Towards Enabling Novel Edge-Enabled Applications*

João Leitão, Pedro Ákos Costa, Maria Cecília Gomes, and Nuno Preguiça
jc.leitao@fct.unl.pt, pah.costa@campus.fct.unl.pt, {mcg, nuno.preguica}@fct.unl.pt
NOVA LINCS & DI-FCT-UNL

Abstract

Edge computing has emerged as a distributed computing paradigm to overcome practical scalability limits of cloud computing. The main principle of edge computing is to leverage on computational resources outside of the cloud for performing computations closer to data sources, avoiding unnecessary data transfers to the cloud and enabling faster responses for clients.

While this paradigm has been successfully employed to improve response times in some contexts, mostly by having clients perform pre-processing and/or filtering of data, or by leveraging on distributed caching infrastructures, we argue that the combination of edge and cloud computing has the potential to enable novel applications. However, to do so, some significant research challenges have to be tackled by the computer science community. In this paper, we discuss different edge resources and their potential use, motivated by envisioned use cases. We then discuss concrete research challenges that once overcome, will allow to realize our vision. We also discuss potential benefits than can be obtained by exploiting the hybrid cloud/edge paradigm.

1 Introduction

Since its inception in 2005, cloud computing has deeply impacted how distributed applications are designed, implemented, and deployed. Cloud computing offers the illusion of infinite resources available in data centers, whose usage can be elastically adapted to meet the needs of applications. Furthermore, data centers in different geographical locations enable application operators to provide better quality of service for large numbers of users scattered around the world by leveraging on geo-distribution and geo-replication.

Cloud computing however, is not a panacea for building reliable, available, and efficient distributed systems. In particular, the increasing popularity of Internet of Things (IoT) and Internet of Everything (IoE) applications, combined with an increase in mobile and user-centric applications, has lead to a significant increase in the quantity of data being produced by application clients. Although cloud computing infrastructures are highly scalable, the time required to process such large amounts of data is becoming prohibitively high. Additionally, the network capacity between clients and data centers is now becoming a significant bottleneck for such applications, namely to timely push data and fetch computation results to, and from the cloud.

Due to this, moving computations towards the edge of systems (i.e, closer to the clients that effectively process and consume data) has become an essential endeavor to sustain the growth of such applications. This led to the emergence of *edge computing*. Edge computing can be defined, in very broad terms, as performing computations outside the boundaries of data centers [43]. Many approaches have already leveraged on some form of edge computing to improve the latency perceived by end-users, such as CDNs [54], or tapping into resources of client devices [41, 52], among others.

This has motivated the emergence of proposals for taking advantage of edge computing. In particular Cisco has proposed the model of Fog Computing [5] which aims at improving the overall performance of IoT applications by collocating servers (and network equipment with computing capacity) with sensors that generate large amounts of data. These (Fog) servers can then pre-process data enabling timely reaction to variations on the sensed data, and filter the relevant information that is propagated towards cloud infrastructures for further processing. Mist computing, is an evolution of the Fog computing model, that has been adopted by industry [7] and that, in its essence, proposed to push computation towards sensors in IoT applications, which enables sensors themselves to perform data filtering computations, alleviating the load imposed on Fog and Cloud servers. While these novel architectures exploit the potential of edge computing, they do so in a limited way, requiring specialized hardware and not taking a significant advantage of computational devices that already exist in the edge. Furthermore, and as noted

*The work presented here was partially supported by the Lightkone European H2020 Project (under grant number 732505) and NOVA LINCS (through the FC&T grant UID/CEC/04516/2013).

for instance in [5] and [7] all of these proposed architectures are highly biased towards IoT applications.

In this paper, we argue that edge computing also offers the opportunity to build new *edge-enabled* applications, whose use of edge resources go beyond what has been done in the past, and in particular beyond proposals such as Fog and Mist computing [5, 7]. Previous authors have already presented their visions for the future of Edge computing [43, 48], Fog computing [47, 35, 38], and IoT specific edge challenges [31]. These works however, present their visions with an emphasis on IoT applications. An exception to this is related with Mobile edge computing [34] which devotes itself to the close cooperation of mobile devices to offload pressure from the cloud. Contrary to these, we take a different approach on edge computing and envision a future where general user-centric applications are supported by a myriad of different and already existing edge resources. In particular, we believe that edge computing will enable the creation of significantly more complex distributed applications, both in terms of their capacity to handle client request and processing data, and also in terms of the number of components. Our vision, is that this will empower the design of user-centric applications that promote additional interactivity among users and between users and their (intelligent) environment.

