# A Preliminary Roadmap for Dependability Research in Fog Computing

Zeinab Bakhshi

Mälardalen University

Sweden

zeinab.bakhshi@mdh.se

Guillermo Rodriguez-Navas

Nokia Bell Labs

Israel

guillermo.rodriguez-navas@nokia-bell-labs.com

## ABSTRACT

Fog computing aims to support novel real-time applications by extending cloud resources to the network edge. This technology is highly heterogeneous and comprises a wide variety of devices interconnected through the so-called fog layer. Compared to traditional cloud infrastructure, fog presents more varied reliability challenges, due to its constrained resources and mobility of nodes. This paper summarizes current research efforts on fault tolerance and dependability in fog computing and identifies less investigated open problems, which constitute interesting research directions to make fogs more dependable.

## CCS CONCEPTS

• **Computer systems organization → Reliability**; **Availability**; **Redundancy**;

## KEYWORDS

Fog Computing, Edge Computing, Internet of Things, Real-time, Fault tolerance, Dependability

## 1 INTRODUCTION

Fog computing is a recent computational paradigm, first introduced by Cisco, to extend cloud computing computational resources, closer to the edge of the network [7, 19]. Fog is a middle layer between the cloud and the devices to have more efficient data processing, effective analysis and storage scalability. It also reduces the amount of data transmitted to the cloud [14]. There is a general understanding that this technology is suitable for Cyber-Physical Systems, IoT and Industrial IoT (IIoT) in different application areas. For instance, smart cities, agriculture domains, vehicular systems, industrial automation, health-care and robotics. It is also claimed that fog represents a solution to improve latency for distributed control systems in general.

According to Bonomi et al. [7] fog computing has the following characteristics, a) Low latency and location awareness; b) Supports geographic distribution; c) End device mobility; d) Capacity of processing with a high number of nodes; e) Wireless access; f) Real-time applications and g) Heterogeneity. These characteristics make fog computing a suitable solution for overcoming problems manifested by the use of traditional cloud computing in Internet of Things (IoT), like high mobility and low latency, but they also give rise to new dependability challenges. Note that each of the factors mentioned above represents a difficulty for achieving dependability, so the combination of all of them makes the whole undertaking even more challenging.

Dependability is the ability of a system to supply trusted and available services. A *dependable* system is a system which is able to avoid service failures that are more frequent and more severe than is acceptable. There are many dimensions that should be considered to analyze whether a fog-based solution is dependable, such as availability, reliability, performability, maintainability; which are well-known dependability attributes (or requirements) [4]. At the same time, there are different ways to implement a dependable system, for instance using fault tolerance algorithms and redundancy techniques. Given the interest in fog computing and the difficulties it introduces in terms of dependability, it is important to understand how dependability and fault tolerance are addressed in the literature on fog computing.

This paper summarizes fog computing dependability requirements and discusses the gap, in terms of dependability, of the existing solutions with respect to the desired dependability requirements. After presenting a basic hierarchical structure of fog architecture, in this paper we will 1) identify and classify current research approaches for dependability in fog computing, 2) compare different proposed solutions considering traditional dependability notions for critical systems, and 3) discuss research gaps related to fog computing dependability. We realized that there is a range of terms alternatively used for fog computing by authors in the literature. For instance edge clouds, cloudlets, mobile edge computing, etc. We considered these terms as related technologies to fog computing in our study. The remainder of this paper is organized as follows. In Section 2 we present the fog computing architecture. In Section 3, we review current approaches for dependability solutions in fog

RTN 9, July 2019, Stuttgart, Germany

computing. In Section 4 we discuss the gaps between current research approaches and fog computing dependability requirements and finally we conclude our work in Section 5.

## 2 FOG COMPUTING ARCHITECTURE

Fog computing is a highly virtualized platform that provides storage, communication, computation, controlling, machine learning services in a decentralized network closer to devices [1, 15]. To the best of our knowledge, there is no reference architecture for fog computing, however, there are basic architectures for fog proposed in the literature, like [15, 21, 25]. The proposed architectures are mostly constituted by a three-layer structure, as depicted in Figure 1, which includes a layer between cloud and devices, known as fog layer. This fog layer carries out the task of computation from clouds closer to the network edge.

