IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received September 12, 2019, accepted October 7, 2019, date of publication October 21, 2019, date of current version October 31, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2948399

What the Fog? Edge Computing Revisited: Promises, Applications and Future Challenges

JULIEN GEDEON, FLORIAN BRANDHERM, ROLF EGERT, TIM GRUBE, AND MAX MÜHLHÄUSER

Department of Computer Science, Technische Universität Darmstadt, 64289 Darmstadt, Germany

Corresponding author: Julien Gedeon (gedeon@tk.tu-darmstadt.de)

This work was supported in part by the German Research Foundation (DFG) and the National Nature Science Foundation of China (NSFC) joint project under Grant 392046569 (DFG) and Grant 61761136014 (NSFC), as part of the Collaborative Research Center 1053–MAKI (DFG) and as part of the project D.4 within the RTG 2050 "Privacy and Trust for Mobile Users" (DFG), and in part by the Open Access Publishing Fund of Technische Universität Darmstadt.

ABSTRACT Edge computing brings computing and storage resources closer to (mobile) end users and data sources, thus bypassing expensive and slow links to distant cloud computing infrastructures. Often leveraged opportunistically, these heterogeneous resources can be used to offload data and computations, enabling upcoming demanding applications such as augmented reality and autonomous driving. Research in this direction has addressed various challenges, from architectural concerns to runtime optimizations. As of today, however, we lack a widespread availability of edge computing—partly because it remains unclear which of the promised benefits of edge computing are relevant for what types of applications. This article provides a comprehensive snapshot of the current edge computing landscape, with a focus on the application perspective. We outline the characteristics of edge computing and its postulated benefits and drawbacks. To understand the functional composition of applications, we first define common application components that are relevant w.r.t. edge computing. We then present a classification of proposed use cases and analyze them according to their expected benefits from edge computing and which components they use. Furthermore, we illustrate existing products and industry solutions that have recently surfaced and outline future research challenges.

INDEX TERMS Edge computing, heterogeneous networks, next generation networking mobile applications, Internet of Things, ubiquitous computing.

I. INTRODUCTION

Edge computing has recently gained tremendous attention in both academia and industry. This new paradigm aims to place resources for storage and computation closer to end users, i.e., to the *edge* of the network. The main motivation to do so are the shortcomings of today's cloud computing infrastructures when processing large-volume data for latency-critical applications. Cloud computing—while offering vast resources that can be flexibly used to fit changing demand—often leads to substantial latencies when using services placed at distant locations [1]. At the same time, wide-area network bandwidth remains a scarce resource [2] and thus, transferring large streams of data to centralized cloud resources is costly.

Until recently, these drawbacks of cloud computing had only a limited impact since most data was both produced and consumed in the cloud. Examples of such applications are big data processing and data warehousing. The producers and consumers of the data in many of those use cases are well-connected fat clients, many of which are located in data centers themselves. This centralization of data has decreased tremendously in the past years. First, more and more people own so-called personal smart devices that are connected to the Internet, such as smartphones or smartwatches. It is predicted that these devices will be complemented by smart glasses and various on-body sensors in the near future. Second, we can observe an increase in small-scale sensors and actuators that

The associate editor coordinating the review of this manuscript and approving it for publication was Mohamed Elhoseny.

capture large amounts of data to make local decisions. The *Internet of Things* (IoT) is a prominent recent buzzword that captures this idea. The number of connected devices in both categories is expected to increase dramatically, supported by Cisco's recent forecasts that their number will reach 28.5 billion by 2022[1] and 500 billion by 2030[2].

All these devices collect huge amounts of data. In many cases, this data has the following characteristics: (i) it is only locally relevant, (ii) it requires further analysis and processing, (iii) the result of processing and subsequent actuation is subject to stringent latency requirements, (iv) the raw data is ephemeral, i.e., it is no longer relevant after it has been processed and can therefore be discarded or moved to persistent storage. Prominent examples that fall into these categories are real-time video analytics [3], cognitive assistance applications [4], mobile gaming [5], and autonomous driving [6].

Even though mobile devices today are equipped with powerful hardware, the hardware is still inadequate for many demanding tasks like video analytics or high-quality 3D graphics. More importantly, form factors and battery life remain limiting factors in the design of mobile devices and hence, carrying out demanding battery-draining tasks on mobile devices remains prohibitive. To circumvent this issue, mobile devices can save battery life by offloading such tasks to the cloud. However, real-time applications are often not feasible for cloud computing due to disadvantages such as high latency. This mismatch—the demand to offload demanding processing tasks, but close to the mobile device—has led to the emergence of *edge computing* [7]–[9]. Edge computing avoids the costly transfer of latency- and bandwidth-critical data to the cloud and uses heterogeneous, nearby resources for storage [10] and computation [11]. Small data centers, often accessed via 1-hop wireless connections, have been termed *cloudlets* [12], [13] and are a crucial enabler for edge computing.

It is worth noting that other terminology has been used to denote the same general concept of placing resources close to end devices; the most notable term being *fog computing* [14]–[16]. We will discuss the issue of different—and sometimes conflicting—terminology in Section II in more detail.

While a large number of publications outline potential use cases and individual building blocks for edge computing—from hardware infrastructure to programming paradigms—it often remains unclear which benefits are relevant for which type of applications. Many works present a single use case to assess a specific problem that is related to edge computing (e.g., the placement of computing tasks), but only evaluate it with respect to a single criterion, e.g., latency or costs and disregard other aspects. Hence, overall it remains unclear which types of applications benefit from which particular aspect of edge computing. This also raises the question if and how applications can be characterized, which could lead to more insights about what distinguishes a killer application for edge computing. This is an interesting question considering the observation that it is possible to identify basic application building blocks that are shared across application domains. For example, understanding the contents of a scene through computer vision is useful for both mobile gaming and for detecting traffic emergencies.

This article is intended to answer the above questions. We aim to provide the novice reader with a comprehensive overview of the proposed use cases of edge computing. For the advanced reader, we aim to give a new perspective on how to classify edge computing applications according to basic shared components and in what way applications benefit from edge computing.

We begin by defining edge computing for the purposes of this paper and outline its main characteristics. In particular, we outline the heterogeneity of edge computing and its enabling technologies. Furthermore, we summarize the benefits and drawbacks of edge computing. For our analysis of edge computing use cases, we define four basic components that applications use: (i) data consolidation, (ii) filtering and pre-processing, (iii) data storage and retrieval, and (iv) computation offloading. We then survey potential use cases for edge computing, classify them into four general categories and analyze how well the benefits of edge computing support representative applications. This analysis is crucial to truly understand where the future potential of edge computing lies and what the remaining challenges are.

Based on the observation that edge computing services are not yet widely available, we analyze existing products and commercial solutions and identify future research challenges that need to be addressed in order to fulfill the vision of computation as a ubiquitous utility.

A. CONTRIBUTIONS

In summary, this article makes the following contributions:

- **Definition & understanding.** We refine the definition and understanding of edge computing, systematizing use cases, potential benefits and common application components.
- **Application survey.** We provide an extensive overview of proposed applications and systematically examine how and to what extent they can benefit from edge computing.
- **Current state in industry.** Albeit not widespread today, some existing products targeted toward edge computing exist. We summarize the industry landscape and describe representative examples.
- **Future research directions.** To engage the readers in further discussions and incentivize upcoming work in the field, we suggest future research directions to bring edge computing to its full potential.

[1] https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white-paper-c11-741490.pdf (Accessed: 2019-09-02)

[2] https://www.cisco.com/c/dam/en/us/products/collateral/se/internet-of-things/at-a-glance-c45-731471.pdf (Accessed: 2019-09-02)

B. RELATED SURVEYS

Given both the timeliness and the broadness of the topic, other surveys with different foci exist. Yousefpour *et al.* [17] have reviewed a vast number of publications in the domain of edge and fog computing. Pang *et al.* [18] have compared the cloudlet approach to cloud computing with five application scenarios. Yi *et al.* [19] have defined and reviewed the concept of fog computing, including future issues. Similarly, Mahmud *et al.* [20] have presented a taxonomy and recent developments in fog computing. Our taxonomy in Section II gives a broader overview without restricting the analysis to fog computing or edge computing alone. Hong and Varghese [21] have focused on resource management in edge and fog computing. Other surveys shed light on edge computing from the networking [22] or security [23]–[25] perspective. For the first case, Baktir *et al.* [26] have detailed how Software Defined Networking can be an important enabler for edge computing.

Li *et al.* [27] have surveyed the edge computing paradigm with a focus on the architecture and management issues. They classify related work according to key design objectives; however, the authors do not provide an in-depth analysis regarding the benefits of edge computing for particular classes of applications. Many works argue about the benefits of edge computing only from the viewpoint of one class of applications, e.g., IoT [28]–[30], or smart city applications [31], [32]. Xiong *et al.* [33] have used edge computing for mobile blockchain applications. In contrast, we aim to provide a survey of use cases across application domains. Puliafito *et al.* [34] have surveyed a variety of applications for fog computing, however, they miss several compelling use cases, especially related to mobile personal devices (e.g., for gaming or AR/VR applications). Furthermore, the benefits of using edge computing are not quantified.

A number of surveys [35]–[37] focus on Mobile Edge Computing (MEC). MEC refers to the colocation of resources at the Radio Access Network (RAN), e.g., at cellular base stations. This particular variant of edge computing is expected to grow as new cellular standards such as 5G will become available [38], [39]. Beck *et al.* [35] have presented a taxonomy of mobile edge computing and detailed several use cases; however, they miss compelling application scenarios like AR/VR and smart city use cases. The authors have also discussed the implications for different stakeholders and emphasized that the main motivation behind MEC is to reduce network stress. Mao *et al.* [36] have reviewed Mobile Edge Computing from the perspective of the different wireless communication technologies and survey approaches for the joint management of radio and computation resources. Mach and Becvar [40] have focused on the offloading problem in MEC. Yi *et al.* [14] have reviewed the terminology and use cases around fog computing.

Contrary to previous works, our article aims to provide (i) a broad overview of the trend of moving computations closer to the edge, (ii) a comprehensive classification of proposed applications that can make use of edge computing, and (iii) an analysis of what actual benefits can be achieved when using edge computing in the different classes of applications.

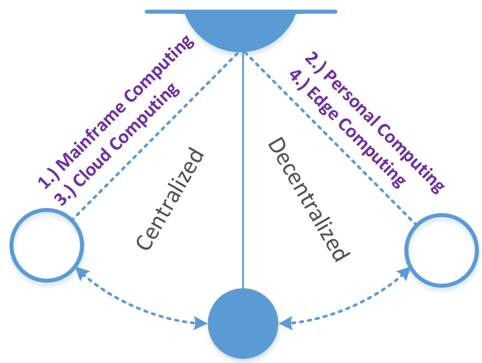

FIGURE 1. Centralized and decentralized computing paradigms.

C. OUTLINE

The remainder of this article is organized as follows. First, we present a taxonomy of edge computing (Section II). In Section III, we elaborate in more detail the characteristics of this new computing paradigm, focusing on its benefits and technological enablers. Section IV describes and classifies the vast amount of applications that have been suggested for edge computing. We especially focus on the question which applications benefit from which aspects of edge computing. To bridge academic research with the commercial reality, Section V gives an overview of existing products. We analyze future research challenges in Section VI before concluding our article in Section VII.

II. A TAXONOMY OF EDGE COMPUTING

Throughout the history of computing, we can observe a constant back-and-forth swinging between centralized and decentralized computing approaches. This general observation is outlined in Figure 1. The first transition toward decentralized computing was the move from centralized mainframes to personal computers. In the mid-2000s, we saw a major disruption with the advent of cloud computing. Cloud computing offers virtualized resources in large data centers. These resources can be used with flexible pricing models, often on a pay-as-you-go basis. Scaling in and out according to current demands can be done at a moment's notice and therefore removes the problem of over- or underprovisioning of resources. We argue that cloud computing belongs to the category of centralized approaches because computing power is concentrated in a few distant locations (compared to the number of clients that use it). Leveraging cloud resources for offloading from mobile devices is termed *Mobile Cloud Computing* (MCC) [41] for which a variety of frameworks exist [42]–[45]. Emerging classes of applications that require fast processing of large data generated from client devices and surrounding sensors have led to the latest distributed computing paradigm, termed *edge computing*.

Edge computing is the concept to place and use storage and computing resources closer to the (mobile) devices that produce and consume the data. The prevalent counterpart to this approach is cloud computing, where said resources are located at data centers. Edge computing is carried out using resources on *edge nodes* or *edge devices*. Similarly, the term *cloudlet* [12] has been coined to denote small-scale data centers close to users. In the context of computation offloading (see Section III-E1), nodes that are leveraged to perform computations are called *surrogates*. *Edge sites* are the physical environments where the edge resources are located. We refer to an *edge system* or *edge framework* to denote the entirety of resources at the edge, their clients, and control entities responsible for managing the resources.

It is important to note that edge computing aims not to be a replacement for cloud computing, but to complement it [46]. This makes sense if we assume that every application needs access to three basic resources: (i) computation, (ii) communication, and (iii) storage. The need for computation and storage resources is well-served by cloud computing; in fact, the reason for the success of cloud computing is its capability to provide elastic resources that instantly scale to customers' needs. On the downside, cloud computing cannot offer any guarantees w.r.t. the communication part because data centers are located away from the consumer and typically, neither the cloud provider nor the user has control over the transit network. Edge computing solves this issue in the sense that it adds scalability in the network dimension, i.e., more users can be served with low-latency links when adding more edge sites (e.g., at users' wireless gateways). However, because of the limited resources at individual edge sites, edge computing cannot offer the same overall elasticity as cloud computing. Furthermore, realizing scalability (w.r.t. any of the three resource dimensions) requires much more complex management in view of the dynamics in the network (e.g., caused by user mobility or sudden local changes in demands).

Therefore, we expect a co-existence of cloud and edge computing. Edge computing can offer the additional scalability that will be required for locally relevant tasks in an environment with a vast amount of data generators and consumers. Complex, long-running and data-driven tasks that are not time-critical will benefit more from the abundance of scalable resources in the cloud. Similarly, edge computing will most likely not be able to replace the cloud for the long-term storage of data because of the limited capabilities of edge devices. However, user-facing time-critical tasks may benefit from a reduced latency in the critical path when using infrastructure at the edge and this might very well include the caching of ephemeral data on edge device.

While the cloud offers virtually unlimited resources in geo-distributed data centers, edge processing power is locally clustered around its consumers. These computing nodes are more heterogeneous w.r.t. to their available resources and are often leveraged opportunistically. Hence, their availability may be limited and they are less reliable than resources in federated data centers. However, edge computing can make the communication more reliable, in the sense that it can offer an alternative if network links to the cloud break down. This is especially interesting for disaster scenarios where edge computing can offer an alternative infrastructure to maintain critical tasks alive. Edge resources can often be accessed within one hop from the wireless gateway that users are connected to. Ideally, edge computing systems can detect and support user mobility, e.g., by migrating their data and computations to the next proximate location. In addition, wireless gateways can provide additional contextual information to the application, something not available in cloud computing.

Another major difference is the granularity of offloading. In cloud computing, we see entire applications being moved to remote resources, while at the edge, offloading is more fine-grained and needs a more careful decision of what to offload. Individually offloaded components at the edge are often part of a processing pipeline consisting of several of those components that not necessarily run on the same edge device. Additionally, we can observe that because of limited resources at the edge and the higher user dynamics, virtualization technologies used for edge computing tend to be more lightweight, e.g., by using containers instead of virtual machines. Another important characteristic of edge computing is the loose coupling between clients and the infrastructure. Hence, in our opinion, static on-premise deployments do not classify as edge computing, when those are only linked to one specific application and do not face the challenges of dynamic environments (e.g., with respect to network changes, device and data mobility or scaling). Table 1 summarizes the differences between edge and cloud computing.

In the remainder of this article, we will use the term *edge computing* to denote the general concept of placing resources close to users, sensors, or actuators. Other terminology that denotes similar concepts exist, most notably the term *fog computing* [14]–[16]. Fog computing is a term originally coined by Cisco [15] in the context of their IOx platform, envisioning to leverage untapped processing power in network middleboxes when those are either overprovisioned or not running at full load.

While the terms edge and fog both allude to the same concept—processing data close to end devices—it is worth noticing that there is a broad spectrum of (sometimes blurry) definitions and arguments in trying to define the differences of the two. One possible distinction is that fog computing extends the cloud toward the edge, while edge computing originates from the need of end devices to offload computations. However, the exact definitions remain an ongoing discussion in academia [47], [48].

Mobile Edge Computing (MEC)—more recently termed Multi-Access Edge Computing—refers to the colocation of resources at the Radio Access Networks (RAN), e.g., at cellular base stations. This can therefore be considered as a special

TABLE 1. Comparison of edge computing with cloud computing.

	Cloud Computing	Edge Computing
Proximity to client devices	low	high
End-to-end latency	high	low
Infrastructure	centralized data centers	decentralized cloudlets
Heterogeneity of computing hardware	low	high
Number of computing resource locations	few	many
Resources at individual locations	many	few
Geo-distribution of computing resources	locally clustered	widespread
Availability & reliability of resources	high	varying
Virtualization	heavyweight	lightweight
Connection to resources	long-thin	short-fat
Access to resources	through core network	typically via 1-hop wireless gateway
Applications	data-driven	user-driven
Offloading granularity	mostly entire applications	computationally intensive and latency-critical parts only

case, mostly from the point of view of mobile network operators. Other hybrid terms exist, notably mist computing [49] and osmotic computing [46]. The former can be thought of being similar to fog but closer to the edge devices, while the latter advocates a seamless migration of services from data centers to the edge.

III. CHARACTERISTICS OF EDGE COMPUTING

In this section, we analyze the characteristics of edge computing. More specifically, we start by analyzing what the potential benefits and drawbacks of edge computing are (Section III-A). We then discuss the characteristics of the edge in terms of communication (Section III-B), the involved devices (Section III-C) and stakeholders (Section III-D). Lastly, we review enabling technologies for edge computing in Section III-E.

A. PROMISES, BENEFITS AND DRAWBACKS OF EDGE COMPUTING

As outlined in the previous section on the taxonomy of edge computing, the general idea is to move storage and processing capabilities from the cloud closer to the clients and to the location of the data's origin, often by opportunistically using the infrastructure in a highly dynamic mobile environment. This potentially brings a number of advantages, the most important of which we describe in the following:

Lower latency: As we will discuss later in this article, many types of applications have stringent requirements on the end-to-end latency, i.e., the overall time from requesting a service (e.g., a computation) to obtaining the result. One important factor on this critical path is the network delay. Cloud computing infrastructures are geographically widely distributed across data centers and the user typically has little to no control over where the requests will be processed. Hence, it is not uncommon for requests to be directed to distant data centers. Many works have presented empirical measurements of network latencies and motivated edge computing based on those numbers [4], [12], [50]. For instance, in [12] the authors measure the mean round-trip times between New York and Berkeley to be 85 ms. If we now imagine an application that needs to process a video scene in near real-time with a delay constraint of less than 50 ms, the mere network latency already violates this constraint. Even by optimizing the transit networks, there are physical lower bounds that cannot be overcome. In contrast, the access delay to a nearby wireless gateway in the case of WiFi is typically in the order of magnitude of a few milliseconds. Chen *et al.* [4] have conducted extensive empirical studies and conclude that using the cloud over a cloudlet adds around 100–200 ms for their use cases which are based on cognitive assistance.

Less bandwidth utilization in the core network: In the current landscape of billions of mobile devices that generate data, we observe that captured data often only is of limited spatial and temporal relevance. As an example, we can imagine an intelligent scheduling scheme for traffic lights that is based on reported sensor data from vehicles [51]. Furthermore, individual sensor readings are rarely of interest. Instead, what is consumed by applications are aggregated values or events derived from the data (e.g., a sensor reading that is outside a normal range). Typically, all raw values would be streamed to the cloud; however, given the increase in data, this might overload the core network. This is especially relevant since wide-area network bandwidth remains a scarce resource [2]. The same holds true for many of today's wireless access networks, e.g., as motivated in [52]. Especially large, continuous data streams can be a burden on backhaul networks. Distributed processing and aggregation of data streams along the path to the consumer can help to mitigate this. In the domain of Wireless Sensor Networks this is a popular approach [53] that can easily be mapped to aggregation by intermediate edge nodes. Besides aggregation, edge computing can also offer storage capabilities [54] that take into account contextual information for the decision on where to store the data [10]. For example, at large-scale events with overloaded mobile networks, edge nodes can provide storage to share data among people that are close-by. Other works have investigated edge storage for caching [55] or buffering of IoT data [56]. It is worth noticing that most of these works assume the data to be short-lived. However, storing non-ephemeral data on unreliable edge nodes requires replication mechanisms, as demonstrated in [57]. The savings in data transfers to the cloud when using edge computing has been demonstrated in practice with various use cases, from document synchronization [58] to mobile gaming [59]. For example, Hao *et al.* [58] demonstrate a reduction of up to 90% for the data that is sent to the cloud.

Energy savings and increased energy efficiency: Mobile devices have an inherently limited battery life. Advances in battery technology have not kept pace with the increased processing capabilities of modern mobile devices. Furthermore, their small form factors limit the size of the battery. Battery life is an important factor for the overall user satisfaction. Carrying out compute-intensive tasks on the device is detrimental to the device's battery life and, thus, has a negative impact on the user's experience. This factor is even more crucial for small-scale sensors that are deployed in the environment and designed to never be serviced. In this case, the battery life equals the lifetime of the device. Therefore, moving the computations away from the devices is beneficial for their battery life. This has been shown for both cloud [60] and edge [1] infrastructures.

Energy saving is not only important for end devices, but also for edge nodes on which the computations take place [61]. Hence, many works have presented energy-efficient mechanisms for resource allocation [62], offloading [63], [64], and data delivery [65] in edge computing. Xiao and Krunz [66] suggest cooperative offloading, in which edge nodes forward tasks among each other. The authors study the trade-off between quality of experience for users and the fog nodes' energy efficiency and present a cooperation strategy for optimal workload allocation.