To realize this vision however, it is relevant to fully understand what are the computational and network resources available for edge-enabled applications, their characteristics, and how they can be used (§2). We further materialize our vision on the potential of edge computing by presenting two envisioned case studies (§3). Using large numbers of heterogeneous edge resources to build novel edge-enabled applications is not trivial, and key research challenges must be addressed by the computer systems community, we further relate these with previous research (§4)[1]. Finally, we conclude this paper by summarizing the main research challenges that are in the way to fully realizing the potential of edge computing (§5).

2 Overview of the Edge

To fully realize the potential of edge computing, one should identify which computational resources lie beyond the cloud boundary, and what are the limitations of such devices and their potential benefits for edge-enabled applications. Figure 1 provides a visual representation of the different edge resources that we envision. We represent these edge resources as being organized in different levels starting with level zero that represents cloud data centers. Edge-enabled applications are not however, required to make use of resources in all edge levels.

To better *characterize* the different levels in the edge resource spectrum, we consider three main dimensions: $i)$ **capacity**, which refers to the processing power, storage capacity, and connectivity of the device; $ii)$ **availability**, which refers to the probability of the resource to be reachable (due to being continuously active or faults); and $iii)$ **domain**, which captures if the device supports the operation of an edge-enabled application as a whole (*application domain*) or just the activities of a given user[2] within an edge-enabled application (*user domain*).

We further classify the *potential uses* of the different edge resources considering two main dimensions: $i)$ **storage**, which refers to the ability of an edge resource to store and serve application data. Devices that can provide storage can do so by either storing *full application state*, *partial application state*, or by providing *caching*. The first two enable state to be modified by that resource, and the later only enables reading (of potentially stale) data; and $ii)$ **computation**, which refers to the ability of performing data processing. Here, we consider three different classes of data processing, from the more general to the more restrictive: *generic computations*, *aggregation and summarization*, and *data filtering*.

We start by observing that as we move farther from the cloud (i.e, to higher edge levels), the capacity and availability of each individual resource tends to decrease, while the number of devices increases. We now discuss each of these edge resources in more detail. We further note that resources could be presented with different granularity however, in this paper we focus on a presentation that allows to distinguish computational resources in terms of their properties and potential uses within the scope of future edge-enabled applications.

E0: Cloud Data Centers Cloud data centers offer pools of computational and storage resources that can be dynamically scaled to support the operation of edge-enabled applications. The existence of geo-distributed locations can be used as a first edge computing level, by enabling data and computations to be performed at the data center closest to the client. These resources have *high capacity and availability* and operate at the *application domain*. They offer the possibility for storing *full application state* and perform *generic computations*.

E1: ISP Servers & Private Data Centers This edge resource represents regional private data centers and dedicated servers located in Internet Service Providers (ISPs) or exchange points that can operate over data produced by users in a particular area. These servers operate at the *application domain*, presenting *large capacity* and *high availability*. They offer the possibility to store *(large) partial application state* and perform *generic computations*.

E2: 5G Towers The new advances in mobile networks

[1] We recognize that fully tapping into the potential of edge computing requires concerted research efforts from many fields in computer science, however in this paper we focus only on a computer systems perspective.

[2] We refer to user in broad terms, meaning an entity that uses an edge-enabled application, either an end-user or a company.

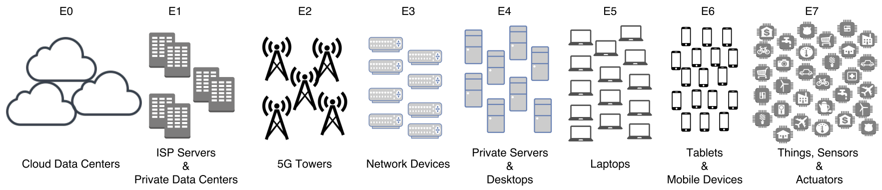

Figure 1: Edge Components Spectrum

		E0 Cloud DCs	E1 ISP Servers & Priv. DCs	E2 5G Towers	E3 Network Devices	E4 Priv. Servers & Desktops	E5 Laptops	E6 Tablets & Mobiles	E7 Things
Characterization	**Capacity**	High	Large	Medium	Low	Medium	Medium	Low	Varied
	Availability	High	High	High	High	Medium	Low	Low	Limited
	Domain	Application	Application	Application	Application	User	User	User	User
Potential uses	**Storage**	Full State	(Large) Partial	(Limited) Partial	None/Caching	(User) Partial	Caching	(User) Caching	(Local) Caching
	Computation	Generic	Generic	Generic	Filtering	Generic	Aggregation	Aggregation	Filtering

Table 1: Edge Devices Per Level Characteristics

will introduce processing and storage power in towers that serve as access points for mobile devices (and tablets) as well as improved connectivity. While we can expect these edge resources to have *medium capacity*, they should have *high availability* operating at the *application domain*. These computational resources can execute *generic computations* over stored *limited partial application state* enabling further interactions among clients (e.g, mobile devices) in close vicinity.

