameras or location sensors. The amount and type of data captured in turn affect computation and communication resources (how often to shuffle data and how much to process or filter before shuffling) and implicitly the choice of where and how much of other resources to deploy. The fifth category we consider is energy as a resource, which is clearly influenced by the amount of computation, communication, storage, and data capturing that goes on. Finally, some works consider resources in a generic way using abstract terms such as "Virtual Resource Value" or just as unitless elements in a model.

Table 1 summarizes the surveyed papers in terms of their mapping to the architectural choices in Figure 1. It also shows which resource is focused on within each work, either specific

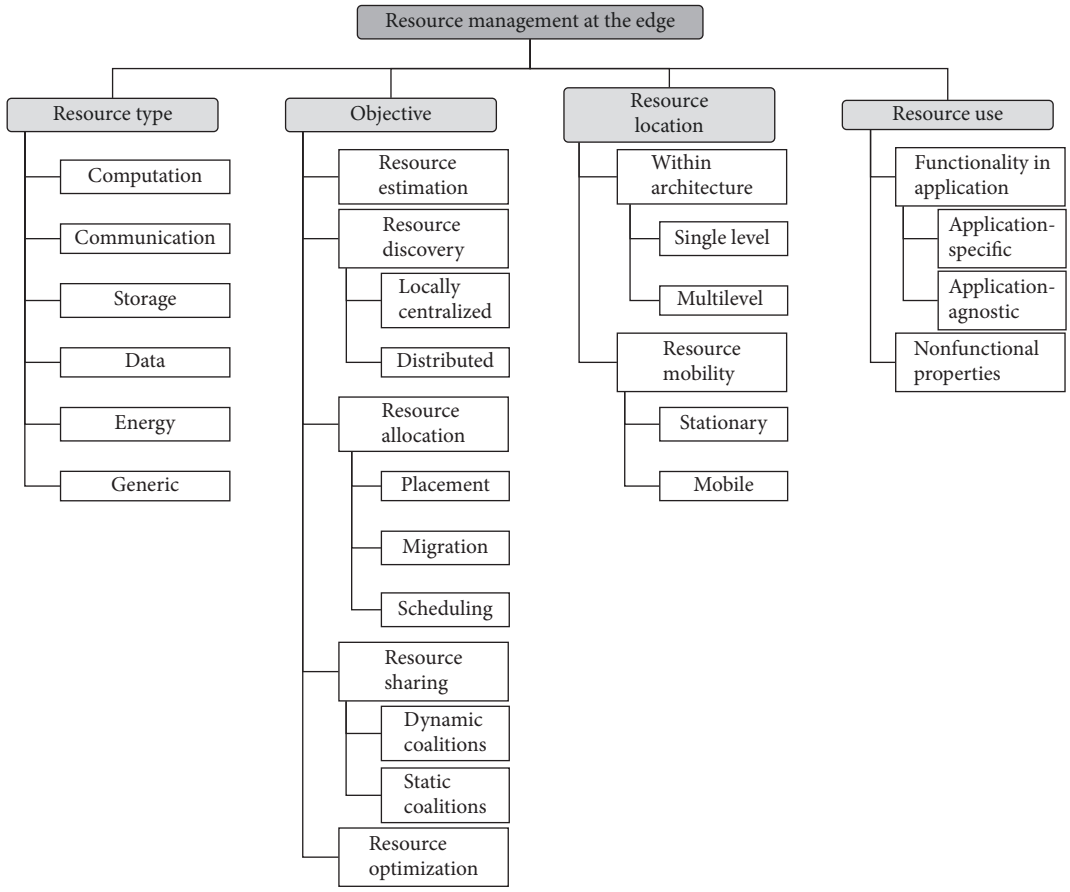

Figure 2: A taxonomy of resource management at the edge.

or generic. As it can be seen, the vast majority of the surveyed articles focus on several resources. Therefore, this section will present the common combinations of resources described above and presented in Figure 2.

3.1. Single Resource Focus. Even though the majority of the surveyed papers choose to focus on several resources, some papers focus on only one resource type. We present those papers in this subsection and then move on to multiresource cases.

3.1.1. Generic. When focusing on a single resource type, most of the works use a generic one, which is used as an abstraction for actual resources.

The abstraction used varies in various articles. For example, Penner et al. [56] work with device capabilities as an abstraction when proposing resource assignment algorithms. Other works, such as Aazam et al. [43, 58], define a new conceptual unit. "Virtual Resource Value" is the unit for any resource, which is then mapped to physical resources according to the type of service and current policies of the cloud service provider.

Sometimes the abstraction is at an even higher level: Wang et al. [77] use generic cost functions that can be used to model many aspects of performance such as monetary cost, service access latency, amount of processing resource consumption, or a combination of these. When proposing a method for online service placement, they, however, analyze its performance for a subset of cost functions related to resource consumption with the claim that this subset is still general.

3.1.2. Energy. Some works focus solely on energy, which is especially important at the edge since devices, in particular end devices, are often resource-constrained. For example, Mtibaa et al. [83] perform offloading between end devices in order to maximize the group lifetime.

Still considering only energy but with another perspective, Borylo et al. [65] classify data centers in two categories (green and brown depending on which source of energy they use) and then use a latency-aware policy to choose a data center for serving a request.

3.1.3. Other. There are works that consider a minimum computational resource unit per device. For example, Fricker et al. [69] use servers as an abstraction (one request occupies one server).

Data as a resource, in addition to sensor data mentioned earlier, can also be seen as content. Gomes et al. [13] propose an algorithm for content migration at the edge, together

Table 1: Surveyed articles according to architecture category from Figure 1 and resource type.

	Article	Computation	Communication	Storage	Data	Energy	Generic
Edge server	Liu et al. [59]	✔	✔				
	Confais et al. [60]		✔	✔			
	Aazam et al. [43]						✔
	Arkian et al. [61]	✔	✔	✔			
	Aazam and Hu [58]						✔
	Fan et al. [62]	✔				✔	
	Oueis [63]	✔	✔			✔	
	Tang et al. [64]	✔	✔				
	Borylo et al. [65]					✔	
	Yousaf and Taleb [48]	✔	✔	✔			
	Wang et al. [49]	✔	✔				
	Gu et al. [66]	✔	✔	✔			
	Tärneberg et al. [67]	✔	✔				
	Plachy et al. [68]	✔	✔				
	Gomes et al. [13]				✔		
	Fricker et al. [69]	✔					
	Rodrigues et al. [70]	✔	✔				
	Zhang et al. [71]	✔			✔		
	Bittencourt et al. [72]	✔	✔				
	Zamani et al. [73]	✔	✔				
	Valancius et al. [45]		✔	✔		✔	
	Chen and Xu [74]	✔	✔			✔	
	Wang et al. [75]	✔	✔	✔			
	Yi et al. [76]	✔	✔				
	Wang et al. [77]						✔
	Sardellitti et al. [78]	✔	✔			✔	
	Singh et al. [44]	✔	✔				
Coordinator device	Nishio et al. [52]	✔	✔		✔	✔	
	Skarlat et al. [50]	✔	✔	✔	✔		
	Borgia et al. [79]		✔		✔		✔
	Athwani and Vidyarthi [80]	✔	✔			✔	
	Arkian et al. [51]	✔	✔	✔			
	Penner et al. [56]						✔
	Bianzino et al. [81]		✔			✔	
	Habak et al. [20]	✔	✔		✔		
Device cloud	Liu et al. [54]					✔	✔
	Mascitti et al. [53]	✔	✔				
	Liu et al. [55]		✔				✔
	Meng et al. [46]	✔	✔				
	Qi et al. [82]					✔	✔
	Mtibaa et al. [83]					✔	

with mobility prediction as an enabler within their new mobile follow-me cloud architecture. This work builds upon the initial follow-me cloud proposal by Taleb and Ksentini [12].

3.2. Multiple Resource Focus. All other surveyed articles are focusing on multiple resource types. In this section, we group the papers according to the different combinations of resources they consider.

3.2.1. Computation and Communication. The most common combination of resource types studied is computational and communicational resources together. Thus, we begin by considering works that study this combination and in one case together with data.

Liu et al. [59] consider wireless bandwidth and computing resource when deciding to handle a request either in a cloudlet or in the cloud. Another example is the work by Bittencourt et al. [72], who consider bandwidth between the

cloud and cloudlet, as well as cloudlet processing capabilities when evaluating different scheduling strategies.

Computational resources can be addressed at a physical level, for example, discussing CPU cycles, or at a conceptual level, for example, use of virtual machines (VMs) as resource elements. In the surveyed articles, Wang et al. [49] consider CPU cycles, Singh et al. [44] consider Millions of Instructions per Second (MIPS), and Rodrigues et al. [70] consider the number of processors per cloudlet. At a conceptual level, Zamani et al. [73] consider different computing resources based on the average number of tasks completed per unit of time, and Plachy et al. [68] allocate computational resources in the form of VMs.

Sometimes the VMs are used as a means to ensure that a task can run given enough underlying resources in the device hosting the VM, for example, in the work by Tärneberg et al. [67].

Instead of using VMs, Yi et al. [76] adopt lightweight OS-level virtualization and a container technique, arguing that resource isolation can be provided at a much lower cost using OS-level virtualization. They also pinpoint that the creation and destruction of container instances are much faster and thus enable the deployment of an edge computing platform with minimal efforts.

As in Section 3.1.3, some works consider a minimum resource unit that corresponds to a device. For example, Meng et al. [46] consider one vehicle as the minimal computing resource unit. Vehicles are aggregated in a resource pool together with communication resources and resource units from the cloud and the edge.

Communication power needed can be considered as a part of the cost when sharing resources [64]. In contrast, communication can be characterized by a delay term impacting the task completion time, like [44, 53, 73].

Finally, Habak et al. [20] consider computation, communication, and data in femtoclouds. The data considered gives information about task dependencies in order to determine in which order the tasks need to be executed and which ones can be run in parallel.

3.2.2. Computation, Communication, and Storage. Other works, in addition to the computation and communication resource types, also include storage in their study.

For example, Arkian et al. [51] tackle resource issues in vehicular clouds by considering all three resource types. Elsewhere, crowdsensing is tackled with the same resource considerations [61].

Another example is the work by Skarlat et al. [50], where they model service demands and a specific kind of resource (sensor data) as well as the computational and storage resources. In this work, communication is considered as a delay term.

VMs can also be considered as an encapsulation of the above three resources in methods that ensure the underlying resources in the device hosting the VM are adequate [66].