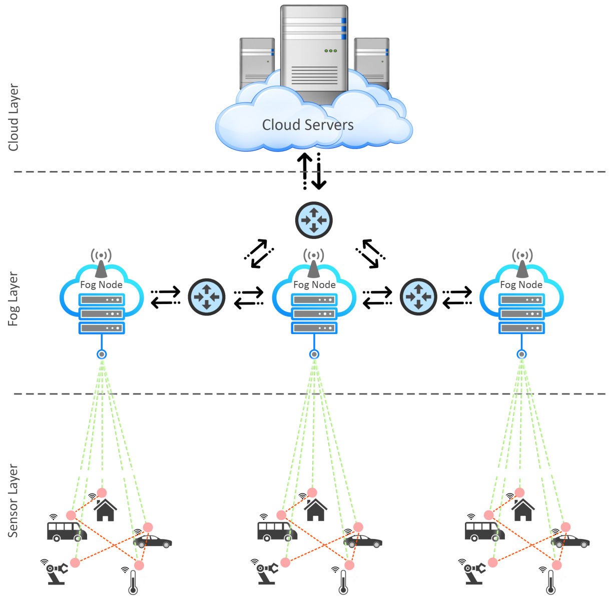

**Figure 1: The basic hierarchical architecture of fog computing.**

There is a somehow diffuse border between fog computing and the paradigm called edge computing, but there is an important difference that we will apply in this work: edge computing does not preclude the existence of a cloud to which the intermediate nodes are connected. However, whenever edge is used in combination with cloud, one can arguably say that both paradigms are equivalent. For this reason, we also investigated systems introduced as edge computing but have considered them as instances of fog.

In the following we will describe each layer of the hierarchical fog computing architecture:

### 2.1 Cloud Layer

This layer consists of multiple, powerful computational resources, storage and servers, which are capable of processing, analyzing and storing large amounts of data. Cloud computing provides services for different application domains, for instance, vehicular systems, smart cities, smart factories, health-care, etc. [25]. The clouds are efficiently managed and scheduled by some control strategies to improve utilization of the cloud resources. Although cloud computing is empowered by huge computational resources and storage capacities, for certain tasks, e.g. those requiring low latency, it might be better to release their execution to other parts of the system, closer to the edge [27].

### 2.2 Fog Layer

According to the OpenFog consortium, the fog computing model moves computation from the cloud closer to the network edge, by placing geo-distributed computational resources between the cloud and sensor layer [21]. The Fog computing layer is composed of fog platforms (the fog nodes) which rely on highly virtualized resources running under hypervisors. Fog platforms are constituted by a large number of fog nodes consisting of routers, switches, Wireless Access Points (WAP), Road Side Units (RSUs), gateways, wireless set-top boxes, network bridges and cellular base stations [6, 11].

These fog nodes, which can be fixed or mobile, are distributed in different geographical locations to provide services in proximity of edge devices. Given that the edge devices (Sensor layer) can be mobile, the Fog layer should enable reallocation of tasks and resources at runtime. In fact, the high mobility characteristic of fog computing typically gives the impression that fog nodes enter in and out the network, which may give rise to novel availability issues. In terms of security, the existence of this intermediate level also increases the attack surface of the system considerably.

### 2.3 Sensor Layer

This layer is bottom layer in the hierarchical architecture which consists of devices, sensors, actuators in a physical environment; for instance, vehicles, smart cards, IoT devices, etc. Devices in this layer are geographically distributed, can be fixed or mobile, and require minimal computational resources, being typically very energy-constrained. Usually utilized as smart sensing devices, they sense data and gather information, and then send it to the upper layer for processing, storage and distribution [2].

## 3 CURRENT DEPENDABILITY APPROACHES FOR FOG COMPUTING

Dependability approaches for fog computing are mainly proposed to address dependability objectives, redundancy models and fault management solutions. Figure 2 present a summary of the approaches in our literature review.

### 3.1 Dependability Objectives

Dependability requirements for fog computing are not clearly defined, as fog computing is a very recent technology. Our review of existing literature shows that authors differ significantly from each other in terms of the types of faults and errors they address, the method applied and even the dependability requirements themselves. Our study shows that the most common objectives are improving availability, reliability and Quality of Service (QoS). The ways to improve these attributes are typically based on redundancy models which are explained in the following subsection. Our study also shows that scalability, i.e. the ability to provide service for a large number of devices in the Sensor layer, is a crucial aspect of Fog. This can be related to the dependability attribute of performability.