E3: Network Devices Network devices (such as routers, switches, and access points) that have processing power capabilities, offer *low capacity* and *high availability*. From the storage perspective, these offer either none or *caching* capacity. Devices in close vicinity of the user will operate at the *user domain* while equipment closer to servers might operate at the *application domain*. These devices will mostly enable in-network processing for edge-enabled applications in the form of *data filtering* activities over data produced by client devices being shipped towards the center of the system.

E4: Private Servers & Desktops This is the first layer (and more powerful in terms of capacity) of devices operating exclusively in the *user domain*. Private servers and desktop computers can easily operate as logical gateways to support the interaction and perform computations over data produced by levels E5-E7. While individual edge resources have *medium capacity* and *medium availability* they can easily perform more sophisticated computing tasks if the resources of multiple devices are combined together. These edge resources are expected to store *(user-specific) partial application state* while enabling *generic computations* to be performed. Private servers in this context are equivalent to *in-premises servers* frequently referred as part of Fog computing architectures [5].

E5: Laptops User laptops are similar to resources in the E4 level albeit, with *low availability*. Low availability in this context is mostly related with the fact that the uptime of laptops can be low due to the user moving from location to location. Due to this, we expect these devices to be used for performing *aggregation and summarization* computations and eventually provide *(user-specific) caching* of data for components running farther from the cloud. Laptops might act as application interaction portals, enabling users to use such devices to directly interact with edge-enabled applications.

E6: Tablets & Mobile Devices Tablets and Mobile devices are nowadays preferred interaction portals, enabling users to access and interact with applications. We expect this trend to become dominant for new edge-enabled applications since users expect continuous and ubiquitous access to applications. These devices have *low capacity* and *low availability*, the latter is mostly justified by the fact that the battery life of these devices will shorten significantly if the device is used to perform continuous computations. These devices however, can be used as logical gateways for devices in the E7 level in the *user domain* context. They can provide *user-specific caching* storage and perform either *aggregation and summarization* or *data filtering* for data produced by E7 devices in the context of a particular user.

E7: Things, Sensors, & Actuators These are the most limited devices in our edge resource spectrum. These devices will act in edge-enabled applications mostly as data producers and consumers. They have *extremely limited capacity* and *varied availability* (in some cases low due to limited power and weak connectivity). They operate in the *user domain*, and can only provide extremely limited forms of *caching* for edge-enabled applications. Due to their limited processing power they are restricted to perform *data filtering* computations. Devices in the E7 layer with computational capacity are the basis for Mist computing architectures [7].

Table 1 summarizes the different characteristics and potential uses of edge resources at each of the considered levels. We expect application data to flow along the edge resource spectrum although, different data might be processed differently at each level (or skip some entirely).

3 Envisioned Case Studies

We now briefly discuss two envisioned case studies of novel edge-enabled applications, and argue how edge resources in different levels of the edge spectrum can be leveraged to enable or improve these case studies.

Mobile Interactive Multiplayer Game Consider an augmented reality mobile game that allows players to use their mobile devices to interact with augmented reality objects and non-playing characters similar to the popular Pókemon Go game [3]. Such game could enable direct interactions among players, (e.g, to trade game objects or fight against each other) and allow players to interact in-game with (local) third party businesses that have agreements with the company operating the game (e.g, a coffee shop that offers in-game objects to people passing by their physical location).

Pókemon Go does not allow these interactions, with some evidences [6] pointing to one of the main reasons being the inability of cloud-based servers to support such interactions in a timely manner due to large volumes of traffic produced by the application. However, edge computing offers the possibility to enable such interactions, by leveraging on edge resources on some of the levels discusses above. Considering that the game is accessed primarily through mobile phones, one could resort to computational and storage capabilities of *5G Towers (E2)* to mediate direct interactions (e.g, fights) between players. One could also leverage on regional *ISP and Private Data centers (E1)* to manage high throughput of write operations (and inter-player transactions) to enable trading objects. Some trades could actually be achieved by having transaction executed directly between the *Tablets & Mobile Devices (E6)* of players and synchronizing operations towards the *Cloud (E0)* later. Special game features provided by third party businesses could be supported by *Private servers (E4)* being accessed through local networks (supported by *Network Devices (E3)*) located on business premises.