Still considering virtualization, Wang et al. [75] propose a system architecture where applications' requests contain computing complexity and storage space requirements. Those requirements are then translated by a SDN controller node into computing power requirements, bandwidth volumes, or requirements on security groups. When trying to allocate more computing and bandwidth resources in an emergency situation, their system will do it by creating new VMs.

Finally, in addition to considering computation, communication, and storage, Yousaf and Taleb [48] emphasize the fact that different resources should not be considered in isolation as there are interactions between them. Thus, they describe and use the concept of resource affinity in their scheme.

3.2.3. Computation, Communication, and Energy. Another combination studied by several of the surveyed articles is computation, communication, and energy resource types.

Athwani and Vidyarthi [80] aim at making resource discovery energy-efficient in order to save battery. Nishio et al. [52] consider energy efficiency in their algorithms but at a more general level, without battery life considerations.

Oueis [63] focuses on energy-efficient communication with the aim of minimizing the communication power needed. Similarly, when studying edge collaboration in ultra-dense small base stations networks with trust considerations, Chen and Xu [74] consider computing (CPU cycles per second), communication as radio-access provisioning, and energy used for both transmission and computation.

Sardellitti et al. [78] propose an algorithmic framework to solve the joint optimization problem of radio and computational resources with the aim of minimizing the overall energy consumption of the users while meeting latency constraints. They first present a solution for the single-user case and then consider the case of offloading with multiple cells in a centralized and a distributed manner.

When considering energy as a resource, a comprehensive discussion of interactions between multiple actions is mapped to energy apportionment policies by Vergara et al. [84]. However, since this work considers edge-/cloud-specific apportionments as one among many application areas, that is, addresses energy sharing in a much wider context, we do not further consider it in our classifications.

3.2.4. Combinations Including Generic Resources. We now consider generic resources in association with other resource types, such as energy or communication.

First, Liu et al. [54] consider abstract tasks and resources to address energy efficiency. They switch between a centralized and a flooding mode depending on energy consumption while keeping the expected value of RIA (Resource Information Availability), which is their quality metric. Qi et al. [82] choose to abstract resources as services and consider energy consumption in the end device when taking an offloading decision.

Regarding communication, Liu et al. [55] use the notion of generic resource (when referring to a combination of bandwidth and CPU available for sharing) as well as concrete bandwidth when nodes are at contact range. Borgia et al. [79] consider data-centric service providers having storage, computing, and networking capabilities but in their evaluation abstract away the storage and computing resources by

only considering the extent to which services are waiting for resources on the provider side.

3.2.5. Other Combinations. Not all works considering computation also consider communication. Less common combinations including computational resources are those with energy and data.

With regard to energy, Fan et al. [62] present a virtual machine migration scheme that aims at using as much green energy as possible in the context of green cloudlet networks.

Data and computation are the focus of Zhang et al. [71] who studied distributed data sharing and processing in order to use data coming from different stakeholders for new IoT applications and propose a new computing paradigm called Firework.

Less common combinations including communication resources include storage and energy resource types.

Confais et al. [60] present how a storage service can be provided for fog/edge infrastructure, based on the InterPlanetary File System, and scale-out network-attached systems. Their aim is to propose a service similar to the Amazon Simple Storage Service solution (https://aws.amazon.com/fr/s3/) for the edge.

Adding energy to storage and communications resources, Valancius et al. [45] consider energy-efficient algorithms when introducing a new distributed data center infrastructure for delivering Internet content and services.

Finally, Bianzino et al. [81] study the trade-off between bandwidth and energy consumption when an end device has access to multiple networking interfaces and can switch between them. They aim at energy efficiency but use an abstract model of power usage based on the amount of data being shuffled.

3.3. Summary of Resource and Architecture Choices. In this section, we have presented the surveyed articles depending on their resource focus. Examining the collection of papers above, resource studies so far seem to focus on computation and communication resources to a greater extent. Moreover, data as a resource is a potential not extensively explored. Similarly, energy is underrepresented among resources studied.

Furthermore, it is noticeable that storage is not the main focus of attention. It could be due to the fact that the cloud is available as a fall-back in many cases. It could also be the case that persistent data storage is not the main focus of most of the applications considered at the edge. Rather, the service or completed task is the main purpose. Another reason could be that presently there are not many critical use cases with latency-constrained storage, but this may change when more and more IoT devices appear in the field. An alternative explanation could be that the authors choose to focus on a reduced set of resources for ease of presentation thinking that the work can be extended to other resources such as storage. Such claims, however, have to be considered with care as this is ignoring the fact that there could be interactions between resources as studied by Yousaf and Taleb [48].

Some resources are dealt with mainly as physical elements, whereas others naturally lend themselves to be defined in abstract ways. For example, sensors are present in the end devices, which can produce useful data needed for the completion of the task (as in [50, 52, 56, 71]), whereas bandwidth (throughput) is a natural abstraction for distinguishing between different radio interfaces or different physical environments (abstracting the impacts of reduced signal strength, interference, etc.).

Moreover, when using a generic resource representation, it is easier to combine several resource types or to combine resources with other performance-related considerations, one example being the generic cost function in the work by Wang et al. [77]. In their performance analysis, they define local and migration resource consumption that can be related, for example, to CPU and bandwidth occupation or the sum of them.

Another point to note is that the first architectural instance (edge server) is the most predominant structure used in the surveyed papers.

4. Objective

A major classification represented in this taxonomy is the objective of resource management. Resource management at the edge can be decomposed into several areas addressing different problems, as shown in the branches under objective in Figure 2. In Table 2, we present which surveyed article addresses which problem(s) and we describe those problems in the following subsections. As it can be seen in the table, one surveyed work can address several of the areas.

The resource management objective is orthogonal to the resource types presented in Section 3 but a discussion of the relationship between objectives of resource management and resource types is conducted in our summary in Section 4.6.

4.1. Resource Estimation. One of the first requirements in resource management is the ability to estimate how many resources will be needed to complete a task or to carry a load. This is important, especially for being able to handle fluctuations in resource demand while maintaining a good quality of service (QoS) for the user. On the supply side, resources at the edge can be mobile and thus become unreachable, which makes them less reliable than resources in a data center. On the demand side, user mobility implies that there can be sudden user churn, with the corresponding dynamic requests having to be handled by the edge.

In their work, Liu et al. [54] use the average of historical data in order to predict the characteristics of resource distribution and usage for the next time slot. The term fog is used by Aazam et al. [43] who propose that it can be used to perform future resource consumption estimation as a first step for allocating resources in advance. They formulate an estimation mechanism that takes into account the reliability of the customer, using what they call the relinquish probability. In another article, Aazam and Huh [58] present the same idea

Table 2: Surveyed articles according to architecture category from Figure 1 and objective of resource management.

		Objective				
		Resource estimation	Resource discovery	Resource allocation	Resource sharing	Resource optimization
Edge server	Liu et al. [59]			✔		✔
	Confais et al. [60]			✔		
	Aazam et al. [43]	✔				
	Arkian et al. [61]			✔		✔
	Aazam and Hu [58]	✔				
	Fan et al. [62]			✔		✔
	Oueis [63]			✔		✔
	Tang et al. [64]				✔	
	Borylo et al. [65]			✔		
	Yousaf and Taleb [48]			✔		✔
	Wang et al. [49]			✔		✔
	Gu et al. [66]			✔		✔
	Tärneberg et al. [67]			✔		✔
	Plachy et al. [68]			✔		
	Gomes et al. [13]			✔		
	Fricker et al. [69]			✔		
	Rodrigues et al. [70]			✔		✔
	Zhang et al. [71]				✔	
	Bittencourt et al. [72]			✔		
	Zamani et al. [73]		✔	✔		✔
	Valancius et al. [45]			✔		✔
	Chen and Xu [74]				✔	
	Wang et al. [75]			✔		
	Yi et al. [76]			✔		✔
	Wang et al. [77]	✔		✔		✔
	Sardellitti et al. [78]			✔		✔
	Singh et al. [44]			✔		
Coordinator device	Nishio et al. [52]				✔	✔
	Skarlat et al. [50]			✔	✔	✔
	Borgia et al. [79]			✔		
	Athwani and Vidyarthi [80]		✔		✔	✔
	Arkian et al. [51]		✔		✔	✔
	Penner et al. [56]			✔		
	Bianzino et al. [81]				✔	✔
	Habak et al. [20]	✔		✔	✔	✔
Device cloud	Liu et al. [54]	✔	✔			✔
	Mascitti et al. [53]			✔		
	Liu et al. [55]				✔	✔
	Meng et al. [46]			✔		✔
	Qi et al. [82]			✔		✔
	Mtibaa et al. [83]	✔		✔	✔	✔

but with an emphasis on how different customers can be charged for the service. Another work by Mtibaa et al. [83] estimates power consumption in order to maximize device lifetime.

Wang et al. [77] use a look-ahead window for prediction into the future in order to minimize cost over time. They study the optimal size for such a window and propose an algorithm using binary search to find this size which they

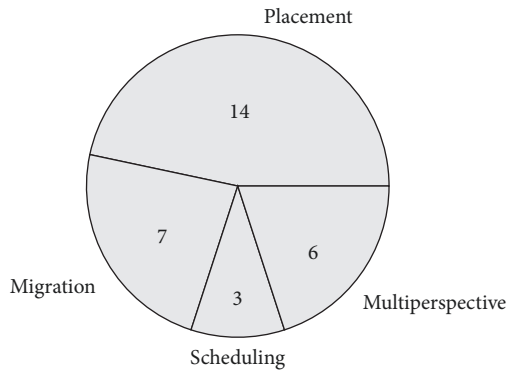

Figure 3: Distribution of resource allocation approaches in the surveyed articles.

evaluate as accurate as it gives results close to the size giving the lowest cost. However, the actual prediction mechanism is assumed to be available.

With respect to computational resources, Habak et al. [20] are estimating the task requirements within a job analyzer. They evaluate the sensitivity of their mechanisms to estimation errors and find that the pipeline job model is insensitive to such errors, whereas the general parallel path model starts exhibiting a significant increase of job completion time if the estimation error variance exceeds 30%.

There are of course many earlier works that use sophisticated prediction mechanisms for estimating future loads in cloud environments (e.g., [85]) but our focus has been on edge-related papers and instances of estimation therein.

4.2. Resource Discovery. As opposed to the estimation problem that relates to the demand side, resource discovery is about the supply side. A management system needs to know which resources are available for use, where they are located, and how long they will be available for use (especially if the resource providing device is moving or it is battery-driven). This area is especially important at the edge, where every resource is not under the control of the system at all times, so the supply is volatile.