### 3.2 Redundancy Models

Proposed redundancy models has been applied at different levels of the systems architecture: the communication links, the computing nodes and the application software. For instance, regarding network connectivity, Cau et al. used 5G communication to satisfy network reliability [10]. Wiss & Forsstrom. consider higher network connectivity as availability by using SCTP protocol instead of TCP [29].

Other works also consider the possibility of node failure. Itani et al. proposed dynamic failure recovery to improve node availability [16]. Zhou et al. used message broadcasting to check node availability for offloading tasks in case of fog node or link failures [33]. Okafor et al. proposed using of Spin-Leaf topology in fog network to ensure availability [20].

There is an interesting family of solutions that rely on software reallocation in order to increase service reliability/availability. Saqib & Hamid. proposed a task off-loading solution to ensure reliable computation in fog computing and IoT network [26]. Aral & Brandic. focused on QoS of VMs in an edge network infrastructure [3] and Osanaiye et al. proposed a live VM migration framework to increase QoS by improving availability of VM fog nodes [22]. Rimal et al. focused on improving system performance to promote QoS [24].

But, although authors allegedly address all these requirements, quantitative goals which would help us to define system thresholds are seldom or partially reported. In the scheme proposed in [9], authors considered strategies to minimize bandwidth and storage usage in which they reported percentage values of the gap between optimal scheme and practical measurements, lower than 6.2% and 30% for bandwidth and storage usage respectively.

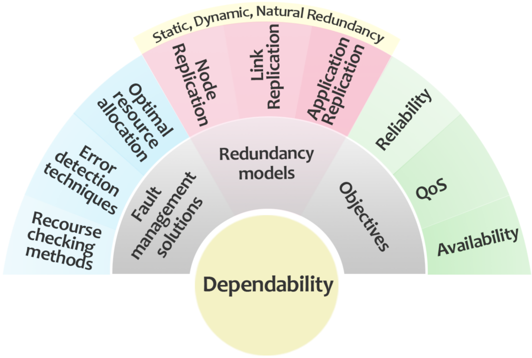
Static, Dynamic, Natural Redundancy
Node Replication
Link Replication
Application Replication
Optimal resource allocation
Error detection techniques
Recourse checking methods
Reliability
QoS
Availability
Redundancy models
Fault management solutions
Objectives
Dependability

**Figure 2: Summary of current dependability approaches for fog computing.**

Availability of replicated nodes or links are checked using different monitoring tools [17] or calculated via mathematical methods [9] or the use of machine learning algorithms [3].

With respect to the applied redundancy schemes, we found out that all types of redundancy have been used by different authors. It was observed that sometimes natural redundancy has been used for path redundancy, for instance as provided by wireless broadcast in [10]. The most common approach is Primary/Backup redundancy, with reconfiguration upon failure. Schemes relying both on Active replication (also known as Active/Active or Hot stand-by) and Passive replication (also known as Active/Passive or Cold stand-by) were found. For instance, Banson et al. proposed a dynamic path selection method based on Software defined networks (SDN), leveraging SDN monitoring tools to check the links availability status [5]. In another work [12] Maximum Distance Separable code (MDS) is used for dynamic clustering to find redundant nearby nodes. Although there are some approaches using dynamic redundancy, most often static allocation of redundancy is used. Cau et al. proposed static signal forwarding to available nodes in case of node failure in the network [10]. Other works also proposed passive replication in which virtual backup resources are pooled and shared across multiple virtual infrastructure [28, 31].

### 3.3 Fault Management Solutions

Solutions for fault management in fog computing proposed in the available publications on dependability and fog computing shows that authors have more importantly focused on these specific problems: a) Optimal allocation of redundancy, to reduce utilization. An instance is the work of Mennes et al. [18] which proposed an algorithm for optimal application placement; b) Techniques for error detection and reconfiguration upon failure, like e.g. Chervaykov et al. proposed a reconfigurable data storage system based on Redundant Residue Number System (RRNS) [13] and Xiao et al. [30] proposed a re-transmission method to re-send data in case of links or nodes failures or delay issues in a fog network; c) Methods for checking availability of redundancy, like monitoring tools, especially tailored for resilient networks [32].