Intelligent Health Care Services Consider an integrated and intelligent medical service that inter-connects patients, physicians (in hospitals and treatment centers), and emergency response services [4], that can leverage on wearable devices (e.g, smart watches or medical sensors), among other IoT devices (e.g, smart pills dispensers), to provide better health care including, handling medical emergencies, and tracking health information in the scope of a city, region, or country.

These systems are not a reality today due to, in our opinion, two main factors. The first is the large amounts of data produced by a large number of health monitors, the second is related with privacy issues regarding the medical data of individual patients. Edge computing and the clever usage of different edge resources located in different levels (as discussed previously) can assist in realizing such application. In particular, *Wearables and medical sensors (E7)* can cooperate among them and interact with users' *Mobile Devices (E6)* and *Laptops (E5)*, which can archive and perform simple analysis over gathered data. The analysis of data in these levels could trigger alerts, to notify the user to take medicine, to report unexpected indicators, or to contact emergency medical services if needed. This data could be further encrypted and uploaded to *Private Servers (E4)* of hospitals, so that physicians could follow their patients' conditions. Additionally, health indicators aggregates could be anonymously uploaded to *Private Data Centers (E1)* for further processing, enabling monitoring at the level of cities, regions, or countries to identify pandemics or to co-related them with environmental aspects.

4 Research Challenges

The presented case studies (§3) rely on the use of multiple edge layers as discussed previously (§2). Other novel edge-enabled applications will have similar requirements. Some of the most challenging aspects of the edge is its high heterogeneity in terms of capacity, and that one has to deal with the increasing number of devices. We identify the following main challenges to fully realize the potential of edge computing, which we also discuss in relation to our envisioned case studies.

Resource management: Resource management solutions are crucial to keep track and manage computational resources across multiple edge levels. Solutions must provide efficient mechanisms to allow the dynamic creation and decommission of application components across multiples resources, allowing these components to interact efficiently. Considering the use case of a *mobile interactive multiplayer game* this translates in two complementary aspects. The first is to track computational resources to enable executing components of the game in cloud platforms, ISP and private data centers, or private servers located in Coffee shops. The second is related with mobile devices in close vicinity finding each other and interacting, either directly or through 5G towers.

Large-scale decentralized resource tracking and management has been previously addressed in the context of decentralized peer-to-peer systems [41]. Overlay networks [26] have been used to enable the tracking and communication among large numbers of resources [28, 27, 51] and also to enable efficient application-level routing [44, 42] and assignment of responsibilities in a decentralized way [22]. Solutions such as Mesosphere [3] and Yarn [49], that are specially tailored for cloud and clus-

[3] `https://www.pokemongo.com/`

[4] A significative evolution of the Denmark Medical System briefly described in [45].

ter environments, can be employed for resource management in edge levels closer to the cloud, potentially complemented with other solutions in different levels. However, most of these solutions assume resources to be homogeneous in terms of capacity and connectivity which makes them unsuitable for edge environments. Since edge resources are located in different levels, hierarchical management of resources could be an interesting approach, feasible at the overlay management protocols level.

Enable computations to move: Edge-enabled applications require computations to be executed across heterogeneous edge resources located in different levels. Computations cannot be executed in arbitrary edge resources (i.e, any edge level), and shipping computations across different edge levels must cope with heterogeneous execution environments (e.g, virtualization, containers, middleware, different operating systems, etc). It is therefore relevant to develop solutions that enable the migration of (generic) computational tasks among different (compatible) edge levels, and also to allow computations to be decomposed into simpler computational tasks, and symmetrically, be recomposed as single processing units dynamically. For instance in the *intelligent heath care services* examples this is relevant to allow computations that perform initial analysis of data gathered by wearable sensors to migrate between patients mobile devices and laptops according to their availability and available battery.

This is not a new proposal, in fact, research from the software systems community on osmotic computing already proposed the migration of microservices that encode fractions of the computational logic of applications to the edge and back to cloud infrastructures according to evolving workloads [50]. Mobile agents [14] and mobile code solutions [20] are proposals that allow having arbitrary code move and execute along a (logical) network of components. However, none of these approaches deal with resource heterogeneity nor the decomposition of computations in simpler tasks (and their coordination). Amazon [1] and Google [2] have enriched their cloud infrastructure to pre-process and redirect HTTP requests to data centers closer to clients. Yet, these only operate at the E0 edge level and extending them towards the edge is an open challenge.