The collaboration at the edge can take the form of clusters, as advocated by Athwani and Vidyarthi [80]. They present an algorithm for forming clusters of devices and performing resource discovery within the cluster. Their strategy is that each member of the cluster will inform the cluster head about their available resources and all requests for resources are handled by the cluster head. From their evaluation with respect to energy consumption and delay, they conclude that maintaining the cluster consumes extra energy, especially if the devices are very mobile. Arkian et al. [51] also present a solution using clusters and an algorithm for selecting the cluster head, that is, the vehicle that will be responsible for maintaining the vehicular cloud resources. They use fuzzy logic and a reinforcement learning technique. In order to select the best vehicle, they need to know which vehicle possesses the best communication to the edge node located on the road-side, hence performing resource discovery. This is done in a similar way to earlier work [80]; that is, each potential cluster head node sends a message to the edge node in order to evaluate the link quality before doing the selection. Therefore, those works use a *locally centralized* strategy for resource discovery.

However, using a locally centralized strategy comes at the cost of the necessity to regularly update the node gathering the resource information. Such updates are costly, for example, in terms of energy consumption, as studied by Liu et al. [54]. They propose an algorithm enabling a switch between a locally centralized mode and a *distributed* mode. In the locally centralized mode, end devices propagate their resource information/request to a specific node. In the distributed mode, end devices look for resources in the neighboring devices by using ad hoc WLAN. They qualify their strategy as adaptive as it takes into account the current characteristics of resource distribution and usage in the network. When evaluating the energy consumption of two variants of the adaptive strategy, these perform close to the ideal energy consumption (10% to 13% more energy) and both perform better than strategies using only a distributed or locally centralized mode.

Finally, Zamani et al. [73] use a framework called Comet-Cloud, which performs resource discovery for video analysis and compare the benefit gained to a solution in the cloud.

4.3. Resource Allocation. Resource allocation can be tackled from two different perspectives: *where* to allocate (both initially, but also where and when to perform a migration if needed) and *when and how much* to allocate. Among the dominant approaches to allocation, we find the following three perspectives: placement (14 articles), migration (7 articles), and scheduling (3 articles), as well as a multiperspective one (6 articles) as shown in Figure 3.

In what follows, we group papers that have a single perspective under Sections 4.3.1, 4.3.2, and 4.3.3 and then move on to papers where several perspectives are present.

4.3.1. Placement. Most of the surveyed works emphasize the first perspective, that is, where should the task be executed and the resource allocated for the best possible execution. The definition of best execution varies depending on the considered system and the focus of the research.

Load distribution to achieve lower latency has attracted attention in a number of surveyed works, and it can be seen as an instance of placement. Fricker et al. [69] propose an offloading strategy between edge data centers under high loads that show the benefit of having a larger data center as back-up for a small one. Latency is also the focus of a study by Borylo et al. [65] who investigate dynamic resource provisioning. They present a policy in which the edge can use the cloud in compliance with the latency requirements of the edge but enables better energy efficiency by using resources in data centers powered by green energy.

Also focusing on energy, Mtibaa et al. [83] propose a power balancing algorithm in which a device decides whether to offload and to which other device depending on the energy left in the devices' batteries. In a single-hop scenario, their solution extended the time before the first device of the group runs out of battery by 60% (from 40 minutes to 2 hours) compared to a greedy solution.

Oueis [63] tackles the issue of load distribution and resource allocation in small cell clusters. She formulates a joint computational and communication resource allocation and optimization problem in a multiuser case with a focus on latency and power efficiency. Similarly, Sardellitti et al. [78] study an offloading problem when the end users are separated into two groups: those who need computation offloading and those who do not. They propose a method to jointly optimize communication and computation resources, where both user groups compete for communication resources but only the first group competes for computation resources. They first present an algorithm for the single-user case and then two algorithms for the multiple cells case, a centralized one, and a distributed version to mitigate the communication overhead induced by the centralized approach.

Valancius et al. [45] propose a content placement strategy, where the content is movies. The focus is first on finding the optimal number of replicas of the data to be stored and then on placing the replicas on available gateways. Similarly, Qi et al. [82] present an allocation scheme, where the resource (coming from either a cloudlet or a cloud) is chosen for each task. The aim is to pick the resource from the most suitable location when the user is moving.

Wang et al. [77] study service placement in a system composed of edge server nodes and traditional cloud nodes. Simulation results with real-world traces from San Francisco taxis show that the proposed approach is close to the case of online placement when the future is known, outperforming edge-only or cloud-only solutions. Similarly, Skarlat et al. [50] present a service placement problem for IoT services.

Mascitti et al. [53] present an algorithm where an end device can choose to either use a resource from a node it is directly in contact with or compose different resources from different nodes in order to complete its task. Coming from the same research group, Borgia et al. [79] present a framework where the decision is taken to obtain a service from a local group of end devices or from the cloud, based on an estimation of the time required to obtain the service.

Confais et al. [60] propose a storage mechanism where the objects to be stored are primarily placed locally at their creation but can then be copied to another location if another edge site is requiring access to it. In vehicular networks where resources are mobile, placement has to take account of changes in location. Meng et al. [46] present a resource allocation scheme that can manage resources from a vehicular cloud, the edge, or the cloud. Their focus is on minimizing delays for the users.

In the application domain of healthcare, Gu et al. [66] include VM placement in their optimization problem for analyzing data. Their two-phase solution has a nearly optimal solution and outperforms a greedy strategy with regard to cost.

4.3.2. Migration. Still considering where the task should be executed, when it comes to virtual entities such as services, applications, tasks, and VMs, the focus could also be on how they can be moved during execution if the new location is better, that is, on migration.

For example, Tärneberg et al. [67] study application-driven placement and present a system model for mobile cloud network with a dynamic placement algorithm that guarantees application performance, minimizes cost, and tackles resource asymmetry problems. Plachy et al. [68] propose a cooperative and dynamic VM placement algorithm associated with another cooperative algorithm for selecting a suitable communication path. They use VM migration to solve user mobility problems.

Other works focus specifically on the problem of migrating resources. For example, Gomes et al. [13] present a content-relocation algorithm for migrating the content of caches present in edge devices. This needs a prediction of user mobility.

With respect to virtual computational resources, Fan et al. [62] and Rodrigues et al. [70] focus on VM migration but with different optimization objectives (increase the use of green energy and minimize delays, resp.). Yousaf and Taleb [48] propose a VM migration (and VM management) system that takes into account the relationship between resource units when making migration decisions. They present this work in the context of 5G but it should be applicable to all physical machines hosting VMs.

Finally, Penner et al. [56] introduce the term *transient cloud* and concentrate on task assignment towards a given node. They present a collaborative computing platform that allows nearby devices to form an ad hoc network and provide the ability to balance the assignments among themselves. This may be considered as a form of migration.

4.3.3. Scheduling. While there is a huge body of researches available on when and how many resources to allocate within networking and cloud-specific areas, our goal here was to identify examples where scheduling decisions are at the edge level in the sense of our terminology in Section 2.1.

Regarding when to allocate resources, Bittencourt et al. [72] study the impact of three different fog scheduling strategies on application QoS (concurrent, first come-first served, and delay-priority). For two applications studied, when more than four users are moved between cloudlets, the concurrent strategy exhibits a lot longer delays than the

sequential ones. However, this same strategy is using the network a lot less than the other two.

With the same focus, Singh et al. [44] consider only scheduling for tasks with a *private* tag. Those can only be executed on the local edge server and will be rejected if not enough resources are available. In their algorithm, tasks are considered in an earliest-deadline-first manner.

Regarding how many resources to allocate, Wang et al. [49] propose a joint cost-effective resource allocation between the mobile cloud computing infrastructures and the cloud radio access network infrastructure. If the need of the application is greater than the available computational resources, then they reduce the amount given to each virtual machine so that it fits the total amount available and adapt the data rate accordingly. They show that joint optimization with respect to cost and energy performs better compared to separate cost- and energy-optimization strategies.

Similarly, Wang et al. [75] propose elastic resource allocation for video-surveillance systems. The elasticity comes from an algorithm they propose to handle some emergency surveillance event (like tracking a criminal) that requires a sudden increase of computation and communication resources to make sure that all the possible images are analyzed within a reasonable timeframe. When such an emergency event happens, network bandwidth allocation is reconfigured and computing resources are reallocated (by launching new VMs in the impacted zone and balancing the workload on nodes). When experimenting in their physical testbed, they verified that data propagation round-trip time is about 5 times lower with edge nodes close to the cameras compared to the cloud. They also found that the time for launching new VMs in the emergency mode is between one and two minutes, which they claim is acceptable in such a scenario.

When addressing scheduling, the surveyed articles most often do it at the same time as placement or migration, which is the topic of the next subsection.

4.3.4. Multiple Perspectives. A work that tackles both perspectives is by Liu et al. [59]. It presents a multiresource allocation system that first decides whether the request should be served or rejected (admission control) and then where to run it (edge or cloud level) and finally how much bandwidth and computing resources should be allocated for this task. To do that, they use Semi-Markov Decision Processes and their aim is to maximize system benefit while guaranteeing QoS for the users. To measure the benefit, they use blocking probability and service time as metrics. When evaluating, they compare to two greedy strategies and show that their proposal outperforms the first one and provides a 90% reduction of the blocking probability with only a slight increase of service time compared to the second greedy strategy, which would be acceptable for congested situations.

In the context of video analysis, Zamani et al. [73] also studied those two perspectives. Their scheduling is based on identified chunks of video, applying two alternatives: minimizing computation time or minimizing computation costs. Their placement is done after resource discovery using CometCloud. In their evaluation, they showed that the solution using edge accepts more tasks and in particular more high-value tasks than a solution using only the cloud. Hence, the overall value obtained from the processed data is maximized at the same time as the throughput of the infrastructure.

Also in the area of video analytics, Yi et al. [76] investigate three task prioritizing schemes for scheduling the task requests at a receiving edge node. Their solution, using the flow job shop model and applying a well-known approach (Johnson's rule), aims at minimizing the makespan. Their simulations compared the approach with other strategies (Short IO First, Longest CPU Last) and found that response time was improved. Their work also includes a second perspective, by investigating three task placement schemes for collaboration within the edge level (Shortest Transmission Time First, Shortest Queue Length First, and Shortest Scheduling Latency First). Using their testbed, they found that the Shortest Scheduling Latency First achieves the best performance in terms of task completion time.

Singh et al. [44] consider both placement and scheduling with respect to semiprivate or public tasks (in addition to what was mentioned in the last subsection for private tasks). Those tasks are placed after a decision is taken for the private ones. Still considering Earliest Deadline First, the placement strategy is to try first one's own edge and then one's own cloud and if they are overloaded go to some external edge and then to an external cloud. In the evaluation, they show that, for tasks having tight deadlines, their system RT-SANE will complete a lot more tasks before their deadline than a cloud-only solution.