The previous examples have outlined the partial benefit from the point of view of mobile devices and edge nodes. However, it is important to note that to analyze the overall energy benefit of edge computing, we need a more holistic view. While offloading might save battery life on the mobile device, this does not answer the question whether the chosen surrogates are more energy-efficient compared to cloud computing infrastructures. As one approach, Jalali *et al.* [67] take into account the energy efficiency of the access network that is used when performing edge computing. The authors conclude that micro data centers at the edge can indeed be more energy efficient than cloud computing. They further identify processing of continuous data streams as an ideal edge application, especially when those data streams are on end user premises and have a low access rate (e.g., video surveillance). Boukerche *et al.* [68] survey energy-efficient offloading in Mobile Cloud Computing from the perspective of both the mobile device and the cloud infrastructure. We argue that this is not only relevant in cloud computing, but also in edge environments, where surrogates might not be optimized for energy-efficient computations. The authors consider different types of deployments and especially mention the possible energy overhead of the offloading process.

Better privacy and data protection: In cloud computing, users typically have little control over their data and where exactly it is processed. Yet, users' end devices generate more and more data at the edge, many of which is personalized and privacy-sensitive. As users become more sensitive to privacy issues, they might not be willing to accept the current practice of how data is processed. For example, Davies *et al.* [69] outline how privacy concerns hinder user acceptance of IoT deployments.

Edge computing offers the opportunity to act as a privacy-enabling mediator between the user's data and cloud-based services, especially when users have access to edge infrastructures that are within their trust domain or that are operated by trusted providers. Edge computing allows to apply privacy-preserving mechanisms (e.g., as proposed in [70]) early in the processing chain and close to the data source, hence reducing the impact of potentially untrustworthy processing entities that subsequently handle the data. Satyanarayanan [7] outline how data owners can use edge resources to enforce privacy policies prior to transferring the data to the cloud for further analysis. For IoT applications, Lu *et al.* [71] present a privacy-preserving data aggregation scheme.

Besides data from individuals, data collected in public spaces is also relevant to privacy. For example, a camera mounted on top of a road intersection captures video streams that are used to optimize traffic and dynamically adapt the traffic lights. This application may be realized in different processing steps, e.g., detecting cars in individual lanes, aggregating their number, computing a strategy to optimize the traffic and so forth. To preserve drivers' privacy, blurring of license plates would be a critical task that has to be carried out at the edge before transferring the video streams for further analysis. Similarly, Basudan *et al.* [72] present an encryption scheme to ensure privacy when monitoring road conditions. Other works explore privacy-preserving publish-subscribe mechanisms at the edge [73] or how edge infrastructures can help in the dissemination of information containing certificate revocations [74].

While many of these potential benefits are acknowledged in literature, less attention has been directed to the possible drawbacks of edge computing. In particular, we consider the following aspects to be problematic:

Unreliable devices: Because edge computing relies on small-scale, often consumer-grade devices for opportunistic usage, their reliability cannot compete with advanced measures for reliability in data center infrastructures, such as UPS[3], emergency power systems, redundant cooling, redundant network connections and high-speed interconnections that enable large-scale replication. Edge computing must therefore either be tolerant of failures or mitigate the effects via replication schemes, e.g., by replicating stored data across edge nodes [57].

Low individual computing power: The computing power of individual edge nodes is usually much lower compared to a cloud data center. For latency tolerant heavy computations,

[3] uninterruptible power supply

such as neural network training, the cloud will remain the predominant model.

Load balancing: Since the capacity at each edge site is limited, it is much more difficult to scale out edge applications with high demands in a small area. Because central data centers are designed to serve large geographical areas, local spikes in demand, e.g., during a city festival, are small in comparison to the total demand. Meanwhile, edge computing infrastructure may be overwhelmed in such a situation as the area over which extra demand can be distributed while still fulfilling good quality of service might be limited.

High operational expenses: Edge computing is likely to be more expensive than traditional cloud computing, which benefits much more from economies of scale. Cost benefits of large-scale data centers[4] [75] that cannot be exploited in edge computing include rental cost, energy cost, and personnel cost. Data centers are often built in places with low taxes, low land cost, and low energy prices. They concentrate vast amounts of homogeneous servers and networking hardware in one easy to reach location. Meanwhile, cloudlets must be geographically much closer to their clients, which prevents strategic positioning in low-price areas. Also, due to the distribution over many small-scale locations, the maintenance is much more complex. For edge computing to be economically viable, the resulting higher cost must be compensated by lower data transmission costs or other benefits, like increased privacy or the need for ultra-low latency.

Security and trust: The idea to opportunistically leverage devices in one's surroundings to carry out computations and store data naturally raises concerns about such a system's security and trustworthiness. According to [76], existing mechanisms for the cloud cannot be applied to edge environments. Roman *et al.* [23] survey the security threats and corresponding challenges in edge and fog computing. To make edge computing pervasive, we need unified trust models and authentication mechanisms across stakeholder boundaries. Another future challenge is how edge systems will react to malicious nodes. For example, Stojmenovic and Wen [77] sketch a stealthy man-in-the-middle attack carried out by edge computing nodes.

B. ACCESS TECHNOLOGIES & COMMUNICATION PATTERNS

We now turn our attention to the typical access technologies and communication patterns in edge computing. While cloud computing is accessed through wired backhaul connections, one characteristic of edge computing is that clients are typically connected to the edge resources through wireless gateways. Edge resources are often either colocated on those gateways or within 1-hop distance. To connect to wireless gateways, client devices use different access technologies. The most common are WiFi [50] or cellular [78] connections. In this domain, we can observe development in two aspects: (i) new communication standards emerge. Examples are 5G networks that are expected to be deployed soon. 5G not only promises much higher bandwidth and lower latencies compared to current cellular networks, but it will also provide additional services like context-awareness on the network access layer [79]. Other future wireless access technologies include millimeter wave [80] and visible light communication [81]. (ii) Existing access technologies are embedded into new devices that can act as gateways. One example in the urban space are street lamps. While today only providing lighting, emerging *smart* lamp posts are designed to offer colocated access and computing resources [82], [83]. A second example is to leverage computing resources present on modern cars [84]. It has also been suggested to place computing resources on UAVs[5] [85], [86].

Besides the wireless access technologies, we can also distinguish edge computing systems by their communication patterns. Generally speaking, our computing world consists of humans and various things that are connected. Depending on which entities communicate with whom, different terminology is used, such as Machine-to-Machine (M2M), Device-to-Device (D2D), Car-to-Infrastructure (C2I), Car-to-Car (C2C), etc. The important distinction between all those terms is whether we have autonomous communication between devices or humans actors in the loop.

C. DEVICE ECOSYSTEM

The initial motivation for edge computing stems from the vast increase in mobile devices and sensors that gather and need to process data. One prominent example are today's smartphones. According to a recent study[6], the number of mobile broadband subscriptions has reached 6 billion as of today and will grow to over 8 billion by 2024. Similarly, the IoT aims to connect a variety of objects such that those are able to communicate with each other [87]. These connected things form smart environments through the joint use and processing of data. Other end devices include smartwatches, smart glasses, and personal on-body sensors. Common to all of them is that they generate large amounts of data that needs to be further processed (and sometimes stored or shared) to provide additional services.

Not only end devices are heterogeneous, but also the devices on which the (edge) computations take place. Every device in the vicinity of the mobile client that has spare resources to perform computations can be considered for edge computing. This can range from consumer-grade hardware to hardware designed for data centers. For example, small-scale single-board computers such as Raspberry Pis have been used in edge computing [88], [89] as well as home routers [90] or compact setups with more powerful hardware

[4] https://vertiv.com/globalassets/documents/brochures/costtosupportcomputereport_11-11_76502_2.pdf (Accessed: 2019-08-08)

[5] unmanned aerial vehicle

[6] https://www.ericsson.com/en/mobility-report/reports/june-2019 (Accessed: 2019-06-21)

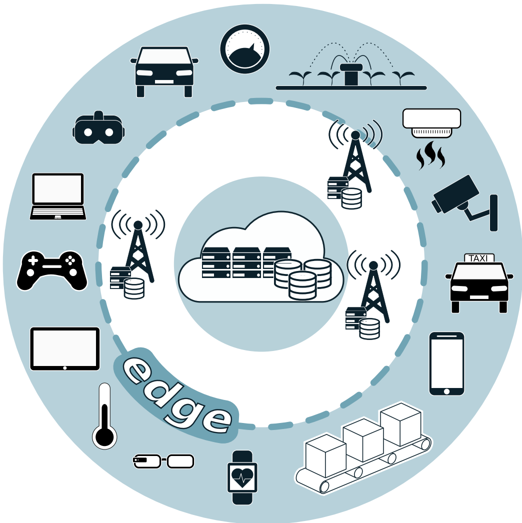

FIGURE 2. Edge computing device ecosystem.

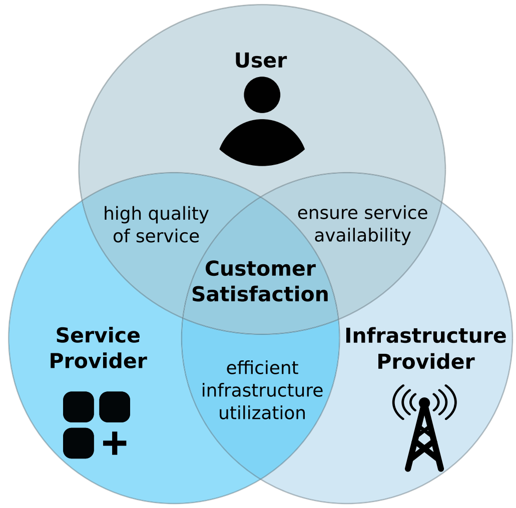

FIGURE 3. Stakeholders in edge computing, adapted from [83].

[61]. In between end-user locations and cloud data centers, we can also leverage network middleboxes that have additional computing capacity available. This was the initial use case for Cisco's vision of fog computing [15]. As we move closer to the edge of the network, we expect to find devices with fewer resources, but in greater number. This reverses as we move closer to the cloud because most locations at the edge of the network cannot physically accommodate the resources found in a data center.

In general, the ecosystem of devices is very heterogeneous, both in terms of their functions and form factors, but also regarding their capabilities and computing power. This device heterogeneity also constitutes one of the main challenges in edge computing. To conclude this section, Figure 2 shows an overview of the device landscape in edge computing.

D. STAKEHOLDERS

In the previous sections, we discussed edge computing's heterogeneity w.r.t. applications and devices. The variety of stakeholders is another dimension of heterogeneity in edge computing. Stakeholders in this context are individual users or organizations that (i) use services deployed at the edge, (ii) operate edge infrastructure or (iii) benefit from edge deployments. As defined in [83], we can identify three types of stakeholders: The *users* of a service, *service providers*, and *providers of edge computing infrastructure*. Figure 3 summarizes the different stakeholders in edge computing and their respective interests.

Users have a certain expectation on the quality of service delivered by applications. Furthermore, they expect a high availability of the service. For many future use cases, leveraging edge computing to meet those demands will be indispensable. Service providers in turn are responsible for ensuring that their services can meet these demands to satisfy their customers. For infrastructure providers (e.g., ISPs and operators of cellular networks) edge computing can be an opportunity to generate additional revenue. This for example can be achieved by either offering resources at the access network or renting out space for server colocation at those access gateways. Besides commercial offerings, we can also imagine that edge computing infrastructures will be offered free of charge; for example, cities could provide those as a service to their citizens [91]. Similarly, private citizens could offer resources for free. This kind of sharing economy has already been seen for providing WiFi connectivity [92]; hence, we believe that offering computing power is the next step.

The diverse ownership of edge resources and the lack of a unified business model is also one of the major reasons why finding appropriate business models for edge computing remains a challenge (see Section VI-D).

E. ENABLING TECHNOLOGIES

Edge computing relies on a variety of technologies that have fostered its development. For example, encapsulating and moving parts of an application would be difficult without lightweight virtualization standards. The most important of those enabling technologies are summarized in this section.

1) OFFLOADING MECHANISMS

Computation offloading—sometimes also named cyber foraging [93], [94]—is the process to run an application or parts thereof outside the client device [40], [95] on a so-called *surrogate*. The motivation for computation offloading can be twofold. The first reason is related to the limited resources of mobile devices. Either the device does not have enough resources, or the resources would only be able to produce an inaccurate or unsatisfiable result. In many cases, offloading to powerful surrogates can reduce the execution time [96] of the task. The other benefit of offloading is related to energy considerations, as pointed out in Section III-A. In this regard, computation offloading can help to save energy on a mobile device and hence prolong its battery life, as demonstrated in [60], [97]. Furthermore, the parallel execution of offloaded tasks can greatly improve the system's scalability [44]. Kumar *et al.* [95] and Sharifi *et al.* [98] provide extensive surveys on computation offloading.

To perform offloading, it needs to be decided *what* to offload, *when* to offload and *where* to offload to. What to offload is typically determined by an application profiler that partitions the applications into offloadable and non-offloadable parts [42], [43]. Besides dynamic approaches, parts to be offloaded could also be specified manually, e.g., through annotations made by the developer. When and where to offload is decided by scheduling mechanisms [99]. It is worth mentioning that deciding if something should be offloaded requires to take into account the overhead. Offloading typically also means that the logic and in some cases the execution environment has to be transferred to the surrogates, which in turn consumes energy and incurs additional latency. This is especially true for low-quality wireless connections. Therefore, the offloading decision should be based on a careful trade-off and ideally take contextual information (e.g., remaining battery, network conditions, and QoS requirements) into account [100]. This is a well-studied optimization problem. As an example, the work of Chen *et al.* [101] focuses on the offloading decision itself. Miettinen and Nurminen [102] analyze the critical factors for energy-efficient offloading. Many applications [103]–[105] take into account the different characteristics of edge and cloud and hence, offload delay-tolerant tasks to the cloud and time-critical tasks to the edge.

Offloading can be done in different granularities, e.g., entire virtual machines [106], threads [107], or pieces of code [42]. Most of the existing approaches are aimed at MCC, i.e., the surrogates are located in the cloud. One notable exception is *Paradrop* [108], a platform for the deployment and orchestration of containerized applications on WiFi access points. Golkarifard *et al.* [109] present a framework that supports both cloud and edge offloading for wearable computing applications. Orthogonal to those approaches, Gedeon *et al.* [110] have suggested the concept of a repository where the services to be offloaded are stored and, thus, do not need to be transferred from the client to the surrogate.

2) LIGHTWEIGHT VIRTUALIZATION

Offloading computations requires the packaging of application logic and/or execution environments into units that can be re-used across runtime instances. Because edge computing resources typically feature multi-tenancy, offloaded applications typically run in virtualized environments. Virtualization serves the following two main purposes: flexibility and isolation. These two form a trade-off and need to be balanced carefully in edge computing environments. On the one hand, virtualized environments should have little overhead in terms of management complexity and startup time when comparing them to processes running on a shared operating system. On the other hand, given the multi-tenant nature of virtualized environments, security and privacy considerations dictate strong isolation between different applications on the same edge device. The type of virtualization environment furthermore determines which techniques for application migration can be used [111].

The visionary paper that introduced cloudlets [12] envisioned virtual machines as the virtualization layer. However, virtual machines encapsulate entire operating systems and while they provide good isolation, they incur a large overhead in terms of size and startup time. Hence, in practice two viable virtualization technologies have emerged for edge computing [112]: containers and unikernels.

Containers do not need a separate guest operating system for each application. Instead, this technology uses virtualization on the operating system level. The operating system, its kernel and libraries are shared and, hence, the isolation is weaker compared to VMs. However, this weaker isolation is traded for an enhanced performance, e.g., by a lower startup time and a smaller size of application images. Ramalho and Neto [113] have analyzed this performance difference between containers and VMs in detail. Using containers also simplifies the management of hardware resources, since only one OS has to be maintained. A container engine provides a format for bundling applications and an interface to control the execution of containerized applications. An important feature of container-based virtualization is that components can be re-used across different containers. Often, a so-called base image is used, on top of which additional components and dependencies are loaded. Consequently, the image of a containerized application is rather small, as both the base image and additional dependencies are typically loaded upon the container's invocation. This allows shipping applications as smaller, portable units compared to virtual machines. These are desirable properties that make containers a viable virtualization technology for edge computing [114]. Besides access to virtualized resources like CPU or memory, the container engine can also provide applications with network connectivity, both internally as well as mapped to the operating system's ports. From a technical perspective, linux-based container engines use two main features of the kernel to realize containerized applications: *cgroups* to control resource utilization and *namespaces* to provide isolation. In practice, Docker[7] has emerged as the de-facto standard container platform, although other products exist, e.g., LXC/LXD[8], OpenVz[9], Rocket[10], and podman[11]. One advantage of Docker is the easy syntax of the *Dockerfile* through which containers are defined. In addition, powerful tools for the orchestration of Docker containers exist, most

[7]https://www.docker.com/ (Accessed: 2019-08-09)
[8]https://linuxcontainers.org/ (Accessed: 2019-08-09)
[9]https://openvz.org/ (Accessed: 2019-08-09)
[10]https://coreos.com/rkt/ (Accessed: 2019-08-09)
[11]https://podman.io/ (Accessed: 2019-08-09)

notably Docker Swarm[12] and Kubernetes[13]. Consequently, Docker has been used in various publications to implement edge computing prototypes, e.g., [89], [108].

While recent efforts have been directed toward making container images more lightweight [115], unikernels are an even more lightweight approach to virtualization than containers [116], [117]. Contrary to containers, which share all of the operating system's libraries, unikernels can be considered an isolated bootable image that can run on bare metal or a type-1 hypervisor. The difference to virtual machine images is that unikernels contain not a complete operating system and its libraries, but only the parts required to execute the functionality, hence they are often referred to as *library operating systems*. Everything required to run the application is compiled into the unikernel's image. At runtime, unikernels run as a single-process and have no distinction between user space and kernel space. This architecture leads to a very fast boot time and execution speed of unikernels, since many management functionalities such as context switching, scheduling, and the management of virtual memory are non-existent. Furthermore, their reduced attack surface makes them inherently more secure. Due to their restrictions, e.g., no forking is supported, not every application can immediately be packed into a unikernel. Some projects like Unik[14] aim to automate unikernel compilation and deployment, but in general, this is highly specific to the particular unikernel and a unified orchestration of unikernels remains challenging. The unikernel landscape is rapidly evolving, with new projects constantly emerging. At the time of writing, MirageOS[15], Rumprun[16], OSv[17], ClickOS[18], and HalVM[19] are among the most popular and active unikernel projects. Some works have already applied unikernels to edge computing environments. Cozzolino *et al.* [118] have suggested unikernels for edge offloading. Wu *et al.* [119] advocate the concept of a rich unikernel to support various applications. As a proof-of-concept, the authors integrated Android system libraries into a OSv unikernel.

3) SOFTWARE-DEFINED NETWORKING

Having emerged from the paradigm of *active networking* [120], the idea of Software-Defined Networking (SDN) is to separate the control plane from the data plane of the network [121]. This follows the general trend of *Softwarization* and brings many advantages from the management and operation's perspective of computer networks. It makes networks more adaptable and simplifies their management, as devices do not have to be configured independently and manually. Instead, configuration is done through a (logically) centralized control entity. Users can specify rules in a high-level way which are then translated by the controller and applied to network devices via protocols like *OpenFlow*. These protocols can be used to perform management functionalities in the network, such as defining flows or network slices.

SDN can be a crucial enabler for edge computing because of two main reasons. First, it abstracts away complex management tasks from the user. For example, the control plane could be responsible for the placement and orchestration of services. Users would just specify what service they request, and the decision where to instantiate it would be taken care of by policies at the control plane. Similarly, the controller's global view can be leveraged for service discovery and to collect measurement data on the state of the network. Second, the capability of SDN to dynamically reconfigure the network is crucial in dynamic edge environments. For instance, these dynamics are related to user mobility or changes in service demands. In case of necessary migrations, e.g., due to intermittent connectivity to unreliable compute nodes, SDN-enabled networks can push new flow rules to the network in order to redirect traffic. Furthermore, SDN can also help to provide guarantees on the quality-of-service of edge services, e.g., by reserving bandwidth on network links.

While SDN today is primarily used for the management of devices in the core network, its principles of software-defined control can also be applied to edge environments. For example, Heuschkel *et al.* [122] present a protocol to extend software-defined control beyond the core network to the end devices. Bi *et al.* [123] show how user mobility can be realized by decoupling mobility control and data forwarding. An extensive overview of how edge computing can benefit from SDN can be found in [26].

4) NETWORK FUNCTION VIRTUALIZATION

Network Function Virtualization (NFV) is the concept of taking network functions out of specialized hardware appliances and instead deploy and run software-based approaches on a completely virtualized infrastructure. Examples for such network functions include deep packet inspection (DPI), firewalls, software-defined radios (SDR) and network address translation (NAT), among others. Because all the functions are in software and run on virtualized hardware, making changes to them is very fast and easy. One popular example of an NFV platform is ClickOS [124]. A comprehensive survey about the current state of NFV can be found in [125].

The main benefit of NFV for network operators and service providers is to make the deployment and operation of their network more cost-efficient [126]. Moving toward virtualized network functions is also interesting in view of new network technologies, e.g., Abdelwahab *et al.* [127] show how NFV can help to fulfill the requirements of 5G networks. Hence, there is a big interest in NFV as a business model.

Similarly, we can observe a kind of ''chicken-or-egg problem'' when it comes to the question why no widespread edge infrastructure is available yet, e.g., at the radio access

[12] https://docker.com/products/orchestration (Accessed: 2019-08-09)
[13] https://kubernetes.io/ (Accessed: 2019-08-09)
[14] https://github.com/solo-io/unik (Accessed: 2019-08-09)
[15] https://mirage.io/ (Accessed: 2019-08-09)
[16] https://github.com/rumpkernel/rumprun (Accessed: 2019-08-09)
[17] http://osv.io/ (Accessed: 2019-08-09)
[18] http://cnp.neclab.eu/projects/clickos/ (Accessed: 2019-08-09)
[19] https://github.com/GaloisInc/HaLVM (Accessed: 2019-08-09)

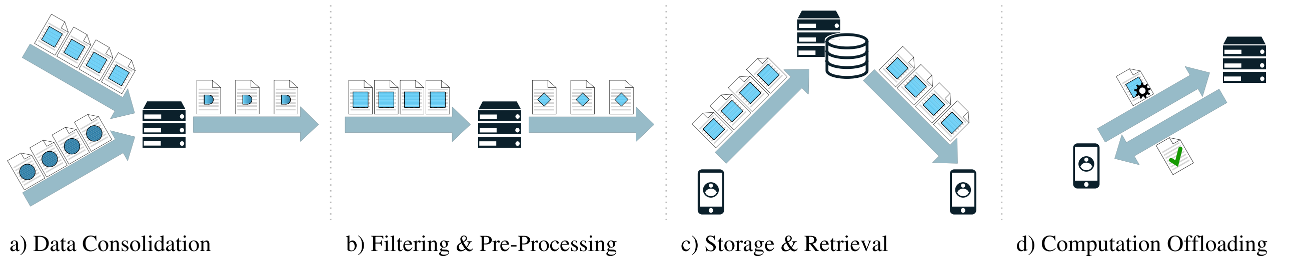

FIGURE 4. Components of edge applications.

network (RAN). Taking the example of a RAN, an infrastructure provider currently might not see a business opportunity for edge computing, because very few novel applications that would benefit for it (and who would be willing to pay to use the service) exist. Similarly, the lack of an edge computing infrastructure hinders the development of such applications. In our opinion, the adoption of NFV can help to break this vicious cycle. Since new commercial off-the-shelf server hardware is already being deployed and virtualized, the foundation to run edge computing already is there, albeit for a different reason. Currently, this hardware serves to implement NFV, but with extra capacity, operators will rent out computing power at those locations. Hence, we believe that NFV is a crucial enabler for edge computing, at least in the access network.