Dynamic and partial state replication: Edge-enabled applications will naturally need to perform computations over application data; however, as the computations can be scattered through multiple edge levels, and to avoid communication overheads with the core of network, application state should be able to move towards the edge. This brings additional challenges, as dynamically spawning replicas of data storage systems supporting edge-enabled applications will inevitably lead to an increase in the overhead produced by replication protocols, requiring also the automatic decommission of (unuseful) replicas. Furthermore, not all edge resources can hold the same amount of application data, which motivates the need for effective partial replication solutions, and replication protocols that, potentially, provide different consistency guarantees at different edge levels. Consider the use case of the *mobile interactive multiplayer game*, in this context 5G towers can provide better quality of service for direct interactions among users if they replicate a fraction of the application state for users in its vicinity. To do this however, it is essential to be able to freely spawn and remove (to minimize operational costs) partial replicas of the application state.

There have been many recent works exploring geo-replication, where replicas are dispersed in remote locations. Some focus on offering strong consistency guarantees [37] while others focus on providing causal+ consistency as to ensure availability [32, 33, 8, 53]. However, few solutions exploit the use of partial replication. The ones that do so, such as Saturn [13] and Kronos [16] are highlly limited in terms of scalability and hence will not be able to cope with large number of replicas managed dynamically. There has also been few works exploring the combination of multiple consistency models. Gemini [30] and Indigo [10] do so per operation type over the data store instead of per replica. Although, none of these approaches entirely address the data management requirements for future edge-enabled applications.

Lightweight and decentralized monitoring: Dynamically migrating computations and storage components along the edge spectrum requires knowledge about available edge resources. This knowledge can be attained by distributed monitoring systems that obtain information regarding applications' operation and the load of edge resources, which can then be employed to perform adequate management decisions. Efficient dissemination of monitoring information will require in-network processing for filter and aggregate data. Again picking up the example of moving the preliminary analysis of sensors data between the mobile devices and laptops of patients in the *intelligent health care services* use case, to decide the best device to conduct computations requires having (somewhat) up-to-date information regarding the status of each device, namely if the device is active, if it has available CPU and RAM to perform the computations efficiently, and its current battery level (or if the device is currently connected to a power source).

There are several previous works that focus on decentralized monitoring of large-scale platforms, typically on cloud and grid infrastructures [9, 46, 29]. Many of these solutions resort to gossip-based dissemination protocols [27, 12] to propagate relevant information towards special sink nodes. While these solutions offer an interesting departing point for devising new monitoring schemes, they do not consider the monitoring of heterogeneous resources organized in hierarchies as captured in our edge resource spectrum. In addition, monitoring tasks and in-network processing, could change the grain of execution dynamically as to adjust to variations in the workload and available resources in the system.

Enable systems to become autonomic: The management of the life cycle and interactions of large numbers of edge components operating at resources scattered throughout multiple edge levels will be unfeasible by hand, particularly when considering the need to timely adapt the operation of the system in reaction to unpredictable failures of components or sudden surges in access patterns (i.e, peak loads). A possible way to overcome this challenge is to design novel mechanisms to enrich edge-enable applications to have autonomic management capabilities. To achieve this, we require the capacity to enable moving, decomposing, and/or replicating computations, novel dynamic partial replication schemes, and lightweight distributed monitoring. Considering the use of 5G towers in the *mobile interactive multiplayer game*, spawning application computational components and partial replicas of the game state is manually unfeasible in a timely way considering the potentially large number of available 5G towers. Such process requires autonomic control that, based on monitoring information, can automatically trigger application management mechanisms.

Autonomic systems have been proposed and studied since 2001 [4] and have been applied to multiple types of systems from web services [19], to management of grid resources and others [24]. Typically autonomic systems are designed around the $MAPE$-K architectural design, where the system has Monitoring, Analysis, Planning, and Execution components that are interconnected by a common Knowledge base. The main challenge in making edge-enabled applications autonomic is due to the typical central nature of the Planning (and potentially Analysis and Execution) components. Large-scale edge-enabled applications combining multiple components scattered across several edge levels will require decentralized planning schemes, capable of operating (and executing) reconfigurations of the system with incomplete and partial knowledge. While some previous works have explored this [25], they are still far from meeting the needs of future edge-enabled applications.