How many resources are to be allocated to a given IoT data generator is a topic of discussion by Arkian et al. [61], in which they first mathematically model deployment and communication costs on various fog nodes and then decide on placement of VMs to achieve lowest costs. Analyzing monetary costs for compute nodes, their fog solution decreased the cost by over 33% compared to using a cloud solution. For routing and storage monetary cost, the decrease is about 20%.

Habak et al. [20] first consider placement for deciding in which end device a task will be run. They use a path-based assignment policy with the aim of minimizing the overhead of transmitting data needed for task execution between end devices. In the evaluation, this translates into performing better than two other baseline solutions in terms of service completion time. Then, they also consider scheduling of the computation resource. This should be done in a predictable way so that the part of the system distributing the tasks can make good decisions. They propose a fair queuing based task pick-up that ensures a fair execution of the tasks belonging to different services. Moreover, they implement an early pick-up mechanism to enhance the previous mechanism so that a task with an urgent deadline but belonging to a service with a lower priority can be executed before a higher priority task if this one still meets its deadline.

While this is not the focus of the survey and as such is not included in the table, Dong et al. [86] study offloading and Earliest Deadline First scheduling within end devices. They find that one of their proposed approaches maintains good predictability for twice higher CPU utilization than

widely used approaches, while keeping energy consumption reasonable.

4.4. Resource Sharing. Resources on end devices are heterogeneous and most of the time scarce, and edge devices also have limited resources compared to (almost infinite) resources in the cloud. Sharing resources between devices or between end and edge devices aims at tackling three different issues: not having the needed resource at all in the device where the task is initiated, not having enough of it, or using other devices' resources in order to get a faster completion of the task.

Sharing resources is typically realized by pooling resources in the local vicinity of client nodes. This can extend to the edge domain (clustering edge servers) or remain at end devices. The latter is investigated by Skarlat et al. in so-called *fog colonies* [50], by Arkian et al. within vehicular clusters [51], or by Bianzino et al. [81] for uploading data streams in presence of mobility.

We can classify the surveyed articles into two categories according to whether they include how to form the groups of devices that will share resources or if they assume that the formation is already done and focus on the actual sharing. We call these two categories *dynamic coalitions* and *static coalitions*, respectively.

Starting with dynamic coalitions, Chen and Xu [74] and Bianzino et al. [81] include the formation of device coalitions. Chen and Xu [74] do it using a coalition game incorporating trust considerations. When the supply matches the demand, they found that using a coalition can lead up to 40% lower weighted cost (including latency and monetary considerations) compared to a noncooperative scenario. When there is overload or light workload, it is either not possible or not needed to collaborate and the gain is very low. Bianzino et al. [81] express resource sharing as an optimization problem, where the aim is to create as few and large groups as possible to minimize the number of high-energy interfaces that will be used. They evaluate that their algorithm leads to over 60% energy saving of the total energy consumed by the end devices.

Still using dynamic coalitions, Arkian et al. [51] and Athwani and Vidyarthi [80] propose methods to create clusters. The former compare their method to an earlier baseline and achieve 3 times lower service discovery delay and 4,5 times lower service consumption delay for a small number (50) of vehicles. The latter show that energy consumption is similar to a centralized approach, while the delay is closer to a flooding approach (i.e., low in both cases).

However, creating and maintaining a group of devices that can share their resources has a cost, for example, shown by Athwani and Vidyarthi [80] who concluded that maintaining the cluster consumes extra energy, especially if the devices are very mobile. This is why it is beneficial to do the resource sharing in two phases, where the first phase is deciding whether the device gains more by working alone or joining a coalition, and the second one is deciding if the device will consume others' resources [81, 87]. Yu et al. [87] show that their cooperative solution improves user QoS (defined by how much computing and how many bandwidth resources are allocated to a user) by 75%. However, this paper is using traditional cloud resources and not edge, so it is not included in the tables.

Moving to static coalitions, Skarlat et al. [50] consider resources shared between two neighbor fog colonies and achieve a 35% reduction of execution cost compared to a cloud-only strategy. Regarding data, a resource that many stakeholders may be interested in sharing, Zhang et al. [71] present a data sharing framework called Firework. They include two case studies, including the search for a person with the help of multiple cameras from different owners.

Some researchers, such as Liu et al. [55], try to exploit opportunistic contacts between the devices, creating a resource sharing mechanism that enables faster task completion. They propose different models for calculating task latencies and their approximation algorithm performs better than two other strategies. Similarly, Mtibaa et al. [83] define three mobile device clusters (one hop, two hops, and opportunistic) that can share their resources. Their aim is to share resources in order to get the longest possible network lifetime, that is, saving as much energy as possible through offloading to another device so that the devices can stay on longer. They identify two important topological factors: number of hops and disconnection rate due to mobility.

Resource sharing can perhaps speed up the execution of a task, but Nishio et al. [52] argue that this is not bringing any advantage for the user if we do not consider task dependencies in order to provide a service to the user. They provide the example of a GPS service: if the best route calculation is very fast but the downloading of the map is not, the service to the user will not get faster as both are needed. Habak et al. [20] consider sharing of end device resources belonging to a femtocloud in order to execute tasks. In their system, the owner of the end device can configure how they want to share resources via their personalized resource sharing policies.

Finally, even if resource sharing can bring benefits for a group of end devices, it is not obvious that users will agree to share their resources, especially if they are always on the providing side. Therefore there is a need to develop incentives for resource sharing such as works by Tang et al. [64], Bianzino et al. [81], and Chen and Xu [74]. The following mechanisms are provided in the above works, respectively:

(a) A double bidding mechanism for demander and supplier of resources where the focus is on how to encourage mobiles with resources to share them

(b) A mechanism for lending energy to vicinity nodes which is rewarded and can be used in future scenarios when the lending node itself needs energy

(c) Payment incentives for lending out resources

On the same topic, Habak et al. [20] performed a pilot study to identify effective incentive mechanisms. They studied the willingness of around 50 students to share their resources in 4 scenarios and found out that they would agree to share their resources if they are getting compensation (e.g., money) for it or if the reason for the computation taking place is significant (e.g., emergencies).

4.5. Resource Optimization. A fifth objective pursued in the surveyed works is to optimize the resource use at the edge. This is usually a joint objective together with one of the previously described objectives. Which aspect should be optimized and the associated constraints vary among the surveyed works but the three main ones are QoS (often understood as latency), energy, and operational cost. How the optimization problem is formulated and solved also varies, and we present those variations in this section.

First, some articles consider selecting the optimum solution by comparing the results from different candidates and selecting the minimum/maximum value depending on the objective. For example, Yousaf and Taleb [48] select the value maximizing the resource utilization, Athwani and Vidyarthi [80] use the minimum value of a custom function to select the cluster head, and Mtibaa et al. [83] select the configuration maximizing the estimated remaining energy.

Another group of works solve their optimization problem using linear programming [45, 59] or an approximation based on linear programming [55].

A third group of works use integer linear programming [50, 81] or mixed-integer linear programming [62]. Qi et al. [82] formulate their task allocation problem using integer programming and solve it by a self-adaptive learning particle swarm optimization algorithm. First formulating using mixed-integer nonlinear programming, Arkian et al. [61] then linearize the problem and solve it using mixed-integer linear programming. Gu et al. [66] do the same and then use heuristics. Using a different approach, Yi et al. [76] first formulate a mixed-integer nonlinear programming problem but then relax the integer constraints and use sequential quadratic programming for solving.

Some works focus on convex problems, like Wang et al. [77] who use an approximation algorithm in the online case and Nishio et al. [52] who use a heuristic. Starting with nonconvex problems, Oueis [63] casts them into convex ones and Wang et al. [49] first use a Weighted Minimum Mean Square Error-based method on their nonconvex problem to obtain a convex problem that they apply the block coordinate descent method to for solving. Finally, Sardellitti et al. [78] have an optimization problem in the multiple-cells case which is nonconvex and they solve it by developing a method based on Successive Convex Approximation for the centralized approach. For the distributed approach, they choose the approximation functions in a way that allows decomposition in smaller subproblems solvable in parallel.

A further group of works propose their own algorithm or heuristic. Tärneberg et al. [67] approximate an exhaustive search approach yielding an optimal solution but having exponential computation complexity with an iterative local search algorithm finding a local optimal solution. Zamani et al. [73] implement an optimization strategy where constraints on computation time and cost are enforced using an admission control strategy. Wang et al. [77] present a binary search algorithm for finding the optimal look-ahead window size, and Habak et al. [20] propose an algorithm in order to do deadline-based optimization when a helper has to handle multiple tasks belonging to different services. Finally, Liu et al. [54] propose a heuristic algorithm that uses different statistics to estimate the energy that is going to be consumed in each of the two possible modes during a time slot and chooses which mode to use depending on this and other parameters.

Other methods can be used to compare heuristics with baselines or to solve a formulation in a custom form. In the offline case, Wang et al. [77] show that their problem is equivalent to the shortest-path problem and solve it by using dynamic programming. Meng et al. [46] solve Bellman equations recursively, Rodrigues et al. [70] use integration techniques, and Arkian et al. [51] consider fuzzy logic and Q-learning.

4.6. Summary of Objectives in Resource Management. By far, the most active area of research in the edge resource management is resource allocation, as visible in Table 2. This is followed by optimization as a goal, where we see a great majority of papers present. Among the objectives from our taxonomy, resource estimation and resource discovery are least studied. Resource sharing, to the extent it is used, is well represented among the second and third types of architectures in Figure 1, that is, coordinator device and device clouds, but not in the first type of architecture (edge server).

Somewhat surprisingly, while scheduling is a major topic in cloud systems, the edge-specific literature does not consider it as the main problem, as evident from fewer works addressing scheduling compared to placement and migration. Where autoscaling is mentioned in an edge context, authors typically deal with offloading to the cloud, which was not the focus of our work. There are several excellent surveys already covering these. The work by Wang et al. [88] addressing autoscaling and the edge is among few exceptions, so we did not create a special category for this type of work.

While the previous breakdown was done in a resource independent manner, it is also interesting to consider the resource type studied with regard to the resource management objectives. Table 3 thus combines the information contained in Tables 1 and 2 to give us this view. Not surprisingly, most of the articles consider computation and communication for resource allocation and optimization. Quite expected as well, the proportion of resource sharing articles (from Table 2) considering energy as a resource (45% according to Table 3) is higher than the proportion of, for example, resource allocation articles considering energy (23%), as an incentive to share resources is when you consider energy-constrained devices. It is interesting to note that, in the surveyed works, resource estimation is most often done for a generic resource type and that none of the articles combined resource estimation and storage and resource discovery and data.