## 4 DISCUSSION

Our study has provided us with very useful information about the current state of the art regarding dependability and fog computing. We have identified a number of research topics that seem to have received much attention from the research community. Namely: the trade-off between resource utilization and fault tolerance, the use of redundancy methods to increase availability and, last, the trade-off between reliability and timeliness, particularly for node replication schemes.

However, there is also an extensive list of challenges that have received very little attention. In the following, we summarize the open research problems that, in our opinion, deserve further investigation. The list does not intend to be exhaustive, but it defines a preliminary roadmap of the issues that need to be addressed next.

**Introducing more complex failure modes**. We noted that only simple (benign) failure modes have been considered in the literature. Authors typically consider crash and omission failures for communication links and available vs. non-available node failures (i.e. Stop failure semantics). However, more complicated failure modes like Byzantine or arbitrary failures, late performance and failures due to malicious faults remain unaddressed in this heterogeneous fog environment with complicated functionality. Another aspect that deserves more research is identifying system specification failures. For instance, late performance, bad design or wrong demand expectation/dimensioning might cause general failures as they have been disregarded while designing dynamic mechanisms such as dynamic reallocation of software. To give just an example, both intentional and unintentional Denial of Service (DoS) failures are possible in systems that do not properly handle oversized loads, even in cases where system allows dynamic changes. We believe that as the technology extends to more domains, the nature and severity of the faults that need to be addressed will have to be clarified.

**Integration of multiple levels of redundancy**. Since fog is a complex, multi-layered architecture, we need to consider failure probability in each layer of fog computing. So far, redundancy schemes have been proposed individually, and the potential interference between them has not been investigated. This also includes clarifying the interaction between application and data replication throughout the architecture, including data source and data transmission, which can be upward (from clouds to fog), downward (from sensors to fog) or internally cashed in fog node. All of this makes the fault-tolerant replication model more complicated, comparing to cloud and traditional critical systems.

**Security issues aggravated by faults**. We found out that there are a number of papers dealing with security [8, 13, 20, 22], but none of them addressed security for fog computing in the presence of unintentional faults. On the other hand, methods to achieving replication securely under differing threat models has not been specifically surveyed to provide secure redundancy techniques.

**Error propagation through the fog structure**. Uncontrolled error propagation is an important problem in any dependable system. The usual way to handle this problem is by defining and substantiating appropriate error-containment regions. This work has not been done for existing fog computing architectures. This aspect is related to the security problems discussed above, since correct error-containment is a good support for security, but it also concerns non-malicious faults, which can spread as subsystem errors and cause unexpected failures in other parts of the system. In a highly-dynamic system like the fog, poor handling of error propagation might even lead to instability system-level problems. This also opens an opportunity to investigate novel methods for error forecasting and dynamic error containment.

**Fault recovery and node reintegration.** Current approaches studied in this work have investigated different methods for fault detection, fault-tolerance, fault prevention and fault diagnostics. However, in a long-lifed system like the fog, it is also needed to develop methods that allow faulty components to recover and be reintegrated in the system operation. This can prevent system failure or shut down caused by fast redundancy attrition.

**Scalability concerns**. Fog nodes should be able to provide services for a large number of heterogeneous devices in different application areas. These application domains can require large-scale deployment of nodes, also for safety-critical domains. For instance, firefighting, transportation systems and industrial robotics. When a fog node fails in such large-scale critical systems, it is usually difficult to coordinate the huge number of sensors and devices in the presence of faults or to recover from failures. Similar unknown risks, caused by the large system size and the massive number of components, might be found in future applications.

**A comprehensive fault management framework** is missing. A fault management framework is a part of large network management structure. This framework can address faults in a higher level as well as designing a high level management infrastructure for addressing faults in a system [23]. Although there are some frameworks proposed for Fog computing and fault tolerance in Fog networks, they do not address faults in all aspects. For instance, considering connection failure, node failure, application placement, task management, etc. combined in the same framework package. As indicated above, certain notions like fault diagnosis and fault treatment have received little attention, as well as the threats posed by malicious faults. All these aspects should also be integrated in this fault management framework. Similarly to other large-scale networked systems, the fog allows application of novel methods based on statistical learning, such as machine learning, in order to identify anomalies and forecast faults, but this type of work is still in its infancy. A suitable framework should include methods for data collection and a repository of considered faults and mitigation techniques.