Security & Data protection: Future edge-enabled applications will manipulate sensitive user data. Doing so can compromise users' privacy. Furthermore, executing computations and storing data in hardware controlled by individual users can also compromise data integrity. To overcome these challenges new data protection schemes have to be developed. Considering the use case of the *intelligent health care services* where medical data would be stored in user devices and in private servers in hospitals. This data is highly sensitive as it can easily compromise the privacy of patients, while compromising its integrity could have serious medical implications, for instance, if a patient receives the incorrect treatment.

Homomorphic [21] and partially homomorphic [40, 15] encryption schemes allow computations to be performed in the encrypted domain. However, these solutions have high computational cost and cipher-text expansion therefore, being unsuitable to support edge computations. A more promising alternative is to use schemes based on (efficient) symmetric cryptography, enabling some operations (such as indexing and search) in the encrypted domain. This has been demonstrated for a few data formats [18, 17]. Nevertheless, these schemes do not provide guarantees of data integrity. Protecting data integrity can rely on trusted hardware [36] (e.g,IntelSGX [11]) that provide attestation and verification mechanisms for outsourced computations. Unfortunately, limited resources and complex key management/distribution schemes, makes the use of trusted hardware an open challenge. The blockchain design [39, 23] offers an interesting approach to make edge-enabled applications more robust, but still lack adequate scalability and efficiency.

5 Conclusion

In this paper we provide our own perspective on the future of edge-computing. We categorize edge-enabled applications as applications that can resort to a myriad of computational resources outside of cloud environments, leveraging on their different properties for distinct purposes. We argue that this vision will enable the emergence of a novel class of edge-enabled applications. To motivate this, we presented two envisioned case studies.

While we note that the future of edge computing requires efforts from all computer science fields, from a systems perspective, we identified a set of key research challenges and related them with previous contributions. These include: i) decentralized and scalable resource management schemes; ii) enable the migration, replication, and decomposition of computational tasks along edge resources; iii) develop dynamic and scalable replication schemes, leveraging on partial replication and capable of extending towards the edge; iv) develop lightweight and scalable monitoring schemes; v) enable systems to become autonomic and self-adapt to dynamic resource availability and workloads; and finally, vi) develop novel cryptographic and computational schemes for data privacy and integrity.

References

[1] Aws lambda@edge. `https://docs.aws.amazon.com/lambda/latest/dg/`

lambda-edge.html. Accessed: 2018-04-24.

[2] Google cloud edge architecture. https://peering.google.com/#/infrastructure. Accessed: 2018-04-24.

[3] Mesosphere. https://mesosphere.com/product/. Accessed: 2018-04-24.

[4] An architectural blueprint for autonomic computing. Technical report, IBM, June 2005.

[5] Cisco. Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are. https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf, 2015. Accessed: 2018-05-16.

[6] Hu Yan (Huawei iLab). Research Report on Pokémon Go's Requirements for Mobile Bearer Networks. http://www.huawei.com/~/media/CORPORATE/PDF/ilab/05-en, 2016. Accessed: 2018-05-16.

[7] Raka Mahesa (IBM). FHow cloud, fog, and mist computing can work together. https://developer.ibm.com/dwblog/2018/cloud-fog-mist-edge-computing-iot/, 2018. Accessed: 2018-05-16.

[8] S. Almeida, J. Leitão, and L. Rodrigues. Chainreaction: A causal+ consistent datastore based on chain replication. In *Proceedings of the 8th ACM European Conference on Computer Systems*, EuroSys '13, pages 85–98, New York, NY, USA, 2013. ACM.

[9] L. Baduel and S. Matsuoka. A decentralized, scalable, and autonomous grid monitoring system. In E. Tovar, P. Tsigas, and H. Fouchal, editors, *Principles of Distributed Systems*, pages 1–15, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[10] V. Balegas, S. Duarte, C. Ferreira, R. Rodrigues, N. Preguiça, M. Najafzadeh, and M. Shapiro. Putting consistency back into eventual consistency. In *Proceedings of the Tenth European Conference on Computer Systems*, EuroSys '15, pages 6:1–6:16, New York, NY, USA, 2015. ACM.

[11] M. Barbosa, B. Portela, G. Scerri, and B. Warinschi. Foundations of hardware-based attested computation and application to sgx. In *EuroS&P*, pages 245–260. IEEE, 2016.

[12] K. P. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky. Bimodal multicast. *ACM Transactions on Computer Systems*, 17(2):41–88, May 1999.