5. Resource Location

Computing at the edge differentiates itself from regular cloud computing with the fact that resources used can belong to different levels. It is indeed not uncommon to use resources at the edge level primarily but also from the cloud level if

Table 3: Surveyed articles according to resource type and objective of resource management.

		Objective				
		Resource estimation	Resource discovery	Resource allocation	Resource sharing	Resource optimization
Resource type	Computation	[20]	[51, 73, 80]	[20, 44, 46, 48–50, 53, 59, 61–63, 66–70, 72, 73, 75, 76, 78]	[20, 50–52, 64, 71, 74, 80]	[20, 46, 48–52, 59, 61–63, 66, 67, 70, 73, 76, 78, 80]
	Communication	[20]	[51, 73, 80]	[20, 44–46, 48–50, 53, 59–61, 63, 66–68, 70, 72, 73, 75, 76, 78, 79]	[20, 50–52, 55, 64, 74, 80, 81]	[20, 45, 46, 48–52, 55, 59, 61, 63, 66, 67, 70, 73, 76, 78, 80, 81]
	Storage		[51]	[45, 48, 50, 60, 61, 66, 75]	[50, 51]	[45, 48, 50, 51, 61, 66]
	Data	[20]		[13, 20, 50, 79]	[20, 50, 52, 71]	[20, 50, 52]
	Energy	[54, 83]	[54, 80]	[45, 62, 63, 65, 78, 82, 83]	[52, 74, 80, 81, 83]	[45, 52, 54, 62, 63, 78, 80–83]
	Generic	[43, 54, 58, 77]	[54]	[56, 77, 79, 82]	[55]	[54, 55, 77, 82]

required. Moreover, end devices and sometimes edge devices do not have to be stationary as in a data center. Note that here we make a distinction between mobility on the demand side and mobility on the supply side. Even though the demand side clients are almost always mobile, the infrastructure that supplies the adequate resources has been invariably stationary in the past.

In this section, we first look at where the *managed* resources considered are located within the architectures presented in Figure 1. We then shift focus and look at the same set of resources again but this time studying their mobility.

5.1. Location within the Architecture. Edge resource management is actually not only about managing resources located at the edge level as a study of the managed resources' location in the surveyed work reveals. This study is presented in Table 4.

5.1.1. Single Level. As expected when surveying edge resource management papers, a large part (54%) of those consider managed resources located only at the edge level, for example, the works by Arkian et al. [61], Fan et al. [62], Gomes et al. [13], Yousaf and Taleb [48], Chen and Xu [74], Sardellitti et al. [78], and Wang et al. [75].

Aazam et al. [43] consider resources located at only one physical location, a fog node, but considering resources within the same architectural level most often does not mean that the resources are located at the same physical location. For example, Oueis [63] considers resources on different cells and Gu et al. [66] and Plachy et al. [68] consider resources on different base stations. Fricker et al. [69] and Rodrigues et al. [70] consider task placement and migration on different types of edge devices (data centers for the former and cloudlets for the latter).

Essentially refining our architecture, some works distinguish different levels in the same architectural level from our Figure 1. For example, Wang et al. [49] consider transmission in the access network and computation in a mobile cloud computing architecture. Tärneberg et al. [67] consider that data centers at the edge can have a different distance to the device and different sizes.

Among the surveyed works, two works consider resources located only at the device level but where the management is performed at the edge, Tang et al. [64] and Nishio et al. [52], who consider resources present on different end devices.

There is no work considering managed resources located at the cloud level only as those were on purpose considered out of the scope of this survey.

5.1.2. Multilevel. We observe that resources do not need to belong to the same architecture level. Among the multilevel works, the most common is to use resources located both at the edge and at the cloud level. This is the case in the works by Liu et al. [59], Borylo et al. [65], Valancius et al. [45], Yi et al. [76], Wang et al. [77], and Singh et al. [44]. Specifically, Skarlat et al. [50] and Bittencourt et al. [72] favor using edge resources over cloud resources. Liu et al. [54] use resources in the device/edge level or in the cloud depending on the availability of the resources and Confais et al. [60] work with different storage locations at the edge or cloud level.

This is, however, not the only combination and Zhang et al. [71] work with data as a resource that can be located both in the end devices and at the edge. This combination is also used by Bianzino et al. [81] and Habak et al. [20], where an end device is promoted to an edge role.

Finally, combining the three levels, Zamani et al. [73] use resources on the device, on the network path to the cloud (edge level), and in the cloud level.

5.2. Resource Mobility. In an edge context, it is not obvious that resources located in the lower two levels of the architecture will be stationary or mobile. Therefore, it is interesting to study the mobility of the managed resources in the surveyed articles.

5.2.1. Stationary Resources. Most of the surveyed articles (71%) consider resources that are stationary only. This can be because the architecture/application considered does not have mobile resources or for simplification reasons. The latter is found in works where the architecture presented has

Table 4: Managed resources and their supply-side mobility.

	Article	Managed resources' location		
		Device level	Edge level	Cloud level
Edge server	Liu et al. [59]		Stationary	Stationary
	Confais et al. [60]		Stationary	Stationary
	Aazam et al. [43]		Stationary	
	Arkian et al. [61]		Stationary	
	Aazam and Hu [58]		Stationary + mobile	
	Fan et al. [62]		Stationary	
	Oueis [63]		Stationary	
	Tang et al. [64]	Stationary		
	Borylo et al. [65]		Stationary	Stationary
	Yousaf and Taleb [48]		Stationary	
	Wang et al. [49]		Stationary	
	Gu et al. [66]		Stationary	
	Tärneberg et al. [67]		Stationary	
	Plachy et al. [68]		Stationary	
	Gomes et al. [13]		Stationary	
	Fricker et al. [69]		Stationary	
	Rodrigues et al. [70]		Stationary	
	Zhang et al. [71]	Stationary	Stationary	
	Bittencourt et al. [72]		Stationary	Stationary
	Zamani et al. [73]	Stationary	Stationary	Stationary
	Valancius et al. [45]		Stationary	Stationary
	Chen and Xu [74]		Stationary	
	Wang et al. [75]		Stationary	
	Yi et al. [76]		Stationary	Stationary
	Wang et al. [77]		Stationary	Stationary
	Sardellitti et al. [78]		Stationary	
	Singh et al. [44]		Stationary	Stationary
Coordinator device	Nishio et al. [52]	Stationary		
	Skarlat et al. [50]		Stationary	Stationary
	Borgia et al. [79]		Mobile	Stationary
	Athwani and Vidyarthi [80]		Mobile	
	Arkian et al. [51]		Mobile	
	Penner et al. [56]		Mobile	
	Bianzino et al. [81]	Mobile	Mobile	
	Habak et al. [20]	Mobile	Mobile	
Device cloud	Liu et al. [54]		Stationary	Stationary
	Mascitti et al. [53]		Mobile	
	Liu et al. [55]		Mobile	
	Meng et al. [46]		Stationary + mobile	Stationary
	Qi et al. [82]		Mobile	Stationary
	Mtibaa et al. [83]		Mobile	

resources that are theoretically mobile but where this part is ignored in the solution or evaluation presented, for example, in [54] or [52].

This preponderance of stationary resources may be explained by the fact that those works consider edge as an extension of the cloud, which has only stationary resources.

5.2.2. Mobile Resources. Having mobile edge devices and thus mobile resources obviously creates lots of challenges such as how to handle the unreliable connectivity of those resources and how to provide seamless handovers. Thus, having mobile resources introduces another level of complexity in resource management algorithms.

Different mobility models are used; for example, Penner et al. [56] model departure and arrival times using statistical models, which is similar to what is used by Bianzino et al. [81]. Also using statistical models, Habak et al. [20] model arrival rate and presence time. In those statistical models, arrivals are modeled using a Poisson distribution, departure most often using an exponential distribution, and presence time using a normal distribution. Another model that is relatively common is the Random Way Point Model, used by Mascitti et al. [53] and Liu et al. [55].

With a different and more uncommon approach, Arkian et al. [51] consider the speed of the vehicles, and Athwani and Vidyarthi [80] consider that 10% of the nodes are moving after a request. Finally, Mtibaa et al. [83] consider both a mobility model with low disconnection rate and a mobility model based on a dataset (Infocom06), where the mobility of the devices is predictable in different communication scenarios.

5.2.3. Combination of Stationary and Mobile Resources. Some works mention a combination of mobile and stationary resources. In the edge level, Aazam and Huh [58] consider different types of devices (stationary or mobile). However, the devices are actually not mobile in their simulations.

Borgia et al. [79] consider the local cloud (i.e., the edge level) as mobile and the global cloud as stationary. They use the Random Way Point Model for mobility. Similarly, Qi et al. [82] have mobile end devices and stationary infrastructure servers and describe their own mobility model. Meng et al. [46] use a mobile vehicular cloud together with a stationary local cloud at the edge level and a stationary remote cloud. The mobility of the vehicles is modeled as a Poisson process.

5.3. Summary of Edge Resource Location. Table 4 reveals the distribution of the papers among the above categories and clearly shows that fewer works are multilevel, and the majority are stationary. As noted before, few works are studying managed resources located at different levels and/or mobile.

Note that this does not mean that the works do not consider mobility at all; it only means that the mobility is not on the supply side. An example of the works including mobility on the demand side only is the paper by Plachy et al. [68] who consider that computational resources needed by a user are allocated in a stationary base station in a VM, which can be transferred to another base station if the user is moving. Similar solutions are presented by Tärneberg et al. [67], Gomes et al. [13], Oueis [63], Fan et al. [62], and Wang et al. [77].

Despite demand side node mobility that may be present in all architectures, the supply side node mobility, that is, the notion of mobile managed resource, is among the promises of what the edge brings. We see more mobile resources present in the second and third types of architecture (coordinator device and device cloud). It remains to be seen if the future works will include more 3-level works in which at least two are mobile.

6. Resource Use

The final aspect of resource management considered in our taxonomy is the purpose for which the resource will be used.

6.1. Functional Properties. Edge computing is promoted as a means of getting access to a given service in most of the surveyed articles, that is, for satisfying functionality in an application. There are numerous articles in the literature providing an overview of edge applications, including [6, 7, 10, 23, 27, 34, 41, 89]. Such applications range over augmented reality, connected vehicles, disaster recovery, and a lot of others.

When looking at the different applications used in the surveyed articles presented in the earlier sections, the first finding is that the majority of them (66%) do not consider a specific application in their study. Instead, they refer to generic applications such as IoT services [60], real-time applications [68], and latency-sensitive applications [59] or name some applications but only as an illustration.