IV. EDGE COMPUTING APPLICATIONS

In this section, we will dissect the application landscape that revolves around edge computing. To do so, we first define four basic components that applications can build upon. Second, we take a detailed look at applications that can benefit from edge computing. We classify the applications into four categories. For each category, we outline representative examples found in recent publications. Our main contribution consists of analyzing the mapping between an individual application's characteristics and how well they can be served by executing the application or parts thereof at the edge. This helps to get a clearer picture of how the challenges differ for different applications and where using edge computing can truly be beneficial.

We define the following four components that applications can use: (i) data consolidation, (ii) filtering & pre-processing, (iii) data storage & retrieval, and (iv) computation offloading. They are visualized in Figure 4. Note that data consolidation and filtering can be viewed as special cases of storage and computation offloading that concern the flow of data. The components are also subject to distribution themselves, which opens up new directions in edge computing research. Concrete implementations of the components may have different levels of complexity and applications can combine multiple of those components.

Data consolidation: We refer to data consolidation as the concept of combining data from multiple sources and reducing it to a smaller, more meaningful joint representation. For example, distributed complex event processing (CEP) [128] could save bandwidth by placing consolidating operations at the network edge [129], [130]. A simple example of such a consolidating operation would be averaging temperature readings over a factory floor. Executing data consolidation tasks at the edge instead of in the cloud has the potential to greatly reduce end-to-end latency and required bandwidth. The amount of bandwidth savings depends not only on the relation between the input bandwidth and the output bandwidth of the data consolidation task but also on the task's location in the network. For example, averaging sensor data on-site might require orders of magnitude less overall bandwidth than averaging the data in the cloud. Furthermore, tight latency requirements might make consolidation in the cloud infeasible in some cases.

Filtering & pre-processing: The purpose of this component is twofold: discarding irrelevant data and pre-processing data. Since not all data is equally important, bandwidth savings can be achieved by discarding irrelevant data before it is transmitted for further processing. A simple example could be thresholding of temperature readings in an application where an alarm should be raised when a certain value is exceeded. In such an application, temperature readings are irrelevant as long as they are within the normal range and thus need not be transmitted. Besides saving bandwidth, reducing data locally can also help to save energy and reduce local storage needs [131]. In pre-processing, data is transformed from one representation to another. Besides discarding data, which could be interpreted as a special case of such a transformation, other possible transformations could be the aggregation of data streams over time, data compression, data alteration, or bridging between formats. For instance, real-time video analysis, a likely killer app for edge computing [132], [133], has the potential to save vast amounts of bandwidth by only forwarding results of the analysis, e.g., the number of objects in the frame, instead of entire video streams. To give a concrete example, Powers *et al.* [134] use cloudlets to pre-process data for a face recognition application. Both of these aspects can save bandwidth, depending on the ratio of data discarded and how much data is reduced by the pre-processing. Furthermore, in case of time-critical data stream processing applications, distributing such operations entirely at the edge can reduce end-to-end latencies substantially [135], [136].

Data storage & retrieval: Applications might want to store data outside of the device for several reasons. First,

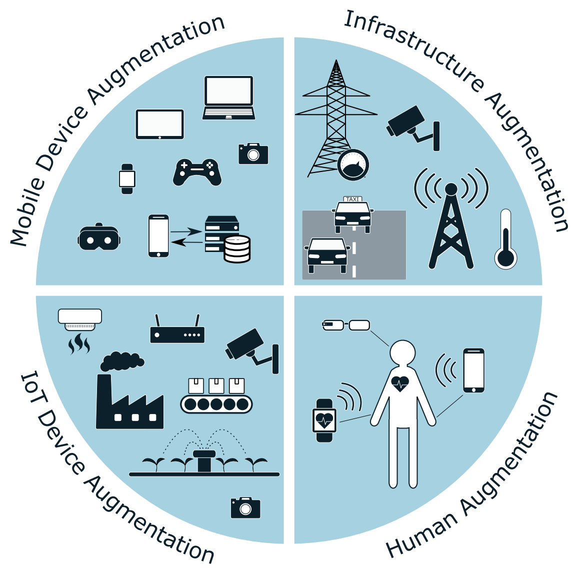

FIGURE 5. Categories of applications.

additional external storage helps to overcome the device's storage limitations. Second, data often needs to be shared across different users and applications. Whenever data is only of local relevance, leveraging the edge, i.e., storing the data on close-by devices, is beneficial for access latencies and bandwidth utilization in the network. Often, this data will be contextual data and short-lived information captured by the user. Storage can either be ephemeral, i.e., short-lived or permanent. In the latter case and if we consider unreliable devices, replication is required. If data is replicated, the client needs to make a decision from where to retrieve it, ideally considering the abovementioned metrics of latency and bandwidth. Section IV-A3 presents several examples of approaches related to data storage and retrieval.

Computation offloading: Computation offloading is the process of executing a (often resource-intensive) task on a remote resource. The client transfers the input data and in some cases the code and execution environment and retrieves the result. For example, offloading computation-intensive neural networks [137]–[139] or blockchain operations [33] from a mobile device could decrease the execution time and save battery energy. We refer the reader to Section III-E1 for a more detailed description of this concept.

We now survey applications for edge computing in detail. Contrary to previous works whose classifications are rather enumerative, our classification starts from the very purpose of edge computing, i.e., to bring data and computations closer to where data is generated and results of the computations are needed. To do so, we define four categories around the notion of *augmentation*, as depicted in Figure 5. On a top-level, we can divide the world into humans and things. For humans, we distinguish between separate (mobile) devices used and/or carried by the user (*mobile device augmentation*, Section IV-A) and devices that are interwoven to a higher degree with the user, e.g., devices that are worn or connected and affect the user's body function (*human augmentation*, Section IV-D). For *things*, following the widespread IoT terminology, Section IV-C defines one category as *IoT device augmentation*. However, we also want to stress the benefits of edge computing in enhancing the smartness of public spaces and larger environments as opposed to single devices in closed ownership. Hence, we also define the category of *infrastructure augmentation* (Section IV-B). We acknowledge that some use cases may be considered to belong to more than one category; however, we share this issue with the vast majority of categorizations in every domain. Nevertheless, we argue that our classification has the advantage of being simple, inclusive and at the same time open to fit future applications.

For each of the categories, we describe representative use cases and analyze the benefits of edge computing w.r.t. the four promises as defined in Section III-A. Note that regarding energy saving and energy efficiency, we will only consider the potential energy saving and hence battery life prolongation of end devices as battery life is often much more relevant than the total energy footprint. Energy savings in network equipment are sufficiently represented by the bandwidth criterion. We do not restrict our analysis to papers where edge computing has been proposed or that are currently associated with edge computing use cases, but include others for which at least some benefits of edge computing apply. Furthermore, following our argumentation from Section II, we also include literature that relates to fog computing. This helps to get a comprehensive overview of potential use cases for edge computing. At the end of this section, we summarize our findings for the most pertinent use cases. Those references that appear in Table 2 are marked with an asterisk (*) throughout the following subsections.

A. MOBILE DEVICE AUGMENTATION

The applications outlined in this section are specifically targeted at consumer's mobile devices, such as smartphones, tablets or head-mounted devices. Specifically, we look into mobile gaming (Section IV-A1) and emerging virtual/augmented reality applications (Section IV-A2). In addition, new applications and usage patterns require appropriate strategies for data storage and caching (Section IV-A3).

1) MOBILE GAMING

In gaming, we observe the trend toward (i) more mobile games (i.e., games played on a connected handheld device), (ii) more games that interact with the player's environment or other player's in close vicinity and (iii) business and deployment models based on *Gaming as a Service* (GaaS). As described in [140], GaaS is the concept of providing scalability and overcoming hardware limitations through modularization of the game. For example, certain functionality (e.g., rendering) is migrated from the mobile device and offloaded to a server. Functionalities like rendering are computationally very demanding and hence, offloading them can significantly improve battery life.

Most games are highly interactive and, thus, players expect crisp responsiveness. The responsiveness is influenced by the computation and communication time of the offloaded components and possibly the latency to other players. Pantel and Wolf [141] claim that depending on the type of the game, only 50 ms–100 ms of delay is tolerable. As analyzed by Choy *et al.* [142], the cloud will be unable to meet latency requirements of gaming applications (assuming a target of 80 ms). To solve this issue, the authors propose to extend the cloud infrastructure with local content delivery servers to serve users' demands. This has led big companies to extend their data centers toward the edge. Plumb and Stutsman [143]* demonstrate how exploiting Google's edge network can reduce the latency for massively multiplayer online games. Specifically, they define an *area of interest latency* as the latency between players that interact in the virtual world. The authors report an improved mean latency from 52 ms to 39 ms and for the 99th percentile an improvement from 110 ms to 91 ms. As shown before, this can make the difference between an enjoyable experience and an unplayable game.

For these reasons, extending the infrastructure for mobile gaming to the edge makes sense. In addition, tasks like rendering typically require only little data as input (e.g., the user's position, field of view or current action) while the size of the returned data (in this example the rendered object) is much larger. This observation matches the asymmetric down- und uplink bandwidth that end users today have in cellular networks. Messaoudi *et al.* [144]* present a general framework for offloading parts of a modularized game engine. Lin and Shen [5]* extend cloud gaming with fog nodes that are responsible for rendering game videos and streaming them to nearby players. Similarly, the work of Kämäräinen *et al.* [145] supports the deployment of game services in hybrid (public, enterprise, private) cloud infrastructures. Using local processing, they were able to almost halve the delay. Furthermore, their results indicate that connecting to a cloud via WiFi is less detrimental to the device's energy consumption compared to 4G cellular networks.

Cai *et al.* [146]* investigate a scenario in which neighboring players cooperate in a game. They advocate cloudlets in the vicinity of players to reduce bandwidth and latency issues and also consider the energy dimension, both for the mobile device and the overall energy cost for data transmission. In many games, players are mobile, e.g., when the game requires them to interact with their environment or visit different locations. Hence, the issue of migrating offloaded parts of the game arises. Braun *et al.* [147] propose an application-level migration technique for latency-sensitive gaming applications. The server is transparently live-migrated during gameplay. The authors argue that the advent of 5G networking will provide ultra-low latency and high-bandwidth to mobile devices. Hence, it would make sense to locate edge game servers at those locations.

2) AUGMENTED REALITY AND VIRTUAL REALITY

Augmented Reality (AR) and Virtual Reality (VR) have gained enormous traction due to the advent of new devices such as the Microsoft *HoloLens* or HTC *Vive*. In augmented reality, virtual objects are integrated into the environment [148]. Augmented reality applications extend the user's real-world view by inserting virtual objects into the environment, related to one's context and movement. Virtual reality (VR) is defined by creating a sense of *presence* in simulated environments [149]. Both variants are typically interfaced to the user via a head-mounted display (HMD). Applications leveraging VR and AR mainly fall into the categories of information (e.g., in retail [150] or for tourist guidance [32]), education [151], and entertainment [152]. Common to all of them is that computationally heavy tasks need to be carried out in order to render and understand a scene.

AR/VR applications are subject to stringent real-time demands on their responsiveness. For head-tracked VR, it has been shown that the JND[20] for latency discrimination is 15 ms [153]. Especially in VR, motion sickness can occur if the delay between tracking the movement and rendering the scene exceeds this value.

Because high framerate 3D rendering is computationally demanding, many headsets carry out those tasks on a standalone computer, connected to the headset through a cable. Removing this cable is desirable for the user experience, but challenging due to the high data rates a wireless channel must guarantee. Some attempts have been made by using new wireless communication technologies like mmWave [80] or by partitioning the computations between the HMD and the rendering server [154]. The latter is required because computational capacity (especially GPU) remains limited on end devices. At the same time, the stringent latency requirements prohibit offloading to the cloud. Hence, the challenge is to design good offloading strategies to nearby rendering servers while coping with current wireless communication technologies. Braud *et al.* [155] have analyzed these challenges with a focus on mobile users. Furthermore, as outlined in [156], battery life and latency form a trade-off.

Despite these challenges, a few solutions to leverage edge computing in AR and VR have been proposed. Elbamby *et al.* [157] have envisioned a combination of edge computing for computation and mmWave for communication as a crucial enabler for wireless VR. Lai *et al.* [158]* have presented the idea of a cooperative renderer. The authors base their system on the observation that less-predictable scene updates are typically more lightweight and therefore, those are rendered at the edge. For highly predictable scene changes, pre-rendered frames are fetched from nearby locations. In addition, compression and panoramic frames are used to reduce the load on the wireless link. Similarly, Shi *et al.* [159]* have presented a rendering scheme that saves more than 80% of bandwidth and therefore enables to deliver VR content over 4G wireless links. Liu and Han [160]

[20] just noticeable difference

have presented *DARE*, a novel network protocol that enables mobile users to dynamically change their AR configuration. Specifically, it adapts to changes in network conditions and load on edge nodes—both crucial in dynamic edge environments. Tirelli *et al.* [156] have used an approach based on NFV. They have developed a framework for live video augmentation by extracting and injecting video streams from or into network flows.

Besides partitioning workload only between the end device and one edge node, some workloads benefit from distributed processing across several cloudlets. For example, Bohez *et al.* [161]* reconstruct 3D maps from the depth cameras of AR headsets. They do this by partitioning sub-models across geographically distributed cloudlets.

Gaming with VR/AR is a popular application scenario, and due to its specific characteristics, we review existing work independent from Section IV-A1. Viitanen *et al.* [162] have presented a rendering scheme for real-time VR gaming that saves energy and computational load on the end device. The rendered views are encoded as HEVC[21] frames and transmitted based on the user's field of vision. In [105], the authors have explored the scalability issue of massively multiplayer games and present a hybrid approach in which changes in the local view of a player are processed at the edge, while global game updates are performed in the cloud. In addition, colocation of multiple players that share a similar view on the same edge node increases the efficiency of the proposed system. Zao *et al.* [163] have developed an AR game with a brain-computer-interface that processes EEG[22] brain activity in real-time. Classifying those readings into game actions is a computationally intensive task. The authors use a combination of edge and cloud computing for this. While the classification is done at the edge, the underlying models in the cloud are continuously adapted according to the sensor readings.

Overall, we see AR and VR applications as one of the most relevant use cases for edge computing, as it combines four important characteristics that benefit from edge deployments: strict constraints on the latency, high-bandwidth data, computationally intensive tasks, and battery-powered end devices. In contrast, security and privacy issues have received only limited attention, e.g., in [164].

3) DATA STORAGE, CONTENT DELIVERY, AND CACHING

Virtually all connected applications need to store or retrieve data from outside the client device. At the beginning of this section, this was defined as one of the components that applications use. We now describe in detail the possible kinds of services that can be offered at the edge.

Content Delivery Networks (CDN) [165] distribute caches across data centers in different geographic regions to provide highly available and performant content retrieval for users. Ericsson forecasted back in 2013 that in 2019 50% of mobile traffic would be video traffic [166]. Today's content delivery networks hence, to a large extent, serve video traffic [167]. This demand for video not only means users have high expectations for the service quality, but also that large-volume data is a huge burden for infrastructure providers and hence, an important incentive to place caches within the access network [35]. Ahlehagh and Dey [168]* have proposed both reactive and proactive caching strategies for video content at the radio access network. Bastug *et al.* [169] have investigated the role of proactive caching in 5G networks. Another example is the retrieval of websites [170]*. Zeydan *et al.* have [171] proposed proactive content caching based on content popularity in 5G networks. Approaches like [172] assume a global knowledge of the content popularity for each location. We argue that this is neither realistic nor scalable in a dynamic edge environment with user mobility. A solution to this might be collaborative approaches as presented in [173]*, where base stations collaborate in replicating content to improve the overall cache hit ratio. Tran *et al.* [174]* have presented a strategy for collaborative caching and processing of on-demand video streams.

Up until today, little research has linked content storage and caching to the domain of edge computing. Drolia *et al.* [175] have presented *Cachier*, a caching system for image recognition applications. It balances the load between the edge and the cloud and exploits the spatio-temporal properties of requests. Psaras *et al.* [56]* have advocated the placement of local storage on WiFi access points to buffer IoT data prior to cloud synchronization. Lujic *et al.* [176]* have proposed a storage management framework for edge analytics. The goal of the framework is to balance the quantity of stored data and the resulting quality of the data analytics, given limited storage capacities at the edge. Such caches can also be used as a distributed way to distribute mobile apps and app updates [177]*, [178]*.

As end devices generate more data, the path of data dissemination changes to a "Reverse CDN" [179]*, [180]*, i.e., the decision is now where to store data generated by those end devices in the network. This decision depends on where the data is to be used and shared. Contextual information is also useful. For example, we can imagine sharing content from large-scale events or tourist sites. In this domain, Gedeon *et al.* [10]* have presented *vStore*, a rule-based framework that uses contextual information of the user to make informed storage decisions. Simoens *et al.* [181] have presented *GigaSight*, a framework for the decentralized collection of videos through cloudlets. Before the videos are indexed and made available, privacy-sensitive information is removed from them (denaturing) by a cloudlet that is specific to one user. The authors have conducted experiments, measuring the throughput and energy consumption and based on the results, modeled a trade-off for the allocation of resources between the cloudlets performing the different computation steps.

Compared to low-latency processing tasks, the benefits of edge computing for storage and caching are less striking and

[21] High Efficiency Video Coding
[22] Electroencephalography

require careful decision-making. Furthermore, using edge nodes for storage can raise the question of resilience in view of their inferior reliability [57] compared to large data centers. However, in cases where data has to be transferred away from end devices, e.g., because it is to be shared, and if that data is only relevant in a certain (geographic) area, edge computing can save bandwidth. For example, Hao et al. have presented *EdgeCourier* [58]*, a framework for live document synchronization. The authors have demonstrated the reduction in network bandwidth usage by performing incremental synchronization at edge nodes. Another example is the work of Hu et al. on face recognition [138]. They have observed that only a subset of biometric features is relevant to identify a face. The paper suggests extracting those features at the cloud and retaining only those at the edge necessary to perform the recognition.

As soon as personal user data is involved, privacy and data security become relevant. The *Databox* project [182]*, [183]* aims at providing a personal data management framework. It enables individuals to manage their personal data and make that data available to others, while retaining control over its usage. Most importantly and in contrast to today's approaches, data is not handed over to third parties. Instead, access and processing is mediated through the Databox. The authors envision a distributed system of personal Databoxes, hosted on a variety of devices, e.g., wireless gateways, lamp posts, and cars. The Databox consists of several components that are logically separated and can run on different physical devices. Access to different sources of data is also logically separated to provide additional security. The project is a good example of how edge infrastructure can help to preserve data security and privacy. However, as noted in [184]*, several challenges remain, such as capturing one's privacy preferences, implementing shared ownership of data or how to approach risk-benefit negotiations for sharing data.

B. INFRASTRUCTURE AUGMENTATION

In this part, we refer to infrastructure as basic services and facilities that we build our lives upon. This basic infrastructure encompasses everything, from our electrical grid (see Section IV-B1) up to emergency services (see Section IV-B4). This infrastructure is especially important and challenging in urban areas as the trend of growing urbanization brings new challenges to create livable environments, such as traffic, pollution, and safety [185]. Hence, the term *smart cities* [186] is used to describe concepts that use information technology to augment infrastructure in urban areas. Schleicher *et al.* [187] define a smart city as a reactive system that makes decisions based on massive amounts of data. Smart cities connect people and objects in the cities in order to create services that enhance the quality of life for citizens of urban environments [188]. Furthermore, monitoring and large-scale data analysis can provide valuable information for municipalities and policymakers, e.g., to plan traffic and transportation (see Section IV-B2) The raw data is collected from sensors that are deployed in the environment. Besides static deployments, humans can also be incentivized to serve as data providers [189], e.g., by providing sensor data through their phones [190]. One application domain where this approach has proven useful is in monitoring the environment (see Section IV-B3).

Previous works in the domain of smart cities have proposed to integrate the captured information through cloud computing [191], [192]. However, we can easily see the potential benefits of nearby cloudlets when high-volume data is used as input for data analytics. A prominent example are video streams from cameras which are ubiquitously present in today's cities. While tolerable delays for smart city applications vary greatly [193], the mere number of sensors will prevent cloud-based solutions from scaling. Perera *et al.* [31] have further surveyed different types of smart city applications that benefit from edge deployments, with a focus on the communication between devices.

1) SMART GRIDS

Energy grids are currently in a state of transformation toward so-called *smart grids*, driven by inherently juxtaposing ecological, economical and political goals. Two important characteristics of a smart grid are (i) information and communication technology is embedded into the energy grid, (ii) its decentralized nature (in contrast to today's strictly hierarchical organization), and (iii) a shift toward *prosumers* that both consume and produce energy. A smart grid is envisioned to support a distributively organized control structure instead of being one large energy grid with central control across all tiers. In particular, this structure is envisioned to consist of multiple cells, which can be controlled individually [194].

As a way to realize the next evolutionary step of cellular energy grids, a *holonic* approach has been suggested [195]. Holons represent entities in a system that are simultaneously a part and a whole. Consequently, a hierarchically organized system structure (called *Holarchy*) emerges, where on different levels, holons exist that encompass other holons, while being part of higher-level holons themselves.