## 5 CONCLUSION

This paper has reviewed the current state of the art regarding dependability and fog computing. We have summarized the current research efforts and discussed a list of open research problems.

## ACKNOWLEDGMENT

This research has received funding from the European Union's Horizon 2020 Research and Innovation programme under the Marie Skłodowska-Curie grant agreement No 764785, and also from the VINNOVA project 2018-02437.

## REFERENCES

[1] M. Aazam and E. Huh. 2016. Fog Computing: The Cloud-IoT, IoE Middleware Paradigm. *IEEE Potentials* 35, 3 (May 2016), 40–44.

[2] Nasir Abbas, Yan Zhang, Amir Taherkordi, and Tor Skeie. 2018. Mobile edge computing: A survey. *IEEE Internet of Things Journal* 5, 1 (2018), 450–465.

[3] A. Aral and I. Brandic. 2017. Quality of Service Channelling for Latency Sensitive Edge Applications. In *2017 IEEE International Conference on Edge Computing (EDGE)*. 166–173.

[4] A. Avizienis, J. C. Laprie, B. Randell, and C. Landwehr. 2004. Basic concepts and taxonomy of dependable and secure computing. *IEEE Transactions on Dependable and Secure Computing* 1, 1 (Jan 2004), 11–33.

[5] K. E. Benson, G. Wang, N. Venkatasubramanian, and Y. Kim. 2018. Ride: A Resilient IoT Data Exchange Middleware Leveraging SDN and Edge Cloud Resources. In *2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI)*. 72–83.

[6] Kashif Bilal, Osman Khalid, Aiman Erbad, and Samee U. Khan. 2018. Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers. *Computer Networks* 130 (2018), 94 – 120.

[7] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog Computing and Its Role in the Internet of Things. In *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing (MCC '12)*. ACM, New York, NY, USA, 13–16.

[8] H. P. Breivold and K. Sandström. 2015. Internet of Things for Industrial Automation – Challenges and Technical Solutions. In *2015 IEEE International Conference on Data Science and Data Intensive Systems*. 532–539.

[9] J. A. Cabrera, D. E. Lucani., and F. H. P. Fitzek. 2016. On network coded distributed storage: How to repair in a fog of unreliable peers. In *2016 International Symposium on Wireless Communication Systems (ISWCS)*. 188–193.

[10] E. Cau, M. Corici, P. Bellavista, L. Foschini, G. Carella, A. Edmonds, and T. M. Bohnert. 2016. Efficient Exploitation of Mobile Edge Computing for Virtualized 5G in EPC Architectures. In *2016 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud)*. 100–109.

[11] X. Chen and L. Wang. 2017. Exploring Fog Computing-Based Adaptive Vehicular Data Scheduling Policies Through a Compositional Formal Method-PEPA. *IEEE Communications Letters* 21, 4 (April 2017), 745–748.

[12] X. Chen, X. Wen, L. Wang, and W. Jing. 2018. A Fault-Tolerant Data Acquisition Scheme with MDS and Dynamic Clustering in Energy Internet. In *2018 IEEE International Conference on Energy Internet (ICEI)*. 175–180.

[13] Nikolay Chervyakov, Mikhail Babenko, Andrei Tchernykh, Nikolay Kucherov, Vanessa Miranda-López, and Jorge M. Cortes-Mendoza. 2019. AR-RRNS: Configurable reliable distributed data storage systems for Internet of Things to ensure security. *Future Generation Computer Systems* 92 (2019), 1080 – 1092.

[14] A. V. Dastjerdi and R. Buyya. 2016. Fog Computing: Helping the Internet of Things Realize Its Potential. *Computer* 49, 8 (Aug 2016), 112–116.

[15] Pengfei Hu, Sahraoui Dhelim, Huansheng Ning, and Tie Qiu. 2017. Survey on fog computing: architecture, key technologies, applications and open issues. *Journal of Network and Computer Applications* 98 (2017), 27 – 42.

[16] May Itani, Sanaa Sharafeddine, and Islam ElKabani. 2018. Dynamic multiple node failure recovery in distributed storage