Table 5 presents the remaining papers according to which type of application they consider. We can distinguish seven areas in which the described applications can be categorized. Note that in the Generic category we place papers that although not fixed towards one domain of application refer specifically to classes of applications that they exemplify clearly.

6.2. Nonfunctional Properties. In addition to enabling functionalities when using the edge computing paradigm, the very organization of the edge architecture and realizing desirable properties require some kind of resource management too. This additional work is not directly related to the service to obtain; that is, it is a nonfunctional property (also referred to as extrafunctional properties). Obviously, papers that are focusing on a functional property can also be interested in some nonfunctional property.

This subsection is related to the categories of objectives for resource management we have already discussed in Section 4. Achieving the objectives in that section was evaluated using metrics that are often representative for measuring nonfunctional properties.

Examples of metrics and their related nonfunctional property which are encountered more often are

(i) *response time* as a measure of *timeliness*

(ii) *energy consumption* as a measure of *energy efficiency*

(iii) *admission ratio*, or its equivalent blocking probability, as a measure of *availability* of the edge service

(iv) *CPU/network utilization* as a measure of *computation/communication resource efficiency*

(v) *monetary cost* paid to an infrastructure owner as a measure of *cost efficiency*

The list of metrics is not exhaustive, but we have focused on the more prevalent ones. Figure 4 shows how popular the above metrics are in the context of the works studied so far.

Table 5: Applications considered in the surveyed articles.

Area	Applications	Articles
Healthcare	Medical cyber-physical systems	[66]
	Connected health	[71]
Video	Video analytics	[73, 76]
	Video surveillance	[71, 72, 75]
	Video on demand	[45]
IoT	Crowd-sensing	[61]
	Sense-Process-Actuate application	[50]
Gaming	Electroencephalography (EEG) tractor beam game	[72]
Transportation	Connected vehicles	[46, 51]
Content management	User profiling	[13]
Generic	Computation/communication-intensive	[70]
	Delay-sensitive/Delay-tolerant	[81]

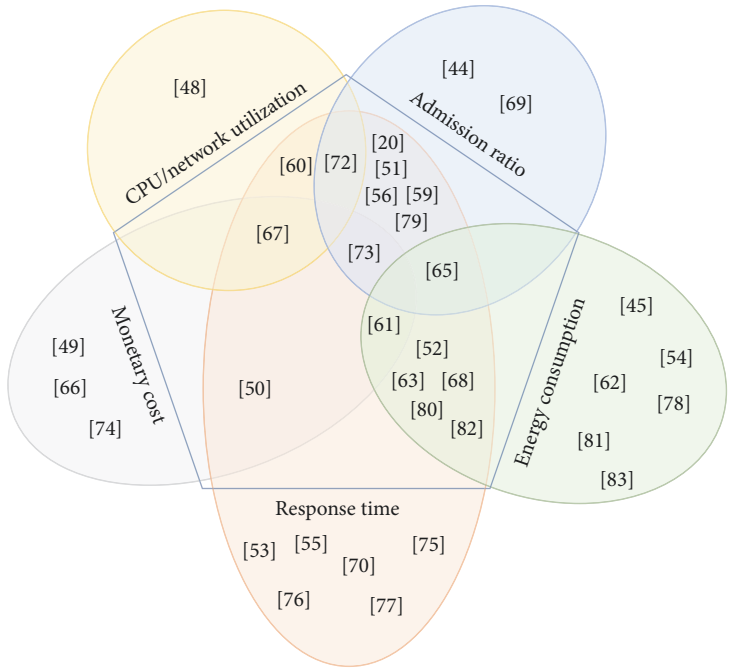

Figure 4: Generic metrics related to a nonfunctional property used in the surveyed articles.

It is not surprising that those metrics that relate to timeliness or availability or resource efficiency are well represented.

As we have noticed earlier, the same paper can deal with multiple resources, multiple objectives, and, also clearly seen in this figure, multiple nonfunctional properties. This illustrates the complex trade-offs involved when dealing with resources in a multistakeholder distributed system.

7. Research Challenges

In this section, we present the research challenges not substantially addressed which could be of interest for further research in the field.

From the previous sections, we noted that the architecture with three active and distinct levels (edge server) is predominant. We also noted that the resource objectives allocation and optimization were well studied. Moreover, computational and communicational resources are the most commonly addressed, typically being stationary and located within a single level. Therefore, research is less prevalent on data, storage, and energy as a resource and less extensive towards the estimation, discovery, and sharing objectives (especially the first two). Furthermore, new works should consider mobility and multilevel locality on the supply side.

Elaborating on mobility, the new phenomenon at the edge is that the supply side can also be mobile and not only the demand side as it was the case in classic clouds. Indeed, edge systems will have to deal with a greater variety of mobility with end devices that are often mobile (like vehicles) but can also be stationary (e.g., video-surveillance

cameras), as well as mobile edge devices. It is, however, not obvious that the mobility patterns of all those devices will be similar, especially between end and edge devices. Considering the large variety of edge applications, their characteristics can potentially vary greatly. For example, an edge solution intended to serve networks of cars moving on a road network will probably be quite different from an edge solution intended to serve persons within a shopping mall. Hence, it is critical to have efficient and thus tailored solutions. But should each application domain rediscover the wheel? Obviously, there is going to be generic wisdom that is transferable across the domains if adequate characterizations of resource requirement patterns are formulated. More work is needed on collecting mobility traces from the different edge applications to see if present patterns can in a generic way be used to create pertinent edge mobility models at both levels of the architecture, the end and the edge level. These can then become a basis for repeatable evaluations of resource management strategies.

Another aspect that will be critical to solving is collaboration. There are new papers appearing where multiple operators at the edge level are modeled, and this introduces new challenges. At the end level, we have seen that different incentives can be provided to enable resource sharing [64, 81] and similarly at the edge level [74]. Such collaboration is especially good for managing workload churns and is interesting for infrastructure owners. The next challenge would be to do multilevel collaboration with a hierarchy of incentive schemes at different levels assuring that they do not cancel out each other's benefits. Moreover, finding more advanced incentive schemes that take both resource efficiency and security into account is needed. Current solutions either choose not to collaborate for security- or privacy-sensitive tasks [44] or rely on classic trust establishment [74] but this will not be enough for a wide collaboration at the edge.

Context adaptation is also one of the properties expected from edge computing and is advocated as a good reason to choose this paradigm [90]. Providing tailored service depending on the user's physical location of course has to be taken care of at the application level. However, it also impacts resource management as those applications will require resources to provide those services, in particular considering data (about supply mobility and abundance) as a resource.

Security, and its subcomponents availability, confidentiality, and integrity, is a key point for edge computing, together with privacy with respect to sensitive end-user data. Although similar, security and privacy have distinct characteristics and should be addressed in depth and separately, which is not the case in the current surveyed works [44, 74]. Regarding availability, most of the works considered focus on admission ratio but do not consider the fact that resources could disappear while executing due to mobility, misuses, or attacks. A notable exception is the work by Habak et al. [20] who propose and evaluate a task checkpointing mechanism that performs result replication to mitigate in case a device disappears. Focusing on availability, several works always consider that the cloud is available as a last fall-back for providing an edge service. If this is not the case (e.g., due to overload, attack, or natural disaster), the availability of the edge service will be impacted. More works in those directions and quantifying edge-specific availability metrics are required. Edge computing will most certainly be interesting for critical infrastructure because of its benefits and those require high standards on security. Research in this direction can be found for the mobile cloud paradigm, for example [91, 92], but they consider scenarios where the edge level is absent.

End-to-end timeliness requires quantification of latencies from an end device towards the cloud (or somewhere at the edge) all the way back to the same device (or to another device). This means traversing the edge networking services, including what we referred to as resource management services in this paper. Since estimation, discovery, sharing, and allocation (including migration) are complex algorithms in such networks, these must also be evaluated in terms of their own resource footprint and thereby their own impact on timeliness and QoS. In the surveyed articles, computing time of the solution is only evaluated by Gomes et al. [13] and Skarlat et al. [50]. Since edge computing cannot become widely used without strong security and privacy properties, it is especially important to research on the resource overhead for providing those properties as well. Too high an overhead can signify a technology that is not feasible in practice.

As shown in Section 3, resources managed at the edge are most often a combination of different resource types. This implies that there will be some interrelations among resource utilization levels, which can create new challenges. Considering resource affinity as in the work by Yousaf and Taleb [48] may be a start but more research is needed to understand and address the complexity of such multiresource problems in the edge context.

As mentioned in Section 2, edge computing brings together diverse business sectors with their existing techniques for solving relevant problems in those areas. Techniques previously applied in only one of those domains may be applicable to edge computing with the required adaptations. For example, performing resource migration requires efficient techniques for this purpose. Ma et al. [93] study container migration and found that the hand-off time decreased by 56% to 80% in comparison to state-of-the-art VM migration for the edge. Results like this should be exploited in the new edge era and utilize technologies that may bring added benefit to edge computing.

Another enabler for resource-efficient edge computing is the development of tools for testing the new proposals in relevant conditions and setups. In the surveyed articles, the most common method used for validating a model or a proposed algorithm is to use an analytical tool (e.g., a solver and/or an optimization engine). Another common approach is to use a simulator, either a generic network simulator such as OMNeT++ (https://omnetpp.org/) or one designed for regular cloud environments such as CloudSim [94], most often with some custom extensions. There also exists a dedicated simulator designed for fog computing, called iFogSim [95], which extends CloudSim, but this one currently has limitations, for example, no mechanisms for offloading or communication between two nodes at the same

level. A third way of evaluating in the surveyed works is the use of physical testbeds. Such evaluations provide invaluable insights into problems that are easy to oversee in simulation and investigate their impact. However, a big challenge for testbeds is to get them to scale, which is to some extent also a problem for simulations. Therefore, there is a need for creating open research testbeds and simulation tools so that configurable architectures and application/domain-specific edge computing methods can be efficiently compared. Coming back to a previous point, such tools should be able to handle mobility of end and edge devices and should obviously be scalable for evaluation of real-world scenarios.

8. Conclusion

The past decade has created tremendous expectations on IoT changing the landscape of data-driven services with benefits for multiple societal sectors. Many researchers have contributed to the development of technologies and addressed challenges that come with resource scarcity in the end devices. Other researchers with a background in cloud computing have looked at how to carry the data generated by the massive IoT deployments and how to efficiently use the cloud resources. The area of edge computing brings these two ends of the same service together in an emerging ecosystem and creates a means to discuss resource adequacy from an end-to-end perspective. In this paper, we have tried to provide an overview, not from a cloud perspective or an IoT device perspective, but with a focus on edge resource management.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Swedish National Graduate School in Computer Science (CUGS).