[13] M. Bravo, L. Rodrigues, and P. Van Roy. Saturn: A distributed metadata service for causal consistency. In *Proceedings of the Twelfth European Conference on Computer Systems*, EuroSys '17, pages 111–126, New York, NY, USA, 2017. ACM.

[14] R. S. Chowhan and R. Purohit. Study of mobile agent server architectures for homogeneous and heterogeneous distributed systems. *International Journal of Computer Applications*, 156(4):32–37, Dec 2016.

[15] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, Jul 1985.

[16] R. Escriva, A. Dubey, B. Wong, and E. G. Sirer. Kronos: The design and implementation of an event ordering service. In *Proceedings of the Ninth European Conference on Computer Systems*, EuroSys '14, pages 3:1–3:14, New York, NY, USA, 2014. ACM.

[17] B. Ferreira, J. Leitão, and H. Domingos. Multimodal indexable encryption for mobile cloud-based applications. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 213–224, June 2017.

[18] B. Ferreira, J. Rodrigues, J. Leitao, and H. Domingos. Practical privacy-preserving content-based retrieval in cloud image repositories. *IEEE Transactions on Cloud Computing*, PP(99):1–1, 2017.

[19] J. Ferreira, J. Leitão, and L. Rodrigues. A-osgi: A framework to support the construction of autonomic osgi-based applications. In *Proceedings of the Third International ICST Conference on Autonomic Computing and Communication Systems*, page (to appear), Limassol, Cyprus, Sept. 2009.

[20] A. Fuggetta, G. P. Picco, and G. Vigna. Understanding code mobility. *IEEE Transactions on Software Engineering*, 24(5):342–361, May 1998.

[21] C. Gentry, S. Halevi, and N. P. Smart. Homomorphic evaluation of the aes circuit. In R. Safavi-Naini and R. Canetti, editors, *Advances in Cryptology – CRYPTO 2012*, pages 850–867, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[22] I. Gupta, K. Birman, P. Linga, A. Demers, and R. van Renesse. Kelips: Building an efficient and stable p2p dht through increased memory and background overhead. In M. F. Kaashoek and I. Stoica, editors, *Peer-to-Peer Systems II*, pages 160–169,

Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[23] A. Hari and T. V. Lakshman. The internet blockchain: A distributed, tamper-resistant transaction framework for the internet. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, HotNets '16, pages 204–210, New York, NY, USA, 2016. ACM.

[24] M. C. Huebscher and J. A. McCann. A survey of autonomic computing—degrees, models, and applications. *ACM Comput. Surv.*, 40(3):7:1–7:28, Aug. 2008.

[25] J. O. Kephart. Engineering decentralized autonomic computing systems. In *Proceedings of the Second International Workshop on Self-organizing Architectures*, SOAR '10, pages 1–2, New York, NY, USA, 2010. ACM.

[26] J. Leitão. *Topology Management for Unstructured Overlay Networks*. PhD thesis, Technical University of Lisbon, Sept. 2012.

[27] J. Leitao, J. Pereira, and L. Rodrigues. Epidemic broadcast trees. In *2007 26th IEEE International Symposium on Reliable Distributed Systems (SRDS 2007)*, pages 301–310, Oct 2007.

[28] J. Leitão, J. Pereira, and L. Rodrigues. Hyparview: A membership protocol for reliable gossip-based broadcast. In *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07)*, pages 419–429, June 2007.

[29] J. Leitão, L. Rosa, and L. Rodrigues. Large-scale peer-to-peer autonomic monitoring. In *GLOBECOM Workshops*, pages 1–5. IEEE, November 2008.

[30] C. Li, D. Porto, A. Clement, J. Gehrke, N. Preguiça, and R. Rodrigues. Making geo-replicated systems fast as possible, consistent when necessary. In *Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation*, OSDI'12, pages 265–278, Berkeley, CA, USA, 2012. USENIX Association.

[31] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5):1125–1142, Oct 2017.

[32] W. Lloyd, M. J. Freedman, M. Kaminsky, and D. G. Andersen. Don't settle for eventual: Scalable causal consistency for wide-area storage with cops. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pages 401–416, New York, NY, USA, 2011. ACM.

[33] W. Lloyd, M. J. Freedman, M. Kaminsky, and D. G. Andersen. Stronger semantics for low-latency geo-replicated storage. In *Proceedings of the 10th USENIX Conference on Networked Systems Design and Implementation*, nsdi'13, pages 313–328, Berkeley, CA, USA, 2013. USENIX Association.