Implementing a Holarchy is a significant challenge, which requires comprehensive ICT[23] for monitoring and automated control [196]*. To establish such a concept, edge computing is a suitable paradigm as it supports data aggregation and filtering, which are both essential to realize the concept of holons. For instance, local consumers can use ICT to optimize the control of the appliances in their smart home to reduce their electricity bill [197]. Simultaneously, this goal may conflict with the process of balancing demand and supply in the overall grid. These problems can now be addressed either locally, using the local controllers responsible for managing individual holons, or the higher level holons can address the issue as they have a larger view on the network. These higher levels of the Holarchy control the holons that they encompass. Hereby, the states of the holons are aggregated

[23] information and communications technology

states of the next lower level. This is necessary to handle the amount of information that needs to be processed by higher level holons and to filter data that is not relevant at that level in the Holarchy (e.g., a holon encompassing a city cannot manage individual devices in each house, but requires aggregated information about districts or streets).

The decision-making in smart grids requires to solve complex optimization problems, like optimally controlling distributed energy sources. One way to tackle the complexity of these problems is to reduce the problem size and, consequently, speed up the optimization process. Edge computing devices can be used as local controllers for holons, which are capable of executing optimization algorithms. Examples of suitable algorithms are presented in [198]*. Moreover, such optimization tasks need to be conducted quickly, as demand and supply deviations in the grid can contribute to the destabilization of the whole energy grid. Edge computing devices facilitate low-latency communication due to their distributed placement. Hence, with edge computing, these problems can quickly be addressed locally, without the need to send the necessary information to a cloud service or a central control center.

Although not obvious at first glance, smart grids contain sensitive data with regards to one's privacy. For example, the authors of [199] have shown that the readings from smart meters can reveal the individual home appliances by analyzing the aggregated smart meter readings. Edge computing devices in combination with smart energy storage technology can address this problem by scheduling the charging of the storage and the consumption of the appliances in a way such that the devices only use the electricity that is currently stored in the home. Consequently, the power consumption profile of the house itself appears to the outside only as the charging behavior of the local energy storage device.

2) SMART TRANSPORTATION & CONNECTED CARS

In big cities, optimizing traffic conditions is a crucial task that tremendously impacts one's quality of life. Like in other types of smart city applications, many primary sensors for traffic-related applications are cameras, whose video streams require further processing. For example, we can think of a smart traffic light that adapts its signal cycles to the actual traffic conditions in certain lanes of the intersection [200]*. While energy is not an issue in these scenarios (sensors are fixed deployments and connected to a permanent power source), this use case requires complex tasks like object recognition (to identify cars) and tracking that might not be feasible given their weak hardware. Analysis of live feeds can also be used to detect traffic anomalies [132] or to recognize license plates [3]. The traffic light example can also be linked to emergency response use cases, e.g., to help making way for an ambulance by setting traffic lights to free up the route and warn others about the approaching vehicle [39]*. In the latter use case, time criticality clearly becomes more important. This is also the case for the detection of immediate road hazards, e.g., as shown in [201]*.

Long-term analysis of urban data can be used to detect and improve flawed urban planning or to identify areas where dangerous situations regularly occur [202]*, [203]*. In these use cases, the long-term analysis would most likely be carried out in the cloud due to the larger amount of available resources. However, we argue that data collection on such a big scale would not be possible without edge computing, due to necessary pre-processing steps (e.g., encoding a video to a lower bitrate or aggregating measurements). Qi *et al.* [204]* have introduced an edge computing platform using on-board computers on public transport vehicles. The platform collects data (e.g., by intercepting WiFi probe packets from phones) to gain insights that help public transport operators to devise better plans (e.g., by identifying popular stations and assessing vehicle occupancy). Such information can be sensitive, as it allows to identify and track individuals. Hence, in this case, the edge infrastructure could also be in charge of performing the anonymization of sensitive data. Besides static planning, this high-volume data could also be used for real-time updates, e.g., to provide passengers with predicted arrival times. This data could be distributed to edge nodes at the relevant location (e.g., to the corresponding WiFi hotspots at a stop). The location-awareness of edge nodes can also be used for applications like toll collection [205] or finding parking spaces [206].

Besides improvements in the planning of city traffic, we see a trend that vehicles themselves are being equipped with more sensing and computing capabilities. In many cases, these components are not isolated, but transform the vehicle to a *connected car* that can interact with its environment and other vehicles. Such cars can form vehicular ad hoc networks (VANET) [207] that communicate with each other or through some close-by infrastructure—often termed roadside unit (RSU). Besides RSU, UAVs have also been proposed as a means to relay communications between cars and/or infrastructure [208]. Datta *et al.* [209]* have presented an infrastructure with a fog computing layer, located at RSUs and M2M-gateways. The type of data and its importance varies in such networks [210], from simple information services (e.g., information about current traffic conditions) to critical, safety-related events (e.g., warnings about emergency situations or sudden breaking of cars ahead). An example of the latter is the work of Cozzolino *et al.* [201]* that uses an edge infrastructure for black ice road fingerprinting. Liu *et al.* [211] have demonstrated that complex tasks, such as recognizing attacks in ridesharing can even be done with little energy impact on the end device.

Edge computing can also be an important enabler for autonomous driving, e.g., by disseminating data from RSUs to vehicles [212]*, or by processing information like point clouds captured by LIDAR sensors [213]*. Besides such latency-critical and compute-intensive tasks, using edge capabilities can also be used for early data aggregation to save bandwidth. In this domain, Lochert *et al.* [214] have presented an aggregation scheme for VANET traffic information. As outlined in [26], strong security mechanisms need to be in place if edge nodes are involved in the control of vehicles in order to make such systems resilient against attacks. Raya and Hubaux [215] have provided a detailed analysis of threats and security architectures in VANETS.

3) ENVIRONMENTAL MONITORING & WASTE MANAGEMENT

Pollution is one of the main problems in growing urban areas, with drastic impacts on people's health. As of today, many kinds of pollution are only measured at very few locations and/or estimated via models based on historical data [216]*, [217]*. Hence, they often fail to provide an accurate and actionable view on current circumstances. However, accurate, real-time information is crucial, both for policymakers and citizens to make informed decisions (e.g., for patients with respiratory diseases or for decisions to restrict traffic). Dutta *et al.* [218]* present *AirSense*, a crowdsensing-based air quality monitoring system. Especially for such opportunistic sensing campaigns, where the location of the data is unknown a priori and constantly changing, the deployment of edge computing resources is both beneficial and challenging.

While environmental data, in general, has a rather low rate of data generation [219], the scalability issue remains [220]*. Edge computing can ensure the scalability of a large-scale sensing system by processing the data close-by and keeping the results in the sensing area. Aggregated or coarse-grained data (e.g., by reducing the temporal resolution) can be sent to the cloud, while the raw sensor readings are processed at the edge. Aazam and Huh [221] have advocated data trimming to reduce unnecessary transfers to the cloud by using a smart gateway that sensing nodes are connected to. Edge infrastructures have also been used to opportunistically deploy different sensing tasks [222]. Zhang *et al.* [223] have presented an allocation scheme for sensing tasks on edge computing nodes. The authors outline the benefits of an edge deployment w.r.t scalability, bandwidth requirements, and better utilization of edge nodes' computing power.

Similar to air pollution, noise pollution is a big problem with adverse effects on people's health. Maisonneuve *et al.* [224]* and Schweizer *et al.* [189]* have proposed deployments in which citizens measure noise through their mobile phones and upload the measurements to a cloud-based service for access and analytics. Besides the potential benefits of edge deployments w.r.t. latency and bandwidth, we also need to consider the privacy aspect. Whenever data is collected by volunteers, privacy has to be guaranteed, otherwise people might not be willing to participate in sensing campaigns. Measurements must inherently contain the users' locations, and hence, this would allow tracking user locations. By processing at the edge, data can be anonymized early, or alternatively, noise can be introduced into the raw data, such that it does not impact the results, but cannot be linked back to an individual. Here, the distributed nature of an edge computing infrastructure itself can be exploited. Marjanović *et al.* [220] have demonstrated how partitioning data and distributing its processing can help reducing privacy threats.

Waste management is a complex process in today's cities and includes the collection, transportation, processing, and disposal of waste. Optimizing these processes can save a city tremendous amounts of money and resources. One way is to optimize the transport routes of garbage collection trucks. Normally these operate at fixed schedules, as no real-time information about filling levels of waste containers is available [225]. Sensors mounted on trash containers could report their filling levels and infer if they need emptying. Based on aggregated data from a neighborhood, the garbage trucks' routes can be optimized. Furthermore, municipalities can provision different sizes of garbage trucks in order to optimize the collection process [226]*. Cloud-based solutions have been proposed [227]*, [228]*, [229]*, however, aggregating the sensor readings at the edge would save bandwidth. This becomes more important if data is annotated with photos or voice messages, as suggested in [229]. Latency and privacy, however, are less of a concern in this use case.

4) EMERGENCY RESPONSE & PUBLIC SAFETY

Detecting emergency situations can be done by inferring events from sensor sources. As soon as an emergency is detected, first responders need to be alerted and directed to the scene. To do so, platforms provide situational awareness and connect first responders to the required data sources. Chung *et al.* [230]* have presented a cloud-based platform, while Aazam and Huh [231]* have extended this idea to incorporate an intermediate fog layer, capable of preprocessing the data and overcome delay problems. Depending on the type of incident, the appropriate emergency departments are notified. Mobile base stations can furthermore serve to notify citizens about a threat. For example, Sapienza *et al.* [232] outline a use case where a fire is detected based on sensor readings and video analysis, and this information is forwarded to car navigation systems in order to alert people.

A distributed infrastructure of edge devices can furthermore be leveraged as an emergency infrastructure in case of disasters. In case of a breakdown of the communication infrastructure, edge cloudlets hosted on private devices like routers can act as emergency devices, providing both computation and communication capabilities [233]*. Satyanarayanan *et al.* [234]* outline a potential use case for such emergency cloudlets in disaster recovery, in which responders take photographs that need to be stitched together in order to obtain a complete overview of the area.

Efforts to increase public safety (e.g., by either preventing or quickly detecting crimes) mostly rely on surveillance. In many cases, the raw data consists of video streams which are then analyzed. Canzian and van der Schaar [235] have presented a hierarchical classifier system for surveillance applications. As the authors point out, the characteristics of the tasks—distributed sources and tasks as well as a high computational complexity—make it necessary to leverage distributed and heterogeneous processing nodes. This definition perfectly reflects the edge computing landscape.

Chen *et al.* [236]* have presented a system for real-time surveillance and tracking of vehicles to detect speeding using a drone. Similarly, Xu *et al.* [237] use a geo-distributed fog computing infrastructure for vehicle tracking. Mihale-Wilson *et al.* [238] have investigated the protective effect of street lamps if they are augmented with functionalities such as video surveillance and gunshot detection via microphones. Their results suggest an increase in safety in areas where such lamps were installed. Just-in-time indexing of video streams across several cloudlets has been demonstrated in [203]*. A practical use case for just-in-time video indexing could be the search for a missing person.

Because all these use cases involve multiple streams of high-volume data and heavy computations, edge deployments are beneficial in terms of latency and bandwidth. Besides offloading computations, data pre-processing is also relevant to some security-related applications. For example, Stojmenovic [239]* has proposed to partition tasks for biometric identification between the mobile device and the cloud.

In all those use cases—especially when video streams are involved—the citizen's privacy is exposed and personal information is subject to analysis. Contrary to other use cases, the privacy-critical information is not incidentally contained in the raw data, but it is the reason for capturing it in the first place. Hence, the positive impact of edge computing in this application domain is limited. At the very best, we could envision to enforce policies to delete personal information once it has been processed by a trusted edge node.

C. IoT DEVICE AUGMENTATION

The IoT refers to connected objects that are able to interact with each other and hence, extend the Internet to the physical world [87]. Originally closely tied to RFID[24] [240] technology, today the IoT encompasses all kinds of sensors, machines, and appliances. An extensive survey about the IoT and its enabling technologies can be found in [241]. The data volume and latency requirements of future IoT devices will likely be challenging to transfer and process at central clouds [242], [243]. In this section, we focus on IoT deployments in three settings: Homes and buildings (Section IV-C1), industrial applications (Section IV-C2), and agriculture and farming (Section IV-C3).

1) SMART HOME/BUILDING

The terms *smart home* and *smart building* describe the concept of collecting data within a building and using it to automate and optimize various aspects of the building. The IoT offers great potential to lower energy/water consumption and increase security and comfort through coordinated management of HVAC[25] systems, lighting, electrical outlets, and various connected devices [244]. Examples for such devices are smart locks, surveillance cameras, TVs, household appliances, or environmental sensors. These devices and building systems produce large amounts of sensitive personal data streams [8]. One classic example of a smart home task is the aggregation and pre-processing of home video surveillance streams [243]*. Such a case was examined by Abdallah *et al.* [245]* in a prototype system. Their results indicate that the storage of all sensor data in the cloud impacts the available bandwidth for the home significantly, pointing to edge computing as a solution.

Edge computing could also aggregate IoT data, and thus, enable cooperation between devices by using sensors and physical capabilities of multiple devices to complete a task [14], [246]. Such a task could be sending a robot vacuum cleaner with a camera to check on suspicious motion through video analysis. Vallati *et al.* [247] have envisioned an MEC-architecture for smart homes that builds on LTE with device-to-device communication for data locality and low-latency. Storing and processing smart home data in-home could also resolve the issue of transferring privacy-critical data [183], [184] and opens up the possibility to combine data from IoT devices with personal data from other services to provide higher-order, yet privacy-aware services. As proposed in [248]*, edge and fog computing could also provide the missing link between various building subsystems that are usually implemented independently. Thus, the resulting integration and interoperability between the individual subsystems, combined with data analysis at the edge, could enable new smart services, like the activation of smart devices when solar power is available.

2) INDUSTRIAL INTERNET OF THINGS

One goal of the Industrial Internet of Things (IIoT)[26] is to improve quality control and increase productivity [249]*, [250]*. Collecting and analyzing vast amounts of data from production processes can be a tool to find inefficiencies and optimize production processes [250]*, [251]*. However, since it is often impractical to store and analyze all collected data at the cloud in real-time, Fu *et al.* [252]* suggest to pre-process, aggregate, and store time-sensitive data on edge nodes, while storing less time-sensitive data in the cloud for later analysis. Early detection of mechanical problems with production machines has the potential to prevent both machine failures and production quality issues [253]*.

Monitoring machines, e.g., with vibration sensors, produces large amounts of real-time data which lends itself to being analyzed at the edge to monitor machine health or to predict tool wear and service intervals in real-time [253]*, [254]*. Another goal of the IIoT is to enable the production of highly customizable products on dynamically scheduled production lines, which can also profit from the low-latency property of edge computing [249]*. Lin and Yang [255]* have presented a further application of edge computing for IIoT, the real-time scheduling of logistics in

[24] Radio Frequency Identification

[25] HVAC: Heating, Ventilation and Air Conditioning

[26] In this article, IIoT is used synonymously with industry 4.0 / advanced manufacturing

highly automated warehouses. However, while edge computing has the potential to augment current manufacturing processes and integrate with operational technology (OT), Steiner and Poledna [256] indicate that it is currently not fit to replace OT which has much stricter real-time requirements.

3) AGRICULTURE & FARMING

Smart farming (or precision farming) targets the management of crop and livestock. Many applications in this domain revolve around automation, e.g., for watering crops or feeding livestock. Monitoring of environmental parameters and tracking of animals can furthermore ensure a timely reaction if abnormal events are detected—a delayed reaction can cause damages and production losses [257]*, and can impact the entire supply chain. More complex tasks include the use of machine learning methods for yield prediction or disease detection [258]*.

Pastor *et al.* [259] have presented a system for distributed computing in agriculture that includes three layers: things (i.e., the individual sensors and subsystems), edge (responsible for the control of subsystems), and fog (providing local storage and analytics). Tasks carried out at the local edge and fog layer include data filtering, classification and detection of events. The primary reason for carrying out those tasks at nearby layers is latency, as the system tries to optimize itself in real-time. Another example of a layered system can be found in [260]*, where Raspberry Pi computers are deployed as sensors in the environment and on animals to monitor temperature and movement. This data is processed at an edge layer at the farm itself, whereas cloud infrastructure is used to collect long-term statistics. Besides a strong focus on the communication technologies at the edge layer, the work of Ahmed *et al.* [257] proposes a hierarchical structure of fog nodes and suggests to carry out computations at local gateways. As one of only a few papers, it also considers the energy aspect of this local processing. However, their main arguments for edge processing are a reduced latency and bandwidth limitations.

Wolfert *et al.* [261] see big data analytics as a major disruption for farming and have identified real-time analytics of agricultural data as a key challenge. They emphasize the issue of data quality, i.e., errors in the raw data make operations such as denoising and transformation necessary before further processing. From the perspective of saving upstream bandwidth, edge deployments are beneficial in such use cases. Similarly, Ivanov *et al.* [262] have observed that many of the sensor data gathered in smart farming contain redundancies that need to be fused before being pushed to a centralized entity.

While a number of edge-enabled systems exist, we observe that most are closed, hybrid deployment, i.e., the intermediate edge layer is deployed on-premise. We believe that collaboration among different sites would bring edge computing to its full potential. For instance, we can imagine weather data to be shared. Similarly, sensor readings from farms with different characteristics could help researchers to develop more resistant crops.

D. HUMAN AUGMENTATION

Human augmentation is the process of improving the well-being and capabilities of humans. This augmentation can be done by oneself, e.g., through quantifying, analyzing, and subsequently influencing one's behavior (Section IV-D1), or in the context of healthcare-related applications (Section IV-D2). Furthermore, Section IV-D3 shows how human cognition can be augmented or assisted.

Privacy and data security are naturally critical concerns in these types of applications due to the intimate nature of the data. As shown by Fereidooni *et al.* [263], today's cloud-based services fail to provide data integrity, authenticity, and confidentiality. However, those factors are critical for the acceptance of such services. Besides the trustworthiness of the nodes that store or process the data, fine-grained access control policies for the remote access and forwarding (e.g., a physician can forward a patient's data to a pharmacy) should also be implemented. To realize this, we can image a network of federated, trusted edge nodes across different organizations.

1) QUANTIFIED SELF

With new affordable personal devices, people have gained an interest in collecting and analyzing data about their own body and behaviors. This concept is commonly referred to as the *quantified self* [264]. Besides getting a deeper understanding of oneself, this data can also be useful in many health-related aspects, e.g., for personalized medicine or preventive medicine [265]. Users are often interested in aggregate values, such as the total walking distance for a single day. Such aggregation can be performed at the edge. If the users wish to rely on cloud services, only those aggregated values are sent to the cloud. Aggregating and storing data is an important use case for personal fitness trackers that count steps, monitor one's heart rate or analyze sleep patterns. This type of wearable fitness technology is a big part of the quantified self community today [266].

Schmidt *et al.* [267]* present a digital fitness coach to support individuals in achieving fitness goals. The system generates training plans and is able to adapt them, e.g., depending on a user's movements. Among other data, data from tracking devices is used. Bajpai *et al.* [268]* use heart-rate readings from wearable sensors to track physical activity, map the activity to calorie consumption, and estimate the cardio-respiratory fitness of a person.

2) PRECISION MEDICINE

The umbrella term *E-Health* describes applications that make use of information systems in order to improve people's treatments and overall health. The concept of *connected health* [8] describes how different actors (e.g., patients, hospitals, physicians) are linked in order to improve their services. In this section, we summarize the above concepts as *preci-*

sion medicine. It has been noted that healthcare is one of the killer applications for future information technologies [269] and that cloud-based health care can help to reduce the overall costs of healthcare [270]. For a detailed survey of healthcare-related applications, we refer the reader to Kraemer *et al.* [271]. Bui and Zorzi [269] have identified three requirements for healthcare applications: (i) interoperability, (ii) reliability and bounded latency, and (iii) privacy, authentication, and integrity. Edge computing can help with the latter two. Latency is especially critical if the application is tied to a cognitive process or a time-critical control loop (see Section IV-D3).

By monitoring parameters of a patient and combining information from health sensors with other ambient sensors, health-related issues can be detected. One example is fall detection for stroke patients [272]*. To perform these tasks, often large amounts of raw data have to be analyzed or complex features need to be extracted [273]. Often the monitoring is part of a sense-process-actuate loop, i.e., whenever an event or anomaly is detected in the monitoring phase, a (timely) action has to be taken. For telesurgery, latencies below 200 ms are optimal [274]*—a constraint that can be challenging when relying on distant clouds. For other applications such as ECG monitoring, delays in the order of several seconds might be acceptable [275]*. For less critical parameters, storing aggregate values for later retrieval is sufficient. Amraoui and Sethom [276]* have presented an architecture for patient monitoring in Wireless Body Area Networks (WBAN) that makes uses of cloudlets for close-by processing of sensor data. Hybrid edge cloud systems also exist, e.g., Althebyan *et al.* [103] have presented a scalable health monitoring system that uses both cloud and edge computing. Similarly, Azimi *et al.* [104] have developed a 3-tier system, in which tasks are partitioned among the tiers. For example, the training procedure for machine learning algorithms is carried out in the cloud, whereas the resulting classifiers are deployed closer to the sensors.

3) COGNITIVE ASSISTANCE

The idea of cognitive assistance comes from the vision of augmenting human cognition through computing systems [277]. These types of applications are, for example, useful to assist the elderly who suffer from deteriorating senses or for patients with neurological diseases, such as Alzheimer's. Applications that aim to assist or substitute the cognitive tasks of humans should preferably not be slower than humans. This is challenging, given that for instance humans recognize familiar faces in the order of a few hundred milliseconds [278]*. Even more challenging, recognizing human voices takes 4 ms [279]*. These tasks are also computationally intensive and hence require to offload the processing. Ha *et al.* [280]* have presented a system for wearable cognitive assistance. The system uses Google Glass to capture live video and performs real-time scene interpretation using different components, such as activity inference, face recognition, and motion classifiers. Chen *et al.* [4] have presented a study that investigates the latency reduction for wearable cognitive assistance.

E. SUMMARY

To conclude this section, Table 2 summarizes prominent use cases from the previous subsections and shows how the promised benefits of edge computing (see Section III-A) are applicable to them. For this, we use the following semantics:

++ Edge computing is absolutely necessary to ensure the requirements and these cannot be fulfilled by relying on the cloud. Furthermore, local processing cannot ensure the expected quality of experience.

\+ Edge computing is beneficial and improves the quality of the service and/or its experience for the users.

∘ The benefit of edge computing heavily depends on the concrete scenario and context in which the application operates.

– Edge computing brings no real-world benefit or the attribute is not relevant for the application. For example, even though edge computing might improve the latency in absolute numbers, this might not be critical in applications where the computation or actuation takes far longer than the communication.

The last four columns of the table indicate which of the components are used by that use case.

To conclude this section, we now summarize our main findings and observations:

- **Diverse objectives.** The motivations to use edge computing are very diverse. Consequently, most use cases only profit from a subset of the potential benefits. The heterogeneity of objectives and use cases means that there is no ''one size fits all'' solution when it comes to the question if an application should be moved to the edge.
- **Not indispensable for most applications.** While most of the presented applications can benefit from one or more aspects of edge computing, resulting in higher QoS or QoE, few applications cannot fundamentally function without edge computing. Thus, economical considerations are important to determine if edge computing is sensible for a given use case. The fact that there are no established business model and ubiquitous infrastructure for edge computing yet prevents most of these applications from being moved to the edge today. However, we believe once edge computing infrastructures are widely available, a large number of applications will use it.
- **Killer apps do exist.** We could identify some applications for which edge computing is indispensable, either regarding latency (e.g., rendering for AR/VR) or bandwidth (e.g., the processing of several video streams or LIDAR data). Furthermore, edge computing can provide an emergency communication and computing infrastructure, thus creating a more resilient overall public infrastructure.

TABLE 2. Systematic overview of surveyed use cases.

	Latency	Bandwidth	Energy	Privacy	Consolidation	Filtering	Storage	Offloading
Mobile Device Augmentation								
Gaming - Scene rendering [5], [144]	++	+	+	−				✓
Gaming - Collaboration of neighboring players [143], [146]	++	+	+	−	✓		✓	✓
AR/VR - Rendering [159]	++	++	+	−		✓		✓
AR/VR - Hybrid rendering [158]	++	+	+	−		✓	✓	✓
AR/VR - Reconstruction of 3D maps [161]	+	++	+	−	✓	✓	✓	✓
Content delivery - Video streams [168]	−	+	−	−		✓	✓	
Content delivery - Website delivery [170]	−	○	−	−			✓	
Content delivery - Applications and updates [177], [178]	−	+	−	−			✓	
Content delivery - Collaborative caching [173], [174]	+	−	○	○		✓	✓	
Storage - Storage for edge analytics [176]	+	+	○	○	✓	✓	✓	✓
Storage - Reverse CDN [10], [56], [179], [180]	○	+	○	+			✓	
Storage - Document synchronization [58]	−	+	−	○	✓		✓	
Storage - Personal data storage [182]–[184]	○	○	○	+			✓	
Infrastructure Augmentation								
Smart grids - Monitoring and control [196]	○	+	○	+	✓	✓	✓	
Smart grids - Scheduling distributed energy resources [198]	+	○	−	○	✓			✓
Traffic & transportation - Adaptive traffic light [200]	−	○	−	+	✓	✓		✓
Traffic & transportation - Detection of road hazards [201]	○	○	−	−	✓	✓		✓
Traffic & transportation - Traffic planning [202]–[204]	−	○	−	○	✓	✓	✓	
Traffic & transportation - Emergency vehicle route clearance [39]	−	+	−	−	✓			
Autonomous driving - Disseminating data to vehicle [209], [212]	++	+	−	○	✓	✓		
Autonomous driving - Processing LIDAR data [213]	+	++	−	−		✓		✓
Environment - Pollution monitoring [216], [217]	−	○	−	−	✓	✓	✓	
Environment - Pollution monitoring via crowdsensing [189], [218], [220], [224]	−	+	+	+	✓	✓	✓	
Environment - Optimize garbage collection [226]–[229]	−	−	−	−	✓	✓	✓	
Emergency response - Emergency notification [231]	−	○	−	−		✓		
Emergency response - Situation awareness/mobile command and control [230]	○	+	−	○	✓	✓	✓	
Emergency response - Ad-hoc communication in disaster scenarios [233], [234]	++	++	−	−	✓	✓	✓	
Surveillance - Vehicle tracking [236]	○	+	−	+		✓		✓
Surveillance - Just-in-time video indexing [203]	+	++	○	+		✓	✓	✓
Surveillance - Biometric identification [239]	−	○	○	+			✓	✓
IoT Device Augmentation								
Smart home/building - Video surveillance [243], [245]	−	++	−	+		✓	✓	
Smart home/building - Coordination of building subsystems [248]	○	○	−	+	✓	✓		
IIoT - Production process analysis [250]–[252]	−	○	○	+	✓	✓	✓	
IIoT - Machine condition monitoring [253], [254]	○	+	−	+	✓	✓		
IIoT - Warehouse logistics scheduling [255]	○	+	−	+	✓	✓		
IIoT - Dynamic production line scheduling [249]	+	○	−	+	✓			✓
Agriculture & farming - Monitoring plants/lifestock [257], [260]	−	++	+	○	✓	✓		
Agriculture & farming - Yield prediction [258]	−	○	−	+	✓		✓	✓
Human Augmentation								
Quantified self - Analyzing fitness tracker data [267], [268]	−	○	+	+		✓	✓	✓
Precision medicine - Fall detection [272]	−	+	+	+		✓		✓
Precision medicine - Patient monitoring with WBAN [276]	○	+	○	+	✓	✓	✓	✓
Precision medicine - Remote surgery [274]	+	+	−	○		✓		
Precision medicine - Analyzing ECG features [275]	−	○	−	+		✓	✓	✓
Cognitive assistance - Face recognition [278]	+	+	○	+				✓
Cognitive assistance - Speech recognition [279]	++	○	○	+				✓
Cognitive assistance - Wearable cognitive assistance [280]	++	+	+	+	✓	✓		✓

- **Cross-layer applications design.** Many works propose layered system architectures, where applications are split between the layers. While the number and naming of the layers differ (the common ones being: local, edge, cloud, and sometimes intermediate fog layers), the common motivation is to exploit the favorable characteristics of each layer. How to efficiently partition applications across layers is still an ongoing field of research.
- **Offloading objectives.** There are two common motivations for offloading. First, offloading can accelerate latency-critical computations on devices with low computing power. Latency reductions are especially useful for demanding real-time computations, especially for mobile gaming, AR, VR, and autonomous driving. The second motivation for offloading is to save energy on battery-constrained devices. Thus, energy benefits

are usually only viewed from the perspective of the end devices and the effect on the overall energy footprint remains unclear.

- **Bandwidth is critical for infrastructure and IoT devices.** Infrastructure and IoT applications tend to have little demand for offloading. Nearly all of our surveyed applications in these two categories implement the consolidation and filtering components. Compared to the cloud, edge computing can achieve large bandwidth savings for applications that process big ephemeral data. Thus, while latency tends to be only a minor issue in those use cases, infrastructure and IoT devices can profit significantly from edge computing.
- **Privacy has great potential.** There is a clear division between applications where the privacy-protecting aspect of edge computing is relevant and those where it is not. The privacy aspect can be of tremendous relevance for sensor data that contains trade secrets or sensitive personal data. We see great untapped potential for research in this direction to fully exploit the privacy benefits of edge computing.

V. EXISTING PRODUCTS & INDUSTRY SOLUTIONS

In this section, we elaborate on notable industry developments in edge computing. We do not intend to deliver an exhaustive review of all available industry solutions. Instead, this section serves to illustrate the already existing commercial potential of edge computing, and to shed light on the different business models on the edge computing market.

A. EXTENDING CLOUD SERVICES TO THE EDGE

The providers of conventional cloud services focus on extensions of their cloud ecosystems to environments with limited connectivity and high data volumes. They offer software frameworks that can be used in places with insufficient or only sporadic connectivity while being tightly integrated with the existing cloud execution environment, offering seamless integration between software components in the cloud and at the edge.

Amazon's AWS Snowball Edge[27] is a mobile computer with very large storage capacity in a suitcase form factor. Its main use case is the collection and pre-processing of large amounts of data in places with limited or no Internet access. The capability to execute jobs with the same API as in the cloud offers the possibility to pre-filter or verify data at the place of collection within the familiar execution environment of AWS. Building on its content delivery network, Amazon offers Lambda@Edge[28], a service for serverless computing that is distributed throughout the CDN. The intended use cases are low-latency web services, for example, real-time image transformation. Amazon also offers AWS IoT Greengrass[29], which is a software framework that lets customers execute AWS Lambda functions on-premises, while still being managed by and synchronized to the AWS data centers. This could serve to pre-filter IoT data and only transmit relevant data to the cloud, while avoiding service interruptions due to a temporarily interrupted Internet connection.

Similarly, Microsoft offers Azure IoT Edge[30], a software framework designed to offload Azure tasks that were designed for execution in the cloud. It allows to pre-process/filter data, and act with low-latency, while being tightly integrated with the Azure IoT management framework, allowing developers to develop software on the cloud (e.g., training a classifier for image recognition) and deploy it at the edge. There are also several open source projects targeted at edge computing, such as Apache Edgent[31], which is a framework to analyze and filter data streams at the edge. EdgeX Foundry[32] is a vendor-neutral open source project that aims to provide a common framework for IoT devices and services. It is designed as a collection of microservices that can be executed at the edge to perform edge analytics and control tasks. StarlingX[33] is an open source project that implements a cloud infrastructure software stack for edge clouds.

B. HARDWARE FOR EDGE COMPUTING

While cloud providers mainly offer solutions for the integration of edge applications into their cloud ecosystems, hardware manufacturers offer devices that are designed for typical deployment environments of edge computing. One example is the aforementioned Amazon AWS Snowball Edge. As data collection is its primary use case, it is equipped with various security features, including resistance against mechanical shock. Another example is Cisco, which offers a general-purpose Industrial Compute Gateway (IC3000[34]), designed to be deployed outside climate-controlled data center conditions. One intended use case is traffic monitoring inside a roadside cabinet. The device is housed in a compact, ruggedized case, includes tamper-proofing technology, and has a strong focus on end-to-end security. Google offers Edge TPU[35], which is a hardware accelerator for neural network inference. It is targeted at edge analytics to exploit the benefits of low-latency access networks with low-latency inference at the edge.

C. MOBILE EDGE COMPUTING

A different approach to edge computing is driven by the telecommunications industry, which focuses on mobile edge computing to provide edge-services to mobile devices. All

[27]https://aws.amazon.com/snowball-edge/ (Accessed: 2019-08-09)
[28]https://aws.amazon.com/lambda/edge/ (Accessed: 2019-08-09)
[29]https://aws.amazon.com/greengrass/ (Accessed: 2019-08-09)

[30]https://azure.microsoft.com/en-us/services/iot-edge/ (Accessed: 2019-08-09)
[31]https://edgent.apache.org/ (Accessed: 2019-09-05)
[32]https://www.edgexfoundry.org/ (Accessed: 2019-08-09)
[33]https://www.starlingx.io/ (Accessed: 2019-09-05)
[34]https://cisco.com/c/en/us/solutions/internet-of-things/ic3000-industrial-compute-gateway.html (Accessed: 2019-08-09)
[35]https://cloud.google.com/edge-tpu/ (Accessed: 2019-08-09)

major manufacturers of telecommunications equipment offer solutions for CloudRAN[36] [37] [38] (a.k.a C-RAN / vRAN), which describes the virtualized management of a group of mobile base stations. Such locally centralized management allows tight coordination of the radio access network (RAN). This application requires extremely low latency to such an extent that the RAN-managers must be in extremely close proximity to each other. As recognized by the ETSI-MEC industry specification group[39], the widespread deployment of virtualized execution environments at the very edge of the RAN offers the potential to be extended to hosting mobile cloud computing services.

MobiledgeX[40] offers ETSI-MEC-compliant management and orchestration infrastructure to integrate cloudlet-based edge computing into the NFV-infrastructure of mobile network operators, with a first integration into the network of Deutsche Telekom. Besides offering low latency, a major selling argument is the ability to verify user locations, which can be important for applications that are subject to location spoofing attacks, such as geographic games, ridesharing or delivery services. Niantic, the maker of Pokemon Go, entered a cooperation with MobiledgeX and Deutsche Telekom[41] with regards to their latest AR game to verify user locations and provide higher QoS through flexible provisioning of local servers. The VaporIO[42] Kinetic Edge project offers colocation of micro data centers at the wireless infrastructure. Potential customers include both mobile network operators (e.g., if they want to deploy network functions) and application providers, e.g., for the deployment of IoT gateways. Live sites have already been deployed in Chicago with the intention to cover an additional 20 cities in the US by the end of 2020. Similar efforts are undertaken by Saguna Networks[43], who offer a MEC solution that is compliant to the proposed ETSI standard.

D. EXAMPLES OF APPLIED EDGE COMPUTING

Besides general-purpose provisioning of edge computing capacities, specific edge-based services were developed. One such application is Grid Edge Control from Varentec[44]. The system performs voltage regulation at the edge in real-time, based on local measurements and monitoring. The goal is to meet demands more efficiently while minimizing the amount of data that has to be sent via expensive wireless links. Several market actors have presented visions of the connected stadium[45] [46] [47] [48]. At events, fans expect reliable connectivity that enables access to extended statistics about the event, greater integration with social media and customized experiences, e.g., through different video streams that can be selected. China Mobile and Nokia deployed Nokia's Edge Video Orchestration together with an ultra-dense heterogeneous RAN to provide attendees of a racing event with low-latency live video feeds of the track[49]. The German Federal Ministry of Transport and Digital Infrastructure has established the Digital Motorway Test Bed[50] for research and development in the field of autonomous driving. In addition to high-speed wireless connectivity alongside the road, this environment provides additional infrastructure, such as sensors embedded in crash barriers. The test bed is targeted at various stakeholders such as automotive and telecommunications companies.

VI. FUTURE RESEARCH CHALLENGES

A. SECURITY

Security challenges in edge computing can be divided into basic security (i.e., challenges resulting from the pure distribution, which are similar to the challenges in cloud computing) and advanced security (i.e., challenges resulting from the lightweight distribution to the edge).

1) BASIC SECURITY

The basic security challenges in edge computing are similar to those in cloud computing. The distribution of data or computation comes at the cost of reduced control over the executing machine. In conventional self-owned environments, users can simply enforce security mechanisms by using the appropriate tools. In contrast, in edge computing, the user has to rely on and trust the execution environment of the edge nodes to provide the desired security mechanisms.

2) ADVANCED SECURITY

Assuming that appropriate basic security measures are in place, edge computing increases the attack surface considerably. First, additional entities are involved. As such, an attacker can attack both entities, i.e., edge nodes, and the network itself. Authentication and integrity need to be considered with increased importance, especially if critical data, functions, or infrastructures are considered. Varghese

[36]https://www.nokia.com/networks/solutions/airscale-cloud-ran/ (Accessed: 2019-08-09)
[37]https://carrier.huawei.com/en/solutions/all-cloud-network-towards-5g/cloudran (Accessed: 2019-08-09)
[38]https://www.ericsson.com/en/networks/offerings/5g/5g-vran (Accessed: 2019-08-09)
[39]https://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp23_MEC_and_CRAN_ed1_FINAL.pdf (Accessed: 2019-08-09)
[40]https://mobiledgex.com/ (Accessed: 2019-08-09)
[41]https://www.telekom.com/de/medien/medieninformationen/detail/deutsche-telekom-niantic-und-mobiledgex-partnerschaft-545524 (Accessed: 2019-08-09, German)
[42]https://vapor.io/ (Accessed: 2019-08-09)
[43]https://www.saguna.net/ (Accessed: 2019-08-09)
[44]https://varentec.com (Accessed: 2019-08-09)
[45]https://tmt.knect365.com/connected-stadium-summit/ (Accessed: 2019-08-09)
[46]https://cisco.com/c/en/us/solutions/industries/sports-entertainment/connected-stadium.html (Accessed: 2019-08-09)
[47]https://sap.com/assetdetail/2016/05/f2d235ad-717c-0010-82c7-eda71af511fa.html (Accessed: 2019-08-09)
[48]https://ericsson.com/en/networks/trending/hot-topics/connected-stadium (Accessed: 2019-08-09)
[49]https://builders.intel.com/docs/networkbuilders/Real-world-impact-of-mobile-edge-computing-MEC.pdf (Accessed: 2019-08-09)
[50]https://bmvi.de/SharedDocs/EN/Articles/DG/digital-motorway-test-bed.html (Accessed: 2019-08-09)

and Buyya [281] recommend further research in malware detection and classification, and intrusion-detection for real-time bandwidth-limited environments (e.g., IoT).

Edge nodes typically utilize lightweight virtualization, e.g., container-based virtualization without hypervisor. The security implications of these variations of virtualization are being discussed within the research community [282], [283].

Beyond these software-based security measures, the high distribution of edge sites increases the risk of *physical* attacks. Conventional data centers (and self-owned environments) can be comparatively easy to be secured against physical attacks, while edge sites may be located in roadside cabinets or unguarded buildings. Here, theft, sabotage, or tampering with the edge nodes may impact not only the physical edge nodes, but may also affect the security and integrity of performed computations, as well as lead to theft of personal or security-relevant data. Ensuring *physical* integrity and security of edge sites demands further research.

Considering the aftermath of a successful attack, Wang *et al.* [284] identify different forensics for fog and cloud computing, and discuss future challenges in the domain of forensics.

B. PRIVACY AND TRUST

Publicly available edge computing services open up new opportunities and challenges w.r.t. privacy and trust, which starts to gain attention in the community [285], [286]. While they enable tracking of users that, e.g., prefer the closest edge instance, these edge computing-services can prevent misuse, e.g., by location spoofing. Regarding privacy, the major questions to ask are:

- (*User perspective:*) Can an edge node be used to offload computations without revealing sensitive information?
- (*Provider perspective:*) Can an edge node execute some computations without revealing sensitive information, e.g., about themselves or about other computations?

Similar, yet more abstract questions follow from the perspective of trust:

- (*User perspective:*) Can an edge node be trusted to perform some offloaded computation according to an agreement?
- (*Provider perspective:*) Can an edge node trust that an offloaded computation will not harm other computations or the node itself?

Privacy and trust in edge computing depend on the application scenario, the actual requirements and the options to attest the fulfillment of these requirements, including potential sanctions. There is the potential for edge computing to significantly improve privacy by avoiding the dissemination of data toward some central entities and service providers. However, to tap into this potential, the question of trustworthiness has to be resolved as well. Future research efforts in the field of edge computing should encompass the following challenges:

- Achieving strong privacy protection for users of edge computing services while allowing for (maybe even anonymous) attestation of offloaded computations *before* their execution.
- Attesting offloaded computations w.r.t. their trust- worthiness without nullifying the efficiency gains of edge computing.

C. MULTI-ACCESS EDGE COMPUTING

Service placement in multi-access edge computing (a recent renaming of mobile edge computing by the ETSI-MEC industry specification group[51] to emphasize the inclusion of access technologies other than cellular networks) also raises some open questions. As indicated in Section V, there are already numerous commercial solutions for edge computing, especially for IoT, most of which are advertised for stationary in-house deployment (i.e., hybrid cloud). However, if network topology and usage patterns vary over time, providing reliable edge services can become challenging. Changes in a user's access point, caused by user mobility or network congestion, can dramatically affect the number of hops between the user and their services, especially if such a transition occurs across network boundaries. Such network changes require rapid and dynamic migration or replication of edge services.

This in turn raises the question of discovering edge sites and managing those. Discovery mechanisms that rely on central repositories (e.g., as in [50]) clearly do not scale. The same is true for the control and management of the infrastructure (e.g., as in [110]). Unified and scalable discovery and management become even more challenging if we consider edge sites owned by different stakeholders. Furthermore, the dynamic edge environment—edge nodes may join or leave spontaneously—prohibits a static dissemination of information about available infrastructure that serves as a basis for placement decisions. Further research is needed to ensure that dynamic service placement strategies can quickly and efficiently mitigate potential QoS losses caused by dynamic network changes [287]. Other important open questions are how services with different requirements are placed w.r.t. fairness, and how the migration of services can be proactively coordinated with access point changes to minimize service interruptions.

D. BUSINESS MODELS

As of today, it still remains unclear what the dominant business model of edge computing will be, despite the fact that it offers new revenue opportunities for owners of edge computing sites. To bring edge computing to its full potential, cooperation between all network and cloud providers is required to enable seamless transitions of services across network boundaries, e.g., from a landline-based WiFi access point to a 5G base station. Ahmed *et al.* [288] argue that edge

[51] https://www.etsi.org/technologies/multi-access-edge-computing (Accessed: 2019-09-06)

computing needs the same flexibility of pay-as-you-go models as cloud computing. As mentioned before, MobiledgeX has shown ambitions to provide a common API for mobile edge computing by renting resources in multiple mobile networks. Nonetheless, distributed ownership will make a unified marketplace with the same standards difficult [281]. New business models, technical standards, and accounting mechanisms need to be developed to enable a fair, seamless experience similar to roaming in mobile networks [289].

If we include opportunistic devices (e.g., owned by end users), the question of business models has to be extended to incentive mechanisms. As an example, there is previous work on initiatives to share one's broadband connection via WiFi [290], [291]. Given the additional resources available on those gateways or in local networks attached to them, we could envision sharing of resources for computations as the next step. Very few recent works have started to investigate incentive mechanisms for offloading in the context of edge computing [292], [293]. However, more in-depth practical studies are required to validate the theoretical results.

VII. CONCLUSION

In this article, we have given a snapshot of the current state of edge computing. We clarified the terminology and characteristics for edge computing and its related concepts and put them into relation with the current computing landscape. Our main contributions are a structured survey of proposed edge computing applications as well as a systematic analysis which benefits they can reap from edge computing and which basic components they have in common (Section IV-E). We found that while most of those applications could exist without edge computing, many could profit tremendously in one way or another. We could also identify "killer applications" that are likely to strongly push the demand for edge computing infrastructure in the near future. To relate the surveyed academic efforts in edge computing to industry solutions, we also included a brief overview of current industry developments. This article concluded with an outline of future research challenges that we believe need to be addressed to realize the full potential of edge computing.

REFERENCES

[1] W. Hu, Y. Gao, K. Ha, J. Wang, B. Amos, Z. Chen, P. Pillai, and M. Satyanarayanan, "Quantifying the impact of edge computing on mobile applications," in *Proc. 7th ACM SIGOPS Asia–Pacific Workshop Syst. (APSys)*, 2016, pp. 5:1–5:8.

[2] A. Vulimiri, C. Curino, P. B. Godfrey, T. Jungblut, J. Padhye, and G. Varghese, "Global analytics in the face of bandwidth and regulatory constraints," in *Proc. 12th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2015, pp. 323–336.

[3] S. Yi, Z. Hao, Q. Zhang, Q. Zhang, W. Shi, and Q. Li, "LAVEA: Latency-aware video analytics on edge computing platform," in *Proc. 37th Int. Conf. Distrib. Comput. Syst.*, Jun. 2017, pp. 2573–2574.

[4] Z. Chen, W. Hu, J. Wang, S. Zhao, B. Amos, G. Wu, K. Ha, K. Elgazzar, P. Pillai, R. L. Klatzky, D. P. Siewiorek, and M. Satyanarayanan, "An empirical study of latency in an emerging class of edge computing applications for wearable cognitive assistance," in *Proc. 2nd ACM/IEEE Symp. Edge Comput. (SEC)*, 2017, pp. 14:1–14:14.

[5] Y. Lin and H. Shen, "CloudFog: Leveraging fog to extend cloud gaming for thin-client MMOG with high quality of service," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 2, pp. 431–445, Feb. 2017.

[6] E.-K. Lee, M. Gerla, G. Pau, U. Lee, and J.-H. Lim, "Internet of vehicles: From intelligent grid to autonomous cars and vehicular fogs," *Int. J. Distrib. Sensor Netw.*, vol. 12, no. 9, pp. 1–14, 2016.

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

[8] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet Things J.*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[9] A. Chandra, J. Weissman, and B. Heintz, "Decentralized edge clouds," *IEEE Internet Comput.*, vol. 17, no. 5, pp. 70–73, Sep./Oct. 2013.

[10] J. Gedeon, N. Himmelmann, P. Felka, F. Herrlich, M. Stein, and M. Mühlhäuser, "vStore: A context-aware framework for mobile micro-storage at the edge," in *Proc. 9th Int. Conf. Mobile Comput., Appl. Services (MobiCASE)*, 2018, pp. 165–182.

[11] H. Chang, A. Hari, S. Mukherjee, and T. V. Lakshman, "Bringing the cloud to the edge," in *Proc. IEEE INFOCOM Workshops*, Apr. 2014, pp. 346–351.

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

[13] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai, "Cloudlets: At the leading edge of mobile-cloud convergence," in *Proc. 6th Int. Conf. Mobile Comput., Appl. Services (MobiCASE)*, 2014, pp. 1–9.

[14] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. 3rd IEEE Workshop Hot Topics Web Syst. Technol. (HotWeb)*, Nov. 2015, pp. 73–78.

[15] F. Bonomi, R. A. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the Internet of Things," in *Proc. 1st Workshop Mobile Cloud Comput. (MCC)*, 2012, pp. 13–16.

[16] J. Gedeon, J. Heuschkel, L. Wang, and M. Mühlhäuser, "Fog computing: Current research and future challenges," in *Proc. GI/ITG KuVS Fachgespräche Fog Comput.*, 2018, pp. 1–4.

[17] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *J. Syst. Archit.*, vol. 98, pp. 289–330, Sep. 2019.

[18] Z. Pang, L. Sun, Z. Wang, E. Tian, and S. Yang, "A survey of cloudlet based mobile computing," in *Proc. Int. Conf. Cloud Comput. Big Data (CCBD)*, Nov. 2015, pp. 268–275.

[19] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. Workshop Mobile Big Data (Mobidata)*, 2015, pp. 37–42.

[20] R. Mahmud, R. Kotagiri, and R. Buyya, "Fog computing: A taxonomy, survey and future directions," in *Internet of Everything: Algorithms, Methodologies, Technologies and Perspectives*, B. Di Martino, K.-C. Li, L. T. Yang, and A. Esposito, Eds. Singapore: Springer, 2018, pp. 103–130.

[21] C.-H. Hong and B. Varghese, "Resource management in fog/edge computing: A survey," 2018, *arXiv:1810.00305*. [Online]. Available: https://arxiv.org/abs/1810.00305

[22] T. H. Luan, L. Gao, Z. Li, Y. Xiang, G. We, and L. Sun, "A view of fog computing from networking perspective," 2016, *arXiv:1602.01509*. [Online]. Available: https://arxiv.org/abs/1602.01509

[23] R. Roman, J. López, and M. Mambo, "Mobile edge computing, fog et al.: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018.

[24] S. Yi, Z. Qin, and Q. Li, "Security and privacy issues of fog computing: A survey," in *Proc. 10th Int. Conf. Wireless Algorithms, Syst., Appl. (WASA)*, 2015, pp. 685–695.

[25] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An overview of fog computing and its security issues," *Concurrency Comput., Pract. Exper.*, vol. 28, no. 10, pp. 2991–3005, 2016.

[26] 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 Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2359–2391, 4th Quart., 2017.

[27] C. Li, Y. Xue, J. Wang, W. Zhang, and T. Li, "Edge-oriented computing paradigms: A survey on architecture design and system management," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 39:1–39:34, Apr. 2018.

[28] N. Hassan, S. Gillani, E. Ahmed, I. Ibrar, and M. Imran, "The role of edge computing in Internet of Things," *IEEE Commun. Mag.*, vol. 56, no. 11, pp. 110–115, Nov. 2018.

[29] M. Heck, J. Edinger, D. Schäfer, and C. Becker, "IoT applications in fog and edge computing: Where are we and where are we going?" in *Proc. 27th Int. Conf. Comput. Commun. Netw. (ICCCN) Workshops*, 2018, pp. 1–6.

[30] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the Internet of Things," *IEEE Access*, vol. 6, pp. 6900–6919, 2018.

[31] C. Perera, Y. Qin, J. C. Estrella, S. Reiff-Marganiec, and A. V. Vasilakos, "Fog computing for sustainable smart cities: A survey," *ACM Comput. Surv.*, vol. 50, no. 3, p. 32, Oct. 2017.

[32] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile edge computing potential in making cities smarter," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 38–43, Mar. 2017.

[33] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 33–39, Aug. 2018.

[34] C. Puliafito, E. Mingozzi, F. Longo, A. Puliafito, and O. Rana, "Fog computing for the Internet of Things: A survey," *ACM Trans. Internet Technol.*, vol. 19, no. 2, p. 18, 2019.

[35] M. T. Beck, M. Werner, S. Feld, and T. Schimper, "Mobile edge computing: A taxonomy," in *Proc. 6th Int. Conf. Adv. Future Internet (AFIN)*, 2014, pp. 48–54.

[36] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2322–2358, 4th Quart., 2017.

[37] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, "Mobile edge computing: A survey," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, Feb. 2018.

[38] Y. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing—A key technology towards 5G," ETSI, Sophia Antipolis, France, White Paper 11, 2015, pp. 1–16.

[39] S. Nunna, A. Kousaridas, M. Ibrahim, M. Dillinger, C. Thuemmler, H. Feussner, and A. Schneider, "Enabling real-time context-aware collaboration through 5G and mobile edge computing," in *Proc. 12th Int. Conf. Inf. Technol. New Gener.*, 2015, pp. 601–605.

[40] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, 3rd Quart., 2017.

[41] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Gener. Comput. Syst.*, vol. 29, no. 1, pp. 84–106, 2013.

[42] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: Making smartphones last longer with code offload," in *Proc. ACM Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2010, pp. 49–62.

[43] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. 6th Eur. Conf. Comput. Syst. (EuroSys)*, 2011, pp. 301–314.

[44] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Mar. 2012, pp. 945–953.

[45] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, "Cuckoo: A computation offloading framework for smartphones," in *Proc. 2nd Int. Conf. Mobile Comput., Appl. Services (MobiCASE)*, 2010, pp. 59–79.

[46] M. Villari, M. Fazio, S. Dustdar, O. Rana, and R. Ranjan, "Osmotic computing: A new paradigm for edge/cloud integration," *IEEE Cloud Comput.*, vol. 3, no. 6, pp. 76–83, Nov./Dec. 2016.

[47] E. Marín-Tordera, X. Masip-Bruin, J. García-Almiñana, A. Jukan, G.-J. Ren, and J. Zhu, "Do we all really know what a fog node is? Current trends towards an open definition," *Comput. Commun.*, vol. 109, pp. 117–130, Sep. 2017.

[48] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 5, pp. 27–32, Oct. 2014.

[49] J. S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis, "The benefits of self-awareness and attention in fog and mist computing," *Computer*, vol. 48, no. 7, pp. 37–45, Jul. 2015.

[50] J. Gedeon, C. Meurisch, D. Bhat, M. Stein, L. Wang, and M. Mühlhäuser, "Router-based brokering for surrogate discovery in edge computing," in *Proc. 37th IEEE Int. Conf. Distrib. Comput. Syst. Workshops (ICDCSW)*, Jun. 2017, pp. 145–150.

[51] M. B. Younes and A. Boukerche, "An intelligent traffic light scheduling algorithm through VANETs," in *Proc. 39th Annu. IEEE Conf. Local Comput. Netw. Workshops*, Sep. 2014, pp. 637–642.

[52] J. Wang, Z. Feng, Z. Chen, S. George, M. Bala, P. Pillai, S. Yang, and M. Satyanarayanan, "Bandwidth-efficient live video analytics for drones via edge computing," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 159–173.

[53] E. Fasolo, M. Rossi, J. Widmer, and M. Zorzi, "In-network aggregation techniques for wireless sensor networks: A survey," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 70–87, Apr. 2007.

[54] R. Mayer, H. Gupta, E. Saurez, and U. Ramachandran, "FogStore: Toward a distributed data store for fog computing," in *Proc. IEEE Fog World Congr.*, Oct./Nov. 2017, pp. 1–6.

[55] F. Zhang, C. Xu, Y. Zhang, K. K. Ramakrishnan, S. Mukherjee, R. Yates, and T. Nguyen, "EdgeBuffer: Caching and prefetching content at the edge in the MobilityFirst future Internet architecture," in *Proc. 16th IEEE Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2015, pp. 1–9.

[56] I. Psaras, O. Ascigil, S. Rene, G. Pavlou, A. Afanasyev, and L. Zhang, "Mobile data repositories at the edge," in *Proc. USENIX Workshop Hot Topics Edge Comput. (HotEdge)*, 2018, pp. 1–6.

[57] S. K. Monga, S. K. R, and Y. Simmhan, "ElfStore: A resilient data storage service for federated edge and fog resources," 2019, *arXiv:1905.08932*. [Online]. Available: https://arxiv.org/abs/1905.08932

[58] P. Hao, Y. Bai, X. Zhang, and Y. Zhang, "EdgeCourier: An edge-hosted personal service for low-bandwidth document synchronization in mobile cloud storage services," in *Proc. 2nd ACM/IEEE Symp. Edge Comput. (SEC)*, Oct. 2017, pp. 7:1–7:14.

[59] B. Varghese, N. Wang, D. S. Nikolopoulos, and R. Buyya, "Feasibility of fog computing," 2017, *arXiv:1701.05451*. [Online]. Available: https://arxiv.org/abs/1701.05451

[60] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, Apr. 2010.

[61] T. Rausch, C. Avasalcai, and S. Dustdar, "Portable energy-aware cluster-based edge computers," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 260–272.

[62] C. You, K. Huang, H. Chae, and B.-H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Trans. Wireless Commun.*, vol. 16, no. 3, pp. 1397–1411, Mar. 2017.

[63] Y. Nan, W. Li, W. Bao, F. C. Delicato, P. F. Pires, Y. Dou, and A. Y. Zomaya, "Adaptive energy-aware computation offloading for cloud of things systems," *IEEE Access*, vol. 5, pp. 23947–23957, 2017.

[64] J. Zhang, X. Hu, Z. Ning, E. C. Ngai, L. Zhou, J. Wei, J. Cheng, and B. Hu, "Energy-latency tradeoff for energy-aware offloading in mobile edge computing networks," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2633–2645, Aug. 2018.

[65] P. P. Jayaraman, J. B. Gomes, H. Nguyen, Z. S. Abdallah, S. Krishnaswamy, and A. B. Zaslavsky, "CARDAP: A scalable energy-efficient context aware distributed mobile data analytics platform for the fog," in *Proc. 18th East Eur. Conf. Adv. Databases Inf. Syst. (ADBIS)*, 2014, pp. 192–206.

[66] Y. Xiao and M. Krunz, "QoE and power efficiency tradeoff for fog computing networks with fog node cooperation," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–9.

[67] F. Jalali, K. Hinton, R. Ayre, T. Alpcan, and R. S. Tucker, "Fog computing may help to save energy in cloud computing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1728–1739, May 2016.

[68] A. Boukerche, S. Guan, and R. E. D. Grande, "Sustainable offloading in mobile cloud computing: Algorithmic design and implementation," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 11:1–11:37, 2019.

[69] N. Davies, N. Taft, M. Satyanarayanan, S. Clinch, and B. Amos, "Privacy mediators: Helping IoT cross the chasm," in *Proc. 17th Int. Workshop Mobile Comput. Syst. Appl. (HotMobile)*, 2016, pp. 39–44.

[70] E. Shi, T. H. Chan, E. G. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2011, pp. 1–17.

[71] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.

[72] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet Things J.*, vol. 4, no. 3, pp. 772–782, Jun. 2017.

[73] Q. Wang, D. Chen, N. Zhang, Z. Ding, and Z. Qin, "PCP: A privacy-preserving content-based publish–subscribe scheme with differential privacy in fog computing," *IEEE Access*, vol. 5, pp. 17962–17974, 2017.
[74] A. Alrawais, A. Alhothaily, C. Hu, and X. Cheng, "Fog computing for the Internet of Things: Security and privacy issues," *IEEE Internet Comput.*, vol. 21, no. 2, pp. 34–42, Mar./Apr. 2017.
[75] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: Research problems in data center networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, 2009.
[76] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar, "Security and privacy in fog computing: Challenges," *IEEE Access*, vol. 5, pp. 19293–19304, 2017.
[77] I. Stojmenovic and S. Wen, "The fog computing paradigm: Scenarios and security issues," in *Proc. Federated Conf. Comput. Sci. Inf. Syst.*, Sep. 2014, pp. 1–8.
[78] M. Syamkumar, P. Barford, and R. Durairajan, "Deployment characteristics of 'the edge' in mobile edge computing," in *Proc. Workshop Mobile Edge Commun. (MECOMM)*, 2018, pp. 43–49.
[79] B. Bangerter, S. Talwar, R. Arefi, and K. Stewart, "Networks and devices for the 5G era," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 90–96, Feb. 2014.
[80] O. Abari, D. Bharadia, A. Duffield, and D. Katabi, "Enabling high-quality untethered virtual reality," in *Proc. 14th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2017, pp. 531–544.
[81] A. Jovicic, J. Li, and T. Richardson, "Visible light communication: Opportunities, challenges and the path to market," *IEEE Commun. Mag.*, vol. 51, no. 12, pp. 26–32, Dec. 2013.
[82] J. Gedeon, J. Krisztinkovics, C. Meurisch, M. Stein, L. Wang, and M. Mühlhäuser, "A multi-cloudlet infrastructure for future smart cities: An empirical study," in *Proc. 1st Int. Workshop Edge Syst., Anal. Netw. (EdgeSys)*, 2018, pp. 19–24.
[83] J. Gedeon, M. Stein, J. Krisztinkovics, P. Felka, K. Keller, C. Meurisch, L. Wang, and M. Mühlhäuser, "From cell towers to smart street lamps: Placing cloudlets on existing urban infrastructures," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 187–202.
[84] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Trans. Veh. Technol.*, vol. 65, no. 6, pp. 3860–3873, Jun. 2016.
[85] A. Sathiaseelan, A. Lertsinsrubtavee, A. Jagan, P. Baskaran, and J. Crowcroft, "Cloudrone: Micro clouds in the sky," in *Proc. 2nd Workshop Micro Aerial Vehicle Netw., Syst., Appl. Civilian Use (DroNet)*, 2016, pp. 41–44.
[86] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Trans. Veh. Technol.*, vol. 67, no. 3, pp. 2049–2063, Mar. 2018.
[87] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
[88] C. Pahl, S. Helmer, L. Miori, J. Sanin, and B. Lee, "A container-based edge cloud paas architecture based on raspberry Pi clusters," in *Proc. 4th IEEE Int. Conf. Future Internet Things Cloud Workshops (FiCloud)*, Aug. 2016, pp. 117–124.
[89] P. Bellavista and A. Zanni, "Feasibility of fog computing deployment based on docker containerization over RaspberryPi," in *Proc. 18th Int. Conf. Distrib. Comput. Netw. (ICDCN)*, 2017, p. 16.
[90] C. Meurisch, A. Seeliger, B. Schmidt, I. Schweizer, F. Kaup, and M. Mühlhäuser, "Upgrading wireless home routers for enabling large-scale deployment of cloudlets," in *Proc. 7th Int. Conf. Mobile Comput., Appl., Services (MobiCASE)*, 2015, pp. 12–29.
[91] V. F. S. Mota, D. F. Macedo, Y. Ghamri-Doudane, and J. M. S. Nogueira, "On the feasibility of WiFi offloading in urban areas: The Paris case study," in *Proc. IFIP Wireless Days (WD)*, Nov. 2013, pp. 1–6.
[92] E. C. Efstathiou, P. A. Frangoudis, and G. C. Polyzos, "Controlled Wi-Fi sharing in cities: A decentralized approach relying on indirect reciprocity," *IEEE Trans. Mobile Comput.*, vol. 9, no. 8, pp. 1147–1160, Aug. 2010.
[93] R. K. Balan, J. Flinn, M. Satyanarayanan, S. Sinnamohideen, and H.-I. Yang, "The case for cyber foraging," in *Proc. 10th ACM SIGOPS Eur. Workshop*, 2002, pp. 87–92.
[94] R. K. Balan and J. Flinn, "Cyber foraging: Fifteen years later," *IEEE Pervasive Comput.*, vol. 16, no. 3, pp. 24–30, Jul. 2017.
[95] K. Kumar, J. Liu, Y.-H. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Netw. Appl.*, vol. 18, no. 1, pp. 129–140, Feb. 2013.
[96] D. Kovachev, T. Yu, and R. Klamma, "Adaptive computation offloading from mobile devices into the cloud," in *Proc. 10th IEEE Int. Symp. Parallel Distrib. Process. Appl. (ISPA)*, Jul. 2012, pp. 784–791.
[97] Z. Li, C. Wang, and R. Xu, "Computation offloading to save energy on handheld devices: A partition scheme," in *Proc. Int. Conf. Compil., Archit. Synth. Embedded Syst. (CASES)*, 2001, pp. 238–246.
[98] M. Sharifi, S. Kafaie, and O. Kashefi, "A survey and taxonomy of cyber foraging of mobile devices," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 1232–1243, 4th Quart., 2012.
[99] H. Deng and H.-I. Hou, "Online scheduling for delayed mobile offloading," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 1867–1875.
[100] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, and R. Buyya, "mCloud: A context-aware offloading framework for heterogeneous mobile cloud," *IEEE Trans. Services Comput.*, vol. 10, no. 5, pp. 797–810, Sep./Oct. 2017.
[101] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Trans. Netw.*, vol. 24, no. 5, pp. 2795–2808, Oct. 2016.
[102] A. P. Miettinen and J. K. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Workshop Hot Topics Cloud Comput. (HotCloud)*, 2010, pp. 1–19.
[103] Q. Althebyan, Q. Yaseen, Y. Jararweh, and M. Al-Ayyoub, "Cloud support for large scale e-healthcare systems," *Ann. Télécommun.*, vol. 71, nos. 9–10, pp. 503–515, 2016.
[104] I. Azimi, A. Anzanpour, A. M. Rahmani, T. Pahikkala, M. Levorato, P. Liljeberg, and N. D. Dutt, "HiCH: Hierarchical fog-assisted computing architecture for healthcare IoT," *ACM Trans. Embedded Comput. Syst.*, vol. 16, no. 5, pp. 174:1–174:20, 2017.
[105] W. Zhang, J. Chen, Y. Zhang, and D. Raychaudhuri, "Towards efficient edge cloud augmentation for virtual reality MMOGs," in *Proc. 2nd ACM/IEEE Symp. Edge Comput. (SEC)*, Oct. 2017, pp. 8:1–8:14.
[106] T. Verbelen, P. Simoens, F. D. Turck, and B. Dhoedt, "AIOLOS: Middleware for improving mobile application performance through cyber foraging," *J. Syst. Softw.*, vol. 85, no. 11, pp. 2629–2639, 2012.
[107] M. S. Gordon, D. A. Jamshidi, S. A. Mahlke, Z. M. Mao, and X. Chen, "COMET: Code offload by migrating execution transparently," in *Proc. 13 USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2012, pp. 93–106.
[108] P. Liu, D. Willis, and S. Banerjee, "ParaDrop: Enabling lightweight multi-tenancy at the network's extreme edge," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 1–13.
[109] M. Golkarifard, J. Yang, Z. Huang, A. Movaghar, and P. Hui, "Dandelion: A unified code offloading system for wearable computing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 3, pp. 546–559, Mar. 2019.
[110] J. Gedeon, M. Wagner, J. Heuschkel, L. Wang, and M. Mühlhäuser, "A microservice store for efficient edge offloading," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Jul. 2019, pp. 1–6.
[111] C. Puliafito, E. Mingozzi, C. Vallati, F. Longo, and G. Merlino, "Virtualization and migration at the network edge: An overview," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, Jun. 2018, pp. 368–374.
[112] R. Morabito, V. Cozzolino, A. Y. Ding, N. Beijar, and J. Ott, "Consolidate IoT edge computing with lightweight virtualization," *IEEE Netw.*, vol. 32, no. 1, pp. 102–111, Jan./Feb. 2018.
[113] F. Ramalho and A. Neto, "Virtualization at the network edge: A performance comparison," in *Proc. 17th IEEE Int. Symp. World Wireless, Mobile Multimedia Netw. (WoWMoM)*, Jun. 2016, pp. 1–6.
[114] C. Pahl and B. Lee, "Containers and clusters for edge cloud architectures—A technology review," in *Proc. 3rd Int. Conf. Future Internet Things Cloud*, Aug. 2015, pp. 379–386.
[115] J. Thalheim, P. Bhatotia, P. Fonseca, and B. Kasikci, "CNTR: Lightweight OS containers," in *Proc. USENIX Annu. Tech. Conf. (ATC)*, 2018, pp. 199–212.
[116] A. Madhavapeddy, R. Mortier, C. Rotsos, D. Scott, B. Singh, T. Gazagnaire, S. Smith, S. Hand, and J. Crowcroft, "Unikernels: Library operating systems for the cloud," in *Proc. 18th Int. Conf. Archit. Support Program. Lang. Oper. Syst. (ASPLOS)*, 2013, pp. 461–472.
[117] A. Madhavapeddy and D. J. Scott, "Unikernels: Rise of the virtual library operating system," *Distrib. Comput.*, vol. 11, no. 11, pp. 30:30–30:44, 2014.
[118] V. Cozzolino, A. Y. Ding, and J. Ott, "FADES: Fine-grained edge offloading with unikernels," in *Proc. Workshop Hot Topics Container Netw. Netw. Syst. (HotConNet)*, 2017, pp. 36–41.

[119] S. Wu, C. Mei, H. Jin, and D. Wang, "Android unikernel: Gearing mobile code offloading towards edge computing," *Future Gener. Comput. Syst.*, vol. 86, pp. 694–703, Sep. 2018.

[120] S. Bhattacharjee, K. L. Calvert, and E. W. Zegura, "An architecture for active networking," in *Proc. 7th IFIP Int. Conf. High Perform. Netw. (HPN)*, 1997, pp. 265–279.

[121] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[122] J. Heuschkel, M. Stein, L. Wang, and M. Mühlhäuser, "Beyond the core: Enabling software-defined control at the network edge," in *Proc. Int. Conf. Netw. Syst. (NetSys)*, Mar. 2017, pp. 1–6.

[123] Y. Bi, G. Han, C. Lin, Q. Deng, L. Guo, and F. Li, "Mobility support for fog computing: An SDN approach," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 53–59, May 2018.

[124] J. Martins, M. Ahmed, C. Raiciu, V. A. Olteanu, M. Honda, R. Bifulco, and F. Huici, "ClickOS and the art of network function virtualization," in *Proc. 11th USENIX Symp. Netw. Syst. Design Implement. (NSDI)*, 2014, pp. 459–473.

[125] R. Mijumbi, J. Serrat, J. Gorricho, N. Bouten, F. De Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[126] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC)," *IEEE Netw.*, vol. 28, no. 6, pp. 18–26, Nov./Dec. 2014.

[127] S. Abdelwahab, B. Hamdaoui, M. Guizani, and T. Znati, "Network function virtualization in 5G," *IEEE Commun. Mag.*, vol. 54, no. 4, pp. 84–91, Apr. 2016.

[128] G. Cugola and A. Margara, "Processing flows of information: From data stream to complex event processing," *ACM Comput. Surv.*, vol. 44, no. 3, pp. 15-1–15-62, 2012.

[129] O. Saleh and K.-U. Sattler, "Distributed complex event processing in sensor networks," in *Proc. IEEE 14th Int. Conf. Mobile Data Manage.*, Jun. 2013, pp. 23–26.

[130] N. Govindarajan, Y. Simmhan, N. Jamadagni, and P. Misra, "Event processing across edge and the cloud for Internet of Things applications," in *Proc. 20th Int. Conf. Manage. Data (COMAD)*, 2014, pp. 101–104.

[131] E. I. Gaura, J. Brusey, M. Allen, R. Wilkins, D. Goldsmith, and R. Rednic, "Edge mining the Internet of Things," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3816–3825, Oct. 2013.

[132] G. Ananthanarayanan, P. Bahl, P. Bodík, K. Chintalapudi, M. Philipose, L. Ravindranath, and S. Sinha, "Real-time video analytics: The killer app for edge computing," *Computer*, vol. 50, no. 10, pp. 58–67, 2017.

[133] P. Liu, B. Qi, and S. Banerjee, "EdgeEye: An edge service framework for real-time intelligent video analytics," in *Proc. 1st Int. Workshop Edge Syst., Anal. Netw. (EdgeSys)*, 2018, pp. 1–6.

[134] N. Powers, A. Alling, K. Osolinsky, T. Soyata, M. Zhu, H. Wang, H. Ba, W. B. Heinzelman, J. Shi, and M. Kwon, "The cloudlet accelerator: Bringing mobile-cloud face recognition into real-time," in *Proc. IEEE Globecom Workshops*, Dec. 2015, pp. 1–7.

[135] M. Nardelli, V. Cardellini, V. Grassi, and F. L. Presti, "Efficient operator placement for distributed data stream processing applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 8, pp. 1753–1767, Aug. 2019.

[136] V. Cardellini, V. Grassi, F. L. Presti, and M. Nardelli, "On QoS-aware scheduling of data stream applications over fog computing infrastructures," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jul. 2015, pp. 271–276.

[137] D. Li, T. Salonidis, N. V. Desai, and M. C. Chuah, "DeepCham: Collaborative edge-mediated adaptive deep learning for mobile object recognition," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2016, pp. 64–76.

[138] P. Hu, H. Ning, T. Qiu, Y. Zhang, and X. Lou, "Fog computing based face identification and resolution scheme in Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 13, no. 4, pp. 1910–1920, Aug. 2017.

[139] E. Li, Z. Zhou, and X. Chen, "Edge intelligence: On-demand deep learning model co-inference with device-edge synergy," in *Proc. Workshop Mobile Edge Commun. (MECOMM)*, 2018, pp. 31–36.

[140] W. Cai, M. Chen, and V. C. M. Leung, "Toward gaming as a service," *IEEE Internet Comput.*, vol. 18, no. 3, pp. 12–18, May/Jun. 2014.

[141] L. Pantel and L. C. Wolf, "On the impact of delay on real-time multiplayer games," in *Proc. 12th Int. Workshop Netw. Oper. Syst. Support Digit. Audio Video (NOSSDAV)*, 2002, pp. 23–29.

[142] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "The brewing storm in cloud gaming: A measurement study on cloud to end-user latency," in *Proc. 11th Annu. Workshop Netw. Syst. Support Games (NetGames)*, 2012, pp. 2:1–2:6.

[143] J. N. Plumb and R. Stutsman, "Exploiting Google's edge network for massively multiplayer online games," in *Proc. 2nd IEEE Int. Conf. Fog Edge Comput. (ICFEC)*, May 2018, pp. 1–8.

[144] F. Messaoudi, A. Ksentini, and P. Bertin, "Toward a mobile gaming based-computation offloading," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–7.

[145] T. Kämäräinen, M. Siekkinen, Y. Xiao, and A. Ylä-Jääski, "Towards pervasive and mobile gaming with distributed cloud infrastructure," in *Proc. 13th Annu. Workshop Netw. Syst. Support Games*, 2014, pp. 1–6.

[146] W. Cai, F. Chi, X. Wang, and V. C. M. Leung, "Toward multiplayer cooperative cloud gaming," *IEEE Cloud Comput.*, vol. 5, no. 5, pp. 70–80, Aug. 2018.

[147] P. J. Braun, S. Pandi, R.-S. Schmoll, and F. H. P. Fitzek, "On the study and deployment of mobile edge cloud for tactile Internet using a 5G gaming application," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2017, pp. 154–159.

[148] R. T. Azuma, "A survey of augmented reality," *Presence*, vol. 6, no. 4, pp. 355–385, 1997.

[149] J. Steuer, "Defining virtual reality: Dimensions determining telepresence," *J. Commun.*, vol. 42, no. 4, pp. 73–93, Dec. 2010.

[150] J. Cho, K. Sundaresan, R. Mahindra, J. Van der Merwe, and S. Rangarajan, "ACACIA: Context-aware edge computing for continuous interactive applications over mobile networks," in *Proc. 12th Int. Conf. Emerg. Netw. Exp. Technol. (CoNEXT)*, 2016, pp. 375–389.

[151] H.-K. Wu, S. W.-Y. Lee, H.-Y. Chang, and J.-C. Liang, "Current status, opportunities and challenges of augmented reality in education," *Comput. Educ.*, vol. 62, pp. 41–49, Mar. 2013.

[152] B. H. Thomas, "A survey of visual, mixed, and augmented reality gaming," *Comput. Entertainment*, vol. 10, no. 1, pp. 3:1–3:33, 2012.

[153] K. Mania, B. D. Adelstein, S. R. Ellis, and M. I. Hill, "Perceptual sensitivity to head tracking latency in virtual environments with varying degrees of scene complexity," in *Proc. 1st Symp. Appl. Perception Graph. Vis. (APGV)*, 2004, pp. 39–47.

[154] E. Cuervo, K. Chintalapudi, and M. Kotaru, "Creating the perfect illusion: What will it take to create life-like virtual reality headsets?" in *Proc. 19th Int. Workshop Mobile Comput. Syst. (HotMobile)*, 2018, pp. 7–12.

[155] T. Braud, F. H. Bijarbooneh, D. Chatzopoulos, and P. Hui, "Future networking challenges: The case of mobile augmented reality," in *Proc. 37th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 1796–1807.

[156] M. Trinelli, M. Gallo, M. Rifai, and F. Pianese, "Transparent AR processing acceleration at the edge," in *Proc. 2nd Int. Workshop Edge Syst., Anal. Netw. (EdgeSys)*, 2019, pp. 30–35.

[157] M. S. Elbamby, C. Perfecto, M. Bennis, and K. Doppler, "Toward low-latency and ultra-reliable virtual reality," *IEEE Netw.*, vol. 32, no. 2, pp. 78–84, Mar./Apr. 2018.

[158] Z. Lai, Y. C. Hu, Y. Cui, L. Sun, and N. Dai, "Furion: Engineering high-quality immersive virtual reality on today's mobile devices," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw. (MobiCom)*, 2017, pp. 409–421.

[159] S. Shi, V. Gupta, M. Hwang, and R. Jana, "Mobile VR on edge cloud: A latency-driven design," in *Proc. 10th ACM Multimedia Syst. Conf. (MMSys)*, 2019, pp. 222–231.

[160] Q. Liu and T. Han, "DARE: Dynamic adaptive mobile augmented reality with edge computing," in *Proc. IEEE 26th Int. Conf. Netw. Protocols (ICNP)*, Sep. 2018, pp. 1–11.

[161] S. Bohez, J. Couvreur, B. Dhoedt, and P. Simoens, "Cloudlet-based large-scale 3D reconstruction using real-time data from mobile depth cameras," in *Proc. 6th Int. Workshop Mobile Cloud Comput. (MCS)*, 2015, pp. 8–14.

[162] M. Viitanen, J. Vanne, T. D. Hämäläinen, and A. Kulmala, "Low latency edge rendering scheme for interactive 360 degree virtual reality gaming," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2018, pp. 1557–1560.

[163] J. K. Zao, T. T. Gan, C. K. You, S. J. R. Mendez, C. E. Chung, Y. Wang, T. R. Mullen, and T. Jung, "Augmented brain computer interaction based on fog computing and linked data," in *Proc. Int. Conf. Intell. Environ.*, Jun./Jul. 2014, pp. 374–377.

[164] Z. Huang, P. Hui, C. Peylo, and D. Chatzopoulos, "Mobile augmented reality survey: A bottom-up approach," 2013, *arXiv:1309.4413*. [Online]. Available: https://arxiv.org/abs/1309.4413

[165] J. Dilley, B. Maggs, J. Parikh, H. Prokop, R. Sitaraman, and B. Weihl, "Globally distributed content delivery," *IEEE Internet Comput.*, vol. 6, no. 5, pp. 50–58, Sep. 2002.

[166] (2013). *Ericsson Mobility Report*. [Online]. Available: https://metis2020.com/res/docs/2013/ericsson-mobility-report-june-2013.pdf

[167] K. Mokhtarian and H.-A. Jacobsen, "Coordinated caching in planet-scale CDNs: Analysis of feasibility and benefits," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.

[168] H. Ahlehagh and S. Dey, "Video-aware scheduling and caching in the radio access network," *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1444–1462, Oct. 2014.

[169] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[170] J. Zhu, D. S. Chan, M. S. Prabhu, P. Natarajan, H. Hu, and F. Bonomi, "Improving Web sites performance using edge servers in fog computing architecture," in *Proc. IEEE 7th Int. Symp. Service-Oriented Syst. Eng.*, Mar. 2013, pp. 320–323.

[171] E. Zeydan, E. Bastug, M. Bennis, M. A. Kader, I. A. Karatepe, A. S. Er, and M. Debbah, "Big data caching for networking: Moving from cloud to edge," *IEEE Commun. Mag.*, vol. 54, no. 9, pp. 36–42, Sep. 2016.

[172] D. Applegate, A. Archer, V. Gopalakrishnan, S. Lee, and K. K. Ramakrishnan, "Optimal content placement for a large-scale VoD system," in *Proc. ACM Conf. Emerg. Netw. Exp. (Co-NEXT)*, 2010, pp. 4:1–4:12.

[173] J. Liu, Q. Yang, and G. Simon, "Optimal and practical algorithms for implementing wireless CDN based on base stations," in *Proc. IEEE 83rd Vehicular Technol. Conf. (VTC)*, May 2016, pp. 1–5.

[174] T. X. Tran, P. Pandey, A. Hajisami, and D. Pompili, "Collaborative multi-bitrate video caching and processing in mobile-edge computing networks," in *Proc. 13th Annu. Conf. Wireless On-Demand Netw. Syst. Services (WONS)*, Feb. 2017, pp. 165–172.

[175] U. Drolia, K. Guo, J. Tan, R. Gandhi, and P. Narasimhan, "Cachier: Edge-caching for recognition applications," in *Proc. 37th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 276–286.

[176] I. Lujic, V. De Maio, and I. Brandic, "Efficient edge storage management based on near real-time forecasts," in *Proc. 1st IEEE Int. Conf. Fog Edge Comput. (ICFEC)*, May 2017, pp. 21–30.

[177] K. Bhardwaj, P. Agrawal, A. Gavrilovska, and K. Schwan, "AppSachet: Distributed app delivery from the edge cloud," in *Proc. 7th Int. Conf. Mobile Comput., Appl. Services (MobiCASE)*, 2015, pp. 89–106.

[178] K. Bhardwaj, P. Agarwal, A. Gavrilovska, K. Schwan, and A. Allred, "AppFlux: Taming app delivery via streaming," in *Proc. USENIX Conf. Timely Results Oper. Syst. (TRIOS)*, 2015, pp. 1–14.

[179] E. M. Schooler, D. Zage, J. Sedayao, H. Moustafa, A. Brown, and M. Ambrosin, "An architectural vision for a data-centric IoT: Rethinking things, trust and clouds," in *Proc. 37th IEEE Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 1717–1728.

[180] H. Moustafa, E. M. Schooler, and J. McCarthy, "Reverse CDN in fog computing: The lifecycle of video data in connected and autonomous vehicles," in *Proc. IEEE Fog World Congr. (FWC)*, Oct./Nov. 2017, pp. 1–5.

[181] P. Simoens, Y. Xiao, P. Pillai, Z. Chen, K. Ha, and M. Satyanarayanan, "Scalable crowd-sourcing of video from mobile devices," in *Proc. 11th Int. Conf. Mobile Syst., Appl. (MobiSys)*, 2013, pp. 139–152.

[182] A. Chaudhry, J. Crowcroft, H. Howard, A. Madhavapeddy, R. Mortier, H. Haddadi, and D. McAuley, "Personal data: Thinking inside the box," in *Proc. 5th Decennial Aarhus Conf. Crit. Alternatives (CA)*, 2015, pp. 29–32.

[183] R. Mortier, J. Zhao, J. Crowcroft, L. Wang, Q. Li, H. Haddadi, Y. Amar, A. Crabtree, J. Colley, T. Lodge, T. Brown, D. McAuley, and C. Greenhalgh, "Personal data management with the databox: What's inside the box?" in *Proc. ACM Workshop Cloud-Assist. Netw.*, 2016, pp. 49–54.

[184] C. Perera, S. Y. Wakenshaw, T. Baarslag, H. Haddadi, A. K. Bandara, R. Mortier, A. Crabtree, I. C. Ng, D. McAuley, and J. Crowcroft, "Valorising the IoT databox: Creating value for everyone," *Trans. Emerg. Telecommun. Technol.*, vol. 28, no. 1, pp. 1–17, 2017.

[185] M. Lacinák and J. Ristvej, "Smart city, safety and security," *Procedia Eng.*, vol. 192, pp. 522–527, Jan. 2017.

[186] H. Schaffers, N. Komninos, M. Pallot, B. Trousse, M. Nilsson, and A. Oliveira, "Smart cities and the future Internet: Towards cooperation frameworks for open innovation," in *Future Internet Future Internet Assembly 2011: Achievements and Technological Promises*. Berlin, Germany: Springer, 2011, pp. 431–446.

[187] J. M. Schleicher, M. Vögler, S. Dustdar, and C. Inzinger, "Enabling a smart city application ecosystem: Requirements and architectural aspects," *IEEE Internet Comput.*, vol. 20, no. 2, pp. 58–65, Feb. 2016.

[188] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, "Urban computing: Concepts, methodologies, and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, p. 38, 2014.

[189] I. Schweizer, C. Meurisch, J. Gedeon, R. Bärtl, and M. Mühlhäuser, "Noisemap: Multi-tier incentive mechanisms for participative urban sensing," in *Proc. 3rd Int. Workshop Sens. Appl. Mobile Phones (PhoneSense)*, 2012, pp. 9:1–9:5.

[190] F. Calabrese, L. Ferrari, and V. De Blondel, "Urban sensing using mobile phone network data: A survey of research," *ACM Comput. Surv.*, vol. 47, no. 2, p. 25, Jan. 2015.

[191] T. Heo, K. Kim, H. Kim, C. Lee, J. H. Ryu, Y. T. Leem, J. A. Jun, C. Pyo, S.-M. Yoo, and J. Ko, "Escaping from ancient Rome! Applications and challenges for designing smart cities," *Trans. Emerg. Telecommun. Technol.*, vol. 25, no. 1, pp. 109–119, 2014.

[192] R. Petrolo, V. Loscrì, and N. Mitton, "Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms," *Trans. Emerg. Telecommun. Technol.*, vol. 28, no. 1, p. e2931, 2017.

[193] A. Zanella, N. Bui, A. Castellani, L. Vangelista, and M. Zorzi, "Internet of Things for smart cities," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[194] S. Howell, Y. Rezgui, J.-L. Hippolyte, B. Jayan, and H. Li, "Towards the next generation of smart grids: Semantic and holonic multi-agent management of distributed energy resources," *Renew. Sustain. Energy Rev.*, vol. 77, pp. 193–214, Sep. 2017.

[195] E. Negeri and N. Baken, "Architecting the smart grid as a holarchy," in *Proc. 1st Int. Conf. Smart Grids Green IT Syst.*, 2012, pp. 1–6.

[196] S. Frey, A. Diaconescu, D. Menga, and I. Demeure, "A holonic control architecture for a heterogeneous multi-objective smart micro-grid," in *Proc. IEEE 7th Int. Conf. Self-Adapt. Self-Organizing Syst.*, Sep. 2013, pp. 21–30.

[197] S. Althaher, P. Mancarella, and J. Mutale, "Automated demand response from home energy management system under dynamic pricing and power and comfort constraints," *IEEE Trans. Smart Grid*, vol. 6, no. 4, pp. 1874–1883, Jul. 2015.

[198] M. A. A. Pedrasa, T. D. Spooner, and I. F. MacGill, "Coordinated scheduling of residential distributed energy resources to optimize smart home energy services," *IEEE Trans. Smart Grid*, vol. 1, no. 2, pp. 134–143, Sep. 2010.

[199] A. Reinhardt, P. Baumann, D. Burgstahler, M. Hollick, H. Chonov, M. Werner, and R. Steinmetz, "On the accuracy of appliance identification based on distributed load metering data," in *Proc. 2nd IFIP Conf. Sustain. Internet ICT Sustainab. (SustainIT)*, Oct. 2012, pp. 1–9.

[200] B. Ghazal, K. ElKhatib, K. Chahine, and M. Kherfan, "Smart traffic light control system," in *Proc. 3rd Int. Conf. Elect., Electron., Comput. Eng. Appl. (EECEA)*, Apr. 2016, pp. 140–145.

[201] V. Cozzolino, A. Y. Ding, and J. Ott, "Edge chaining framework for black ice road fingerprinting," in *Proc. 2nd Int. Workshop Edge Syst., Anal. Netw. (EdgeSys)*, 2019, pp. 42–47.

[202] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban computing with taxicabs," in *Proc. 13th Int. Conf. Ubiquitous Comput. (UbiComp)*, 2011, pp. 89–98.

[203] M. Satyanarayanan, P. B. Gibbons, L. B. Mummert, P. Pillai, P. Simoens, and R. Sukthankar, "Cloudlet-based just-in-time indexing of IoT video," in *Proc. Global Internet Things Summit (GIoTS)*, Jun. 2017, pp. 1–8.

[204] B. Qi, L. Kang, and S. Banerjee, "A vehicle-based edge computing platform for transit and human mobility analytics," in *Proc. 2nd ACM/IEEE Symp. Edge Comput. (SEC)*, 2017, pp. 1:1–1:14.

[205] O. Abari, D. Vasisht, D. Katabi, and A. Chandrakasan, "Caraoke: An e-toll transponder network for smart cities," in *Proc. ACM Conf. Special Interest Group Data Commun. (SIGCOMM)*, 2015, pp. 297–310.

[206] F. J. Villanueva, D. Villa, M. J. Santofimia, J. Barba, and J. C. López, "Crowdsensing smart city parking monitoring," in *Proc. 2nd IEEE World Forum Internet Things (WF-IoT)*, Dec. 2015, pp. 751–756.

[207] H. Hartenstein and L. P. Laberteaux, "A tutorial survey on vehicular ad hoc networks," *IEEE Commun. Mag.*, vol. 46, no. 6, pp. 164–171, Jun. 2008.

[208] H. Menouar, I. Güvenç, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer, "UAV-enabled intelligent transportation systems for the smart city: Applications and challenges," *IEEE Commun. Mag.*, vol. 55, no. 3, pp. 22–28, 2017.