References

[1] Ericsson, *Ericsson Mobility Report*, 2017, https://www.ericsson.com/assets/local/mobility-report/documents/2017/ericsson-mobility-report-june-2017.pdf.

[2] M. Satyanarayanan, P. Simoens, Y. Xiao et al., "Edge analytics in the internet of things," *IEEE Pervasive Computing*, vol. 14, no. 2, pp. 24–31, 2015.

[3] M. Satyanarayanan, "The emergence of edge computing," *The Computer Journal*, vol. 50, no. 1, pp. 30–39, 2017.

[4] A. Mehta, W. Tärneberg, C. Klein, J. Tordsson, M. Kihl, and E. Elmroth, "How Beneficial Are Intermediate Layer Data Centers in Mobile Edge Networks?" in *Proceedings of the IEEE Workshops on Foundations and Applications of Self∗ Systems (FAS∗W)*, pp. 222–229, Augsburg, Germany, September 2016.

[5] M. Etemad, M. Aazam, and M. St-Hilaire, "Using DEVS for modeling and simulating a Fog Computing environment," in *Proceedings of the 2017 International Conference on Computing, Networking and Communications, (ICNC)*, pp. 849–854, Santa Clara, Calif, USA, January 2017.

[6] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: a survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.

[7] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the 1st ACM Mobile Cloud Computing Workshop, MCC '12*, pp. 13–16, ACM, Helsinki, Finland, August 2012.

[8] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog et al.: a survey and analysis of security threats and challenges," *Future Generation Computer Systems*, vol. 78, no. Part 2, pp. 680–698, 2018.

[9] S. H. Mortazavi, M. Salehe, C. S. Gomes, C. Phillips, and E. de Lara, "Cloudpath: a multi-tier cloud computing framework," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, vol. 13, pp. 1–20, ACM, San Jose, Calif, USA, October 2017.

[10] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. de Foy, and Y. Zhang, "Mobile Edge Cloud System: Architectures, Challenges, and Approaches," *IEEE Systems Journal*, pp. 1–14, 2017.

[11] W. Tärneberg, A. Mehta, J. Tordsson, M. Kihl, E. Elmroth, and W. Tärneberg, "Resource management challenges for the infinite cloud," in *Proceedings of the 10th International Workshop on Feedback Computing at CPSWeek*, Lund University, Seattle, Wash, USA, 2015.

[12] T. Taleb and A. Ksentini, "Follow Me cloud: interworking federated clouds and distributed mobile networks," *IEEE Network*, vol. 27, no. 5, pp. 12–19, 2013.

[13] A. S. Gomes, B. Sousa, D. Palma et al., "Edge caching with mobility prediction in virtualized LTE mobile networks," *Future Generation Computer Systems*, vol. 70, pp. 148–162, 2017.

[14] V. Xhagjika, L. Navarro, and V. Vlassov, "Enhancing real-time applications by means of multi-tier cloud federations," in *Proceedings of the 7th IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 397–404, IEEE, Vancouver, Canada, December 2015.

[15] F. Lobillo, Z. Becvar, M. A. Puente et al., "An architecture for mobile computation offloading on cloud-enabled LTE small cells," in *Proceedings of the 2014 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 1–6, IEEE, Istanbul, Turkey, April 2014.

[16] K. Wang, A. Banerjee, M. Shen, J. Van der Merwe, J. Cho, and K. Webb, "MobiScud: A fast moving personal cloud in the mobile network," in *Proceedings of the 5th Workshop on All Things Cellular: Operations, Applications and Challenges*, pp. 19–24, ACM, London, UK, 2015.

[17] J. Liu, T. Zhao, S. Zhou, Y. Cheng, and Z. Niu, "Concert: A cloud-based architecture for next-generation cellular systems," *IEEE Wireless Communications Magazine*, vol. 21, no. 6, article no. A9, pp. 14–22, 2014.

[18] P. T. Endo, A. V. De Almeida Palhares, N. N. Pereira et al., "Resource allocation for distributed cloud: Concepts and research challenges," *IEEE Network*, vol. 25, no. 4, pp. 42–46, 2011.

[19] K. Habak, M. Ammar, K. A. Harras, and E. Zegura, "Femto clouds: leveraging mobile devices to provide cloud service at the edge," in *Proceedings of the 8th IEEE International Conference on Cloud Computing*, pp. 9–16, IEEE, New York, NY, USA, July 2015.

[20] K. Habak, E. W. Zegura, M. Ammar, and K. A. Harras, "Workload management for dynamic mobile device clusters

in edge femtoclouds," in *Proceedings of the Second ACM/IEEE Symposium on Edge Computing*, pp. 1–14, ACM, San Jose, Calif, USA, October 2017.

[21] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *Proceedings of the 10th International Conference on Intelligent Systems and Control (ISCO)*, pp. 1–8, IEEE, Coimbatore, India, January 2016.

[22] I. Stojmenovic, "Fog computing: A cloud to the ground support for smart things and machine-to-machine networks," in *Proceedings of the 2014 Australasian Telecommunication Networks and Applications Conference (ATNAC)*, pp. 117–122, Southbank, Australia, November 2014.

[23] M. Chiang and T. Zhang, "Fog and IoT: an overview of research opportunities," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 854–864, 2016.

[24] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[25] N. M. Gonzalez, W. A. Goya, R. de Fatima Pereira et al., "Fog computing: Data analytics and cloud distributed processing on the network edges," in *Proceedings of the 2016 35th International Conference of the Chilean Computer Science Society (SCCC)*, pp. 1–9, Valparaíso, Chile, October 2016.

[26] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: a survey," in *Proceedings of the 10th International Conference on Wireless Algorithms, Systems, and Applications*, vol. 9204 of *Lecture Notes in Computer Science*, pp. 685–695, Springer International Publishing, 2015.

[27] A. C. Baktir, A. Ozgovde, and C. Ersoy, "How Can Edge Computing Benefit From Software-Defined Networking: A Survey, Use Cases, and Future Directions," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2359–2391, 2017.

[28] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the Workshop on Mobile Big Data (Mobidata '15)*, pp. 37–42, ACM, Hangzhou, China, June 2015.

[29] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog Computing: A Taxonomy, Survey and Future Directions," in *Internet of Everything. Internet of Things (Technology, Communications and Computing)*, pp. 103–130, Springer, Singapore, 2018.

[30] A. Bhattacharya and P. De, "A survey of adaptation techniques in computation offloading," *Journal of Network and Computer Applications*, vol. 78, pp. 97–115, 2017.

[31] F. Rebecchi, M. Dias de Amorim, V. Conan, A. Passarella, R. Bruno, and M. Conti, "Data offloading techniques in cellular networks: a survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 2, pp. 580–603, 2015.

[32] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "Mobile Edge Computing: Survey and Research Outlook," https://arxiv.org/abs/1701.01090, 2017.

[33] P. Mach and Z. Becvar, "Mobile Edge Computing: A Survey on Architecture and Computation Offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[34] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A Survey on Mobile Edge Computing: The Communication Perspective," *IEEE Communications Surveys & Tutorials*, 2017.

[35] L. Chunlin and L. LaYuan, "Cost and energy aware service provisioning for mobile client in cloud computing environment," *The Journal of Supercomputing*, vol. 71, no. 4, pp. 1196–1223, 2015.

[36] S. Shekhar, A. D. Chhokra, A. Bhattacharjee, G. Aupy, and A. Gokhale, "INDICES: exploiting edge resources for performance-aware cloud-hosted services," in *Proceedings of the 1st IEEE International Conference on Fog and Edge Computing (ICFEC)*, pp. 75–80, IEEE, Madrid, Spain, 2017.

[37] K. Toczé and S. Nadjm-Tehrani, "Where Resources Meet at the Edge," in *Proceedings of the 2017 IEEE International Conference on Computer and Information Technology (CIT)*, pp. 302–307, IEEE, Helsinki, Finland, August 2017.

[38] R. Cziva and D. P. Pezaros, "Container Network Functions: Bringing NFV to the Network Edge," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 24–31, 2017.

[39] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A Comprehensive Survey on Fog Computing: State-of-the-art and Research Challenges," *IEEE Communications Surveys & Tutorials*, 2017.

[40] A. Mtibaa, K. A. Harras, K. Habak, M. Ammar, and E. W. Zegura, "Towards mobile opportunistic computing," in *Proceedings of the 8th IEEE International Conference on Cloud Computing*, pp. 1111–1114, IEEE, New York, NY, USA, July 2015.

[41] M. Satyanarayanan, P. Bahl, R. Cáceres, and N. Davies, "The case for VM-based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.

[42] H. Frank, W. Fuhrmann, and B. Ghita, "Mobile Edge Computing: Requirements for powerful mobile near real-time applications," in *Proceedings of the 11th International Network Conference (INC 2016)*, pp. 63–66, Germany, July 2016.

[43] M. Aazam, M. St-Hilaire, C.-H. Lung, and I. Lambadaris, "PRE-Fog: IoT trace based probabilistic resource estimation at Fog," in *Proceedings of the 13th IEEE Annual Consumer Communications and Networking Conference (CCNC 2016)*, pp. 12–17, IEEE, Las Vegas, Nev, USA, January 2016.

[44] A. Singh, N. Auluck, O. Rana, A. Jones, and S. Nepal, "RT-SANE: Real time security aware scheduling on the network edge," in *Proceedings of the10th International Conference on Utility and Cloud Computing*, pp. 131–140, ACM, New York, NY, USA, 2017.

[45] V. Valancius, N. Laoutaris, L. Massoulié, C. Diot, and P. Rodriguez, "Greening the internet with nano data centers," in *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*, pp. 37–48, ACM, Rome, Italy, December 2009.

[46] H. Meng, K. Zheng, P. Chatzimisios, H. Zhao, and L. Ma, "A utility-based resource allocation scheme in cloud-assisted vehicular network architecture," in *Proceedings of the IEEE International Conference on Communication Workshop, ICCW 2015*, pp. 1833–1838, IEEE, London, UK, June 2015.

[47] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 1617–1655, 2016.

[48] F. Z. Yousaf and T. Taleb, "Fine-grained resource-aware virtual network function management for 5G carrier cloud," *IEEE Network*, vol. 30, no. 2, pp. 110–115, 2016.

[49] K. Wang, K. Yang, X. Wang, and C. S. Magurawalage, "Cost-effective resource allocation in C-RAN with mobile cloud," in *Proceedings of the IEEE International Conference on Communications, ICC 2016*, IEEE, Kuala Lumpur, Malaysia, May 2016.