[34] P. Mach and Z. Becvar. Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys Tutorials*, 19(3):1628–1656, thirdquarter 2017.

[35] R. Mahmud, R. Kotagiri, and R. Buyya. *Fog Computing: A Taxonomy, Survey and Future Directions*, pages 103–130. Springer Singapore, Singapore, 2018.

[36] F. McKeen, I. Alexandrovich, A. Berenzon, C. V. Rozas, H. Shafi, V. Shanbhogue, and U. R. Savagaonkar. Innovative instructions and software model for isolated execution. In *Proceedings of the 2Nd International Workshop on Hardware and Architectural Support for Security and Privacy*, HASP '13, pages 10:1–10:1, New York, NY, USA, 2013. ACM.

[37] H. Moniz, J. Leitão, R. J. Dias, J. Gehrke, N. Preguiça, and R. Rodrigues. Blotter: Low latency transactions for geo-replicated storage. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, pages 263–272, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[38] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos. A comprehensive survey on fog computing: State-of-the-art and research challenges. *IEEE Communications Surveys Tutorials*, 20(1):416–464, Firstquarter 2018.

[39] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system, 2009.

[40] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In J. Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, pages 223–238, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[41] R. Rodrigues and P. Druschel. Peer-to-peer systems. *Commun. ACM*, 53(10):72–82, Oct. 2010.

[42] A. Rowstron and P. Druschel. Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In R. Guerraoui, editor,

Middleware 2001, pages 329–350, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

[43] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE Internet of Things Journal*, 3(5):637–646, Oct 2016.

[44] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. *SIGCOMM Comput. Commun. Rev.*, 31(4):149–160, Aug. 2001.

[45] G. Tomás, P. Zeller, V. Balegas, D. Akkoorath, A. Bieniusa, J. a. Leitão, and N. Preguiça. Fmke: A real-world benchmark for key-value data stores. In *Proceedings of the 3rd International Workshop on Principles and Practice of Consistency for Distributed Data*, PaPoC '17, pages 7:1–7:4, New York, NY, USA, 2017. ACM.

[46] R. Van Renesse, K. P. Birman, and W. Vogels. Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining. *ACM Trans. Comput. Syst.*, 21(2):164–206, May 2003.

[47] L. M. Vaquero and L. Rodero-Merino. Finding your way in the fog: Towards a comprehensive definition of fog computing. *SIGCOMM Comput. Commun. Rev.*, 44(5):27–32, Oct. 2014.

[48] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos. Challenges and opportunities in edge computing. In *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, pages 20–26, Nov 2016.

[49] V. K. Vavilapalli, A. C. Murthy, C. Douglas, S. Agarwal, M. Konar, R. Evans, T. Graves, J. Lowe, H. Shah, S. Seth, B. Saha, C. Curino, O. O'Malley, S. Radia, B. Reed, and E. Baldeschwieler. Apache hadoop yarn: Yet another resource negotiator. In *Proceedings of the 4th Annual Symposium on Cloud Computing*, SOCC '13, pages 5:1–5:16, New York, NY, USA, 2013. ACM.

[50] M. Villari, M. Fazio, S. Dustdar, O. Rana, and R. Ranjan. Osmotic computing: A new paradigm for edge/cloud integration. *IEEE Cloud Computing*, 3(6):76–83, Nov 2016.

[51] S. Voulgaris, D. Gavidia, and M. van Steen. Cyclon: Inexpensive membership management for unstructured p2p overlays. *Journal of Network and Systems Management*, 13(2):197–217, Jun 2005.

[52] L. Vu, I. Gupta, K. Nahrstedt, and J. Liang. Understanding overlay characteristics of a large-scale peer-to-peer iptv system. *ACM Trans. Multimedia Comput. Commun. Appl.*, 6(4):31:1–31:24, Nov. 2010.

[53] M. Zawirski, N. Preguiça, S. Duarte, A. Bieniusa, V. Balegas, and M. Shapiro. Write fast, read in the past: Causal consistency for client-side applications. In *Proceedings of the 16th Annual Middleware Conference*, Middleware '15, pages 75–87, New York, NY, USA, 2015. ACM.

[54] M. Zhao, P. Aditya, A. Chen, Y. Lin, A. Haeberlen, P. Druschel, B. Maggs, B. Wishon, and M. Ponec. Peer-assisted content distribution in akamai netsession. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 31–42, New York, NY, USA, 2013. ACM.