[209] S. K. Datta, C. Bonnet, and J. Haerri, "Fog computing architecture to enable consumer centric Internet of Things services," in *Proc. Int. Symp. Consum. Electron. (ISCE)*, Jun. 2015, pp. 1–2.
[210] E. Schoch, F. Kargl, M. Weber, and T. Leinmuller, "Communication patterns in VANETs," *IEEE Commun. Mag.*, vol. 46, no. 11, pp. 119–125, Nov. 2008.
[211] L. Liu, X. Zhang, M. Qiao, and W. Shi, "SafeShareRide: Edge-based attack detection in ridesharing services," in *Proc. IEEE/ACM Symp. Edge Comput. (SEC)*, Oct. 2018, pp. 17–29.
[212] Q. Yuan, H. Zhou, J. Li, Z. Liu, F. Yang, and X. Shen, "Toward efficient content delivery for automated driving services: An edge computing solution," *IEEE Netw.*, vol. 32, no. 1, pp. 80–86, Jan./Feb. 2018.
[213] H. Qiu, F. Ahmad, F. Bai, M. Gruteser, and R. Govindan, "AVR: Augmented vehicular reality," in *Proc. 16th Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2018, pp. 81–95.
[214] C. Lochert, B. Scheuermann, C. Wewetzer, A. Lüebke, and M. Mauve, "Data aggregation and roadside unit placement for a VANET traffic information system," in *Proc. 5th Int. Workshop Veh. Ad Hoc Netw. (VANET)*, 2008, pp. 58–65.
[215] M. Raya and J.-P. Hubaux, "The security of vehicular ad hoc networks," in *Proc. 3rd ACM Workshop Secur. Ad Hoc Sensor Netw. (SASN)*, 2005, pp. 11–21.
[216] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-Air: When urban air quality inference meets big data," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 2013, pp. 1436–1444.
[217] Y. Zheng, T. Liu, Y. Wang, Y. Zhu, Y. Liu, and E. Chang, "Diagnosing New York City's noises with ubiquitous data," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. (UbiComp)*, 2014, pp. 715–725.
[218] J. Dutta, C. Chowdhury, S. Roy, A. I. Middya, and F. Gazi, "Towards smart city: Sensing air quality in city based on opportunistic crowdsensing," in *Proc. 18th Int. Conf. Distrib. Comput. Netw. (ICDCN)*, 2017, pp. 42:1–42:6.
[219] F. Samie, L. Bauer, and J. Henkel, "IoT technologies for embedded computing: A survey," in *Proc. 11th IEEE/ACM/IFIP Int. Conf. Hardw./Softw. Codesign Syst. Synth. (CODES)*, 2016, pp. 8:1–8:10.
[220] M. Marjanović, A. Antonić, and I. P. Žarko, "Edge computing architecture for mobile crowdsensing," *IEEE Access*, vol. 6, pp. 10662–10674, 2018.
[221] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in *Proc. IEEE Int. Conf. Future Internet Things Cloud (FiCloud)*, Aug. 2014, pp. 464–470.
[222] P.-H. Tsai, H.-J. Hong, A.-C. Cheng, and C.-H. Hsu, "Distributed analytics in fog computing platforms using tensorflow and kubernetes," in *Proc. 19th Asia–Pacific Netw. Oper. Manage. Symp. (APNOMS)*, Sep. 2017, pp. 145–150.
[223] D. Zhang, Y. Ma, Y. Zhang, S. Lin, X. S. Hu, and D. Wang, "A real-time and non-cooperative task allocation framework for social sensing applications in edge computing systems," in *Proc. IEEE Real-Time Embedded Technol. Appl. Symp. (RTAS)*, Apr. 2018, pp. 316–326.
[224] N. Maisonneuve, M. Stevens, M. E. Niessen, and L. Steels, "NoiseTube: Measuring and mapping noise pollution with mobile phones," in *Proc. 4th Int. ICSC Symp.*, 2009, pp. 215–228.
[225] T. Nuortio, J. Kytöjoki, H. Niska, and O. Bräysy, "Improved route planning and scheduling of waste collection and transport," *Expert Syst. Appl.*, vol. 30, no. 2, pp. 223–232, 2006.
[226] T. Anagnostopoulos, A. Zaslavsky, and A. Medvedev, "Robust waste collection exploiting cost efficiency of IoT potentiality in smart cities," in *Proc. Int. Conf. Recent Adv. Internet Things (RIoT)*, Apr. 2015, pp. 1–6.
[227] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos, "Sensing as a service model for smart cities supported by Internet of Things," *Trans. Emerg. Telecommun. Technol.*, vol. 25, no. 1, pp. 81–93, 2014.
[228] M. Aazam, M. St-Hilaire, C.-H. Lung, and I. Lambadaris, "Cloud-based smart waste management for smart cities," in *Proc. IEEE 21st Int. Workshop Comput. Aided Modeling Design Commun. Links Netw. (CAMAD)*, Oct. 2016, pp. 188–193.
[229] A. Medvedev, P. Fedchenkov, A. Zaslavsky, T. Anagnostopoulos, and S. Khoruzhnikov, "Waste management as an IoT-enabled service in smart cities," in *Proc. 15th Int. Conf. Internet Things, Smart Spaces, Next Gener. Netw. Syst. (NEW2AN)*, S. Balandin, S. Andreev, and Y. Koucheryavy, Eds. Springer, 2015, pp. 104–115.
[230] C. Chung, D. Egan, A. Jain, N. Caruso, C. Misner, and R. Wallace, "A cloud-based mobile computing applications platform for first responders," in *Proc. 7th IEEE Int. Symp. Service-Oriented Syst. (SOSE)*, Mar. 2013, pp. 503–508.
[231] M. Aazam and E.-N. Huh, "E-HAMC: Leveraging fog computing for emergency alert service," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, Mar. 2015, pp. 518–523.
[232] M. Sapienza, E. Guardo, M. Cavallo, G. L. Torre, G. Leombruno, and O. Tomarchio, "Solving critical events through mobile edge computing: An approach for smart cities," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, May 2016, pp. 1–5.
[233] C. Meurisch, T. A. B. Nguyen, J. Gedeon, F. Konhauser, M. Schmittner, S. Niemczyk, S. Wullkotte, and M. Mühlhäuser, "Upgrading wireless home routers as emergency cloudlet and secure DTN communication bridge," in *Proc. 26th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2017, pp. 1–2.
[234] M. Satyanarayanan, G. Lewis, E. Morris, S. Simanta, J. Boleng, and K. Ha, "The role of cloudlets in hostile environments," *IEEE Pervasive Comput.*, vol. 12, no. 4, pp. 40–49, Oct. 2013.
[235] L. Canzian and M. van der Schaar, "Real-time stream mining: Online knowledge extraction using classifier networks," *IEEE Netw.*, vol. 29, no. 5, pp. 10–16, Sep. 2015.
[236] N. Chen, Y. Chen, X. Ye, H. Ling, S. Song, and C.-T. Huang, "Smart city surveillance in fog computing," in *Advances in Mobile Cloud Computing and Big Data in the 5G Era*, C. X. Mavromoustakis, G. Mastorakis, and C. Dobre, Eds. Springer, 2017, pp. 203–226.
[237] Z. Xu, H. Gupta, and U. Ramachandran, "STTR: A system for tracking all vehicles all the time at the edge of the network," in *Proc. 12th ACM Int. Conf. Distrib. Event-Based Syst. (DEBS)*, Jun. 2018, pp. 124–135.
[238] C. Mihale-Wilson, P. Felka, and O. Hinz, "The bright and the dark side of smart lights? The protective effect of smart city infrastructures," in *Proc. Hawaii Int. Conf. Syst. Sci. (HICSS)*, 2019, pp. 3345–3354.
[239] M. Stojmenovic, "Mobile cloud computing for biometric applications," in *Proc. 15th Int. Conf. Netw.-Based Inf. Syst. (NBiS)*, 2012, pp. 654–659.
[240] R. Want, "An introduction to RFID technology," *IEEE Pervasive Comput.*, vol. 5, no. 1, pp. 25–33, Jan./Mar. 2006.
[241] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, 4th Quart., 2015.
[242] J. Pan and J. McElhannon, "Future edge cloud and edge computing for Internet of Things applications," *IEEE Internet Things J.*, vol. 5, no. 1, pp. 439–449, Feb. 2018.
[243] N. M. Sandar, S. Chaisiri, S. Yongchareon, and V. Liesaputra, "Cloud-based video monitoring framework: An approach based on software-defined networking for addressing scalability problems," in *Proc. Web Inf. Syst. Eng. (WISE) Workshops*, 2014, pp. 176–189.
[244] M. Casini, "Internet of Things for energy efficiency of buildings," *Int. Sci. J. Archit. Eng.*, vol. 2, no. 1, pp. 24–28, 2014.
[245] R. Abdallah, L. Xu, and W. Shi, "Lessons and experiences of a DIY smart home," in *Proc. Workshop Smart Internet Things (SmartIoT)*, 2017, pp. 4:1–4:6.
[246] J. Pan, L. Ma, R. Ravindran, and P. TalebiFard, "HomeCloud: An edge cloud framework and testbed for new application delivery," in *Proc. 23rd Int. Conf. Telecommun. (ICT)*, May 2016, pp. 1–6.
[247] C. Vallati, A. Virdis, E. Mingozzi, and G. Stea, "Mobile-edge computing come home connecting things in future smart homes using LTE device-to-device communications," *IEEE Consum. Electron. Mag.*, vol. 5, no. 4, pp. 77–83, Oct. 2016.
[248] F.-J. Ferrández-Pastor, H. Mora, A. Jimeno-Morenilla, and B. Volckaert, "Deployment of IoT edge and fog computing technologies to develop smart building services," *Sustainability*, vol. 10, no. 11, p. 3832, 2018.
[249] J. Wan, B. Chen, S. Wang, M. Xia, D. Li, and C. Liu, "Fog computing for energy-aware load balancing and scheduling in smart factory," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4548–4556, Oct. 2018.
[250] P. Lade, R. Ghosh, and S. Srinivasan, "Manufacturing analytics and industrial Internet of Things," *IEEE Intell. Syst.*, vol. 32, no. 3, pp. 74–79, May/Jun. 2017.
[251] M. Aazam, S. Zeadally, and K. A. Harras, "Deploying fog computing in industrial Internet of Things and industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4674–4682, Oct. 2018.
[252] J.-S. Fu, Y. Liu, H.-C. Chao, B. K. Bhargava, and Z.-J. Zhang, "Secure data storage and searching for industrial IoT by integrating fog computing and cloud computing," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4519–4528, Oct. 2018.

[253] E. Oyekanlu, "Predictive edge computing for time series of industrial IoT and large scale critical infrastructure based on open-source software analytic of big data," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 1663–1669.
[254] D. Wu, S. Liu, L. Zhang, J. Terpenny, R. X. Gao, T. Kurfess, and J. A. Guzzo, "A fog computing-based framework for process monitoring and prognosis in cyber-manufacturing," *J. Manuf. Syst.*, vol. 43, pp. 25–34, Apr. 2017.
[255] C.-C. Lin and J.-W. Yang, "Cost-efficient deployment of fog computing systems at logistics centers in industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 14, no. 10, pp. 4603–4611, Apr. 2018.
[256] W. Steiner and S. Poledna, "Fog computing as enabler for the industrial Internet of Things," *Elektrotechn. Informationstechn.*, vol. 133, no. 7, pp. 310–314, 2016.
[257] N. Ahmed, D. De, and I. Hussain, "Internet of Things (IoT) for smart precision agriculture and farming in rural areas," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4890–4899, Dec. 2018.
[258] K. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine learning in agriculture: A review," *Sensors*, vol. 18, no. 8, p. 2674, 2018.
[259] F. J. F. Pastor, J. M. G. Chamizo, M. Nieto-Hidalgo, and J. Mora-Martínez, "Precision agriculture design method using a distributed computing architecture on Internet of Things context," *Sensors*, vol. 18, no. 6, p. 1731, 2018.
[260] M. Caria, J. Schudrowitz, A. Jukan, and N. Kemper, "Smart farm computing systems for animal welfare monitoring," in *Proc. 40th Int. Conv. Inf. Commun. Technol., Electron. Microelectron. (MIPRO)*, May 2017, pp. 152–157.
[261] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big data in smart farming—A review," *Agricult. Syst.*, vol. 153, pp. 69–80, May 2017.
[262] S. Ivanov, K. Bhargava, and W. Donnelly, "Precision farming: Sensor analytics," *IEEE Intell. Syst.*, vol. 30, no. 4, pp. 76–80, Jul. 2015.
[263] H. Fereidooni, T. Frassetto, M. Miettinen, A. Sadeghi, and M. Conti, "Fitness trackers: Fit for health but unfit for security and privacy," in *Proc. IEEE/ACM Int. Conf. Connected Health, Appl., Syst. Eng. Technol. (CHASE)*, Jul. 2017, pp. 19–24.
[264] D. Nafus and J. Sherman, "Big data, big questions| this one does not go up to 11: The quantified self movement as an alternative big data practice," *Int. J. Commun.*, vol. 8, no. 11, pp. 1784–1794, 2014.
[265] M. Swan, "Health 2050: The realization of personalized medicine through crowdsourcing, the quantified self, and the participatory biocitizen," *J. Pers. Med.*, vol. 2, no. 3, pp. 93–118, 2012.
[266] J. N. Gilmore, "EveryWear: The quantified self and wearable fitness technologies," *New Media Soc.*, vol. 18, no. 11, pp. 2524–2539, 2016.
[267] B. Schmidt, S. Benchea, R. Eichin, and C. Meurisch, "Fitness tracker or digital personal coach: How to personalize training," in *Proc. ACM Int. Joint Conf. Pervasive Ubiquitous Comput. ACM Int. Symp. Wearable Comput. (UbiComp/ISWC)*, 2015, pp. 1063–1067.
[268] A. Bajpai, V. Jilla, V. N. Tiwari, S. M. Venkatesan, and R. Narayanan, "Quantifiable fitness tracking using wearable devices," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2015, pp. 1633–1637.
[269] N. Bui and M. Zorzi, "Health care applications: A solution based on the Internet of Things," in *Proc. 4th Int. Symp. Appl. Sci. Biomed. Commun. Technol. (ISABEL)*, 2011, pp. 131:1–131:5.
[270] F. Alagöz, A. C. Valdez, W. Wilkowska, M. Ziefle, S. Dorner, and A. Holzinger, "From cloud computing to mobile Internet, from user focus to culture and hedonism: The crucible of mobile health care and wellness applications," in *Proc. 5th Int. Conf. Pervasive Comput. Appl. (ICPCA)*, 2010, pp. 38–45.
[271] F. A. Kraemer, A. E. Braten, N. Tamkittikhun, and D. Palma, "Fog computing in healthcare—A review and discussion," *IEEE Access*, vol. 5, pp. 9206–9222, 2017.
[272] Y. Cao, S. Chen, P. Hou, and D. Brown, "FAST: A fog computing assisted distributed analytics system to monitor fall for stroke mitigation," in *Proc. 10th IEEE Int. Conf. Netw., Archit. Storage (NAS)*, Aug. 2015, pp. 2–11.
[273] T. N. Gia, M. Jiang, A. Rahmani, T. Westerlund, P. Liljeberg, and H. Tenhunen, "Fog computing in healthcare Internet of Things: A case study on ECG feature extraction," in *Proc. IEEE Int. Conf. Comput. Inf. Technol., Ubiquitous Comput. Commun., Dependable, Auton. Secure Comput., Pervasive Intell. Comput.*, 2015, pp. 356–363.
[274] S. Xu, M. Pérez, K. Yang, C. Perrenot, J. Felblinger, and J. Hubert, "Determination of the latency effects on surgical performance and the acceptable latency levels in telesurgery using the dV-trainer simulator," *Surg. Endoscopy*, vol. 28, no. 9, pp. 2569–2576, Sep. 2014.
[275] A. Alesanco and J. García, "Clinical assessment of wireless ECG transmission in real-time cardiac telemonitoring," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 5, pp. 1144–1152, Apr. 2010.
[276] A. E. Amraoui and K. Sethom, "Cloudlet softwarization for pervasive healthcare," in *Proc. 30th Int. Conf. Adv. Inf. Netw. Appl. (AINA) Workshops*, Mar. 2016, pp. 628–632.
[277] M. Satyanarayanan, "Augmenting cognition," *IEEE Pervasive Comput.*, vol. 3, no. 2, pp. 4–5, Apr. 2004.
[278] M. Ramon, S. Caharel, and B. Rossion, "The speed of recognition of personally familiar faces," *Perception*, vol. 40, pp. 437–449, Apr. 2011.
[279] T. R. Agus, C. Suied, S. J. Thorpe, and D. Pressnitzer, "Characteristics of human voice processing," in *Proc. Int. Symp. Circuits Syst. (ISCAS)*, May 2010, pp. 509–512.
[280] K. Ha, Z. Chen, W. Hu, W. Richter, P. Pillai, and M. Satyanarayanan, "Towards wearable cognitive assistance," in *Proc. 12th Annu. Int. Conf. Mobile Syst., Appl., Services (MobiSys)*, 2014, pp. 68–81.
[281] B. Varghese and R. Buyya, "Next generation cloud computing: New trends and research directions," *Future Gener. Comput. Syst.*, vol. 79, pp. 849–861, Feb. 2018.
[282] F. Manco, C. Lupu, F. Schmidt, J. Mendes, S. Kuenzer, S. Sati, K. Yasukata, C. Raiciu, and F. Huici, "My VM is lighter (and safer) than your container," in *Proc. 26th Symp. Oper. Syst. Princ. (SOSP)*, 2017, pp. 218–233.
[283] T. Combe, A. Martin, and R. Di Pietro, "To docker or not to docker: A security perspective," *IEEE Cloud Comput.*, vol. 3, no. 5, pp. 54–62, Sep. 2016.
[284] Y. Wang, T. Uehara, and R. Sasaki, "Fog computing: Issues and challenges in security and forensics," in *Proc. IEEE 39th Annu. Comput. Softw. Appl. Conf. (COMPSAC)*, vol. 3, Jul. 2015, pp. 53–59.
[285] E. Vasilomanolakis, J. Daubert, M. Luthra, V. Gazis, A. Wiesmaier, and P. Kikiras, "On the security and privacy of Internet of Things architectures and systems," in *Proc. Int. Workshop Secure Internet Things (SIoT)*, 2015, pp. 49–57.
[286] C. Perera, R. Ranjan, L. Wang, S. U. Khan, and A. Y. Zomaya, "Big data privacy in the Internet of Things era," *IT Prof.*, vol. 17, no. 3, pp. 32–39, May/Jun. 2015.
[287] F. Brandherm, L. Wang, and M. Mühlhäuser, "A learning-based framework for optimizing service migration in mobile edge clouds," in *Proc. 2nd Int. Workshop Edge Syst., Anal. Netw. (EdgeSys)*, 2019, pp. 12–17.
[288] E. Ahmed, A. Ahmed, I. Yaqoob, J. Shuja, A. Gani, M. Imran, and M. Shoaib, "Bringing computation closer toward the user network: Is edge computing the solution?" *IEEE Commun. Mag.*, vol. 55, no. 11, pp. 138–144, Nov. 2017.
[289] A. Reznik, A. Sulisti, A. Artemenko, Y. Fang, D. Frydman, F. Giust, H. Lv, S. U. Sheikh, Y. Yu, and Z. Zheng, "MEC in an enterprise setting: A solution outline," ETSI, Sophia Antipolis, France, White Paper 30, Sep. 2018, pp. 1–20.
[290] J. Shi, L. Gui, D. Koutsonikolas, C. Qiao, and G. Challen, "A little sharing goes a long way: The case for reciprocal WiFi sharing," in *Proc. 2nd Int. Workshop Hot Topics Wireless (HotWireless)*, 2015, pp. 6–10.
[291] L. Mamatas, I. Psaras, and G. Pavlou, "Incentives and algorithms for broadband access sharing," in *Proc. ACM SIGCOMM Workshop Home Netw. (HomeNets)*, 2010, pp. 19–24.
[292] M. Zeng, Y. Li, K. Zhang, M. Waqas, and D. Jin, "Incentive mechanism design for computation offloading in heterogeneous fog computing: A contract-based approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
[293] Y. Liu, C. Xu, Y. Zhan, Z. Liu, J. Guan, and H. Zhang, "Incentive mechanism for computation offloading using edge computing: A Stackelberg game approach," *Comput. Netw.*, vol. 129, pp. 399–409, Dec. 2017.

JULIEN GEDEON received the B.Sc. and M.Sc. degrees from Technische Universität Darmstadt, in 2013 and 2015, respectively, where he is currently pursuing the Ph.D. degree with the Telecooperation Laboratory, under the supervision of Prof. M. Mühlhäuser. Since December 2018, he has been co-heading the Smart Ubiquitous Networks (SUN) Group, Telecooperation Laboratory, Technische Universität Darmstadt. The group is deeply involved in a DFG-Funded Collaborative Research Center on the future Internet. His research interests include edge and fog computing, serverless computing, mobile cloud computing, software-defined networking, wireless sensor networks, and smart cities.

FLORIAN BRANDHERM received the B.Sc. degree in computer science and the M.Sc. degree in autonomous systems from Technische Universität Darmstadt, Germany, in 2015 and 2018, respectively, where he is currently pursuing the Ph.D. degree under the supervision of Prof. M. Mühlhäuser. He is a part of the Smart Ubiquitous Networks (SUN) Group, Prof. Mühlhäuser's Telecooperation Laboratory, and associated with the collaborative research center investigating multi-mechanism adaptation for the future Internet (MAKI). His current research interests include mobile edge computing, in-network processing, distributed resource allocation, game theory, and reinforcement learning.

ROLF EGERT is currently pursuing the Ph.D. degree in smart protection in infrastructures and networks (SPIN) with the Telecooperation Laboratory, Technische Universität Darmstadt, under the direction of Prof. M. Mühlhäuser. His current research interests include IT-security aspects in the domain of smart energy grids, bio-inspired optimization approaches, and resilient complex system infrastructures.

TIM GRUBE received the M.Sc. degree in computer science with a focus on distributed systems from Technische Universität Darmstadt, in 2014, and the Ph.D. (Dr. rer. nat.) degree with a focus on efficient anonymous communication, in 2018. Since 2018, he has been coordinating the security research of Prof. M. Mühlhäuser's Telecooperation Laboratory, Smart Protection in Infrastructures and Networks (SPIN) Group, Technische Universität Darmstadt. Since 2019, he has also been coordinating the research training group (doctoral college/Graduiertenkolleg) "Privacy and Trust for Mobile Users." His current research interests include privacy and trust in mobile, heterogeneous environments, and the application of machine learning for privacy assessment and privacy education.

MAX MÜHLHÄUSER is currently a Full Professor with Technische Universität Darmstadt, where he is also the Head of the Telecooperation Laboratory. He holds key positions in several large collaborative research centers and is leading the Doctoral School on Privacy and Trust for Mobile Users. He and his lab members conduct research on the future Internet, human–computer interaction and cybersecurity, and privacy and trust. He founded and managed industrial research centers and worked as either a Professor or a Visiting Professor at universities in Germany, USA, Canada, Australia, France, and Austria. He is a member of acatech, the German Academy of the Technical Sciences. He was and is active in numerous conference program committees, as an Organizer for several annual conferences and as a member of editorial boards or a Guest Editor for journals, such as IMWUT, *Pervasive Computing*, *ACM Multimedia*, and *Pervasive and Mobile Computing*.