[50] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards QoS-Aware Fog Service Placement," in *Proceedings of the 1st IEEE International Conference on Fog and Edge Computing, ICFEC 2017*, pp. 89–96, IEEE, Madrid, Spain, 2017.

[51] H. R. Arkian, R. E. Atani, and A. Pourkhalili, "A cluster-based vehicular cloud architecture with learning-based resource management," in *Proceedings of the 2014 6th IEEE International Conference on Cloud Computing Technology and Science, CloudCom 2014*, pp. 162–167, IEEE, Singapore, December 2014.

[52] T. Nishio, R. Shinkuma, T. Takahashi, and N. B. Mandayam, "Service-oriented heterogeneous resource sharing for optimizing service latency in mobile cloud," in *Proceedings of the First International Workshop on Mobile Cloud Computing & Networking*, pp. 19–26, ACM, New York, NY, USA, 2013.

[53] D. Mascitti, M. Conti, A. Passarella, and L. Ricci, "Service Provisioning through Opportunistic Computing in Mobile Clouds," *Procedia Computer Science*, vol. 40, pp. 143–150, 2014.

[54] W. Liu, T. Nishio, R. Shinkuma, and T. Takahashi, "Adaptive resource discovery in mobile cloud computing," *Computer Communications*, vol. 50, pp. 119–129, 2014.

[55] W. Liu, R. Shinkuma, and T. Takahashi, "Opportunistic resource sharing in mobile cloud computing: The single-copy case," in *Proceedings of the 16th Asia-Pacific Network Operations and Management Symposium, APNOMS 2014*, pp. 1–6, IEEE, Hsinchu, Taiwan, September 2014.

[56] T. Penner, A. Johnson, B. Van Slyke, M. Guirguis, and Q. Gu, "Transient clouds: Assignment and collaborative execution of tasks on mobile devices," in *Proceedings of the 2014 IEEE Global Communications Conference, GLOBECOM 2014*, pp. 2801–2806, IEEE, Austin, Tex, USA, December 2014.

[57] Z. Hao, E. Novak, S. Yi, and Q. Li, "Challenges and Software Architecture for Fog Computing," *IEEE Internet Computing*, vol. 21, no. 2, pp. 44–53, 2017.

[58] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT," in *Proceedings of the IEEE 29th International Conference on Advanced Information Networking and Applications (AINA '15)*, pp. 687–694, IEEE, Gwangiu, South Korea, March 2015.

[59] Y. Liu, M. J. Lee, and Y. Zheng, "Adaptive Multi-Resource Allocation for Cloudlet-Based Mobile Cloud Computing System," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2398–2410, 2016.

[60] B. Confais, A. Lebre, and B. Parrein, "An Object Store Service for a Fog/Edge Computing Infrastructure Based on IPFS and a Scale-Out NAS," in *Proceedings of the 1st IEEE International Conference on Fog and Edge Computing, ICFEC 2017*, pp. 41–50, IEEE, Madrid, Spain, 2017.

[61] H. R. Arkian, A. Diyanat, and A. Pourkhalili, "MIST: Fog-based data analytics scheme with cost-efficient resource provisioning for IoT crowdsensing applications," *Journal of Network and Computer Applications*, vol. 82, pp. 152–165, 2017.

[62] Q. Fan, N. Ansari, and X. Sun, "Energy Driven Avatar Migration in Green Cloudlet Networks," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1601–1604, 2017.

[63] J. Oueis, *Joint Communication and Computation Resources Allocation for Cloud-Empowered Future Wireless Networks [Ph.D. thesis]*, Université Grenoble, 2016.

[64] L. Tang, S. He, and Q. Li, "Double-Sided Bidding Mechanism for Resource Sharing in Mobile Cloud," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1798–1809, 2017.

[65] P. Borylo, A. Lason, J. Rzasa, A. Szymanski, and A. Jajszczyk, "Energy-aware fog and cloud interplay supported by wide area software defined networking," in *Proceedings of the 2016 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, Kuala Lumpur, Malaysia, May 2016.

[66] L. Gu, D. Zeng, S. Guo, A. Barnawi, and Y. Xiang, "Cost efficient resource management in fog computing supported medical cyber-physical system," *IEEE Transactions on Emerging Topics in Computing*, vol. 5, no. 1, pp. 108–119, 2017.

[67] W. Tärneberg, A. Mehta, E. Wadbro et al., "Dynamic application placement in the Mobile Cloud Network," *Future Generation Computer Systems*, vol. 70, pp. 163–177, 2017.

[68] J. Plachy, Z. Becvar, and E. C. Strinati, "Dynamic resource allocation exploiting mobility prediction in mobile edge computing," in *Proceedings of the 27th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, IEEE, Valencia, Spain, September 2016.

[69] C. Fricker, F. Guillemin, P. Robert, and G. Thompson, "Analysis of an Offloading Scheme for Data Centers in the Framework of Fog Computing," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 1, no. 4, pp. 1–18, 2016.

[70] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," *Institute of Electrical and Electronics Engineers. Transactions on Computers*, vol. 66, no. 5, pp. 810–819, 2017.

[71] Q. Zhang, X. Zhang, Q. Zhang, W. Shi, and H. Zhong, "Firework: Big data sharing and processing in collaborative edge environment," in *Proceedings of the 4th IEEE Workshop on Hot Topics in Web Systems and Technologies, HotWeb 2016*, pp. 20–25, Washington, DC, USA, October 2016.

[72] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-Aware Application Scheduling in Fog Computing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, 2017.

[73] A. R. Zamani, M. Zou, J. Diaz-Montes et al., "Deadline Constrained Video Analysis via In-Transit Computational Environments," *IEEE Transactions on Services Computing*, pp. 1-1, 2017.

[74] L. Chen and J. Xu, "Socially trusted collaborative edge computing in ultra dense networks," in *Proceedings of the the Second ACM/IEEE Symposium on Edge Computing*, pp. 1–11, ACM, San Jose, Calif, USA, October 2017.

[75] J. Wang, J. Pan, and F. Esposito, "Elastic urban video surveillance system using edge computing," in *Proceedings of the the Workshop on Smart Internet of Things*, pp. 1–6, ACM, San Jose, Calif, USA, October 2017.

[76] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, and Q. Li, "Lavea: Latency-aware video analytics on edge computing platform," in *Proceedings of the the Second ACM/IEEE Symposium on Edge Computing*, pp. 1–13, ACM, San Jose, Calif, USA, October 2017.

[77] S. Wang, R. Urgaonkar, T. He, K. Chan, M. Zafer, and K. K. Leung, "Dynamic Service Placement for Mobile Micro-Clouds with Predicted Future Costs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 4, pp. 1002–1016, 2017.

[78] S. Sardellitti, G. Scutari, and S. Barbarossa, "Joint optimization of radio and computational resources for multicell mobile-edge computing," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 1, no. 2, pp. 89–103, 2015.

[79] E. Borgia, R. Bruno, M. Conti, D. Mascitti, and A. Passarella, "Mobile edge clouds for Information-Centric IoT services," in *Proceedings of the 2016 IEEE Symposium on Computers and Communication, ISCC 2016*, pp. 422–428, IEEE, Messina, Italy, July 2016.

[80] P. Athwani and D. P. Vidyarthi, "Resource discovery in mobile cloud computing: A clustering based approach," in *Proceedings*

of the IEEE UP Section Conference on Electrical Computer and Electronics, UPCON 2015, IEEE, Allahabad, India, December 2015.

[81] A. P. Bianzino, J. Rougier, C. Chaudet, and D. Rossi, "The Green-Game: Accounting for Device Criticality in Resource Consolidation for Backbone IP Networks," *Strategic Behavior and the Environment*, vol. 4, no. 2, pp. 131–153, 2014.

[82] Q. Qi, J. Liao, J. Wang, Q. Li, and Y. Cao, "Dynamic resource orchestration for multi-task application in heterogeneous mobile cloud computing," in *Proceedings of the 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pp. 221–226, IEEE, San Francisco, Calif, USA, April 2016.

[83] A. Mtibaa, A. Fahim, K. A. Harras, and M. H. Ammar, "Towards resource sharing in mobile device clouds: Power balancing across mobile devices," in *Proceedings of the 2013 2nd ACM SIGCOMM Workshop on Mobile Cloud Computing, MCC 2013*, pp. 51–56, ACM, Hong Kong, China, August 2013.

[84] E. J. Vergara, S. Nadjm-Tehrani, and M. Asplund, "Fairness and Incentive Considerations in Energy Apportionment Policies," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 2, no. 1, pp. 1–29, 2016.

[85] Z. Xiao, W. Song, and Q. Chen, "Dynamic resource allocation using virtual machines for cloud computing environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1107–1117, 2013.

[86] Z. Dong, Y. Liu, H. Zhou et al., "An energy-efficient offloading framework with predictable temporal correctness," in *Proceedings of the the Second ACM/IEEE Symposium on Edge Computing*, pp. 1–12, ACM, San Jose, Calif, USA, October 2017.

[87] R. Yu, X. Huang, J. Kang et al., "Cooperative resource management in cloud-enabled vehicular networks," *IEEE Transactions on Industrial Electronics*, vol. 62, no. 12, pp. 7938–7951, 2015.

[88] N. Wang, B. Varghese, M. Matthaiou, and D. S. Nikolopoulos, "ENORM: a framework for edge node resource management," *IEEE Transactions on Services Computing*, 2017.

[89] M. Satyanarayanan, G. Lewis, E. Morris, S. Simanta, J. Boleng, and K. Ha, "The role of cloudlets in hostile environments," *IEEE Pervasive Computing*, vol. 12, no. 4, pp. 40–49, 2013.

[90] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, and R. Buyya, "A Context Sensitive Offloading Scheme for Mobile Cloud Computing Service," in *Proceedings of the 8th IEEE International Conference on Cloud Computing*, pp. 869–876, IEEE, New York, NY, USA, July 2015.

[91] Y. Liu and M. J. Lee, "Security-Aware Resource Allocation for Mobile Cloud Computing Systems," in *Proceedings of the 2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pp. 1–8, IEEE, Las Vegas, Nev, USA, August 2015.

[92] H. Liang, D. Huang, L. X. Cai, X. Shen, and D. Peng, "Resource allocation for security services in mobile cloud computing," in *Proceedings of the 2011 IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS 2011*, pp. 191–195, IEEE, Shangha, China, April 2011.

[93] L. Ma, S. Yi, and Q. Li, "Efficient service handoff across edge servers via docker container migration," in *Proceedings of the the Second ACM/IEEE Symposium on Edge Computing*, pp. 1–13, ACM, San Jose, Calif, USA, October 2017.

[94] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. de Rose, and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, 2011.

[95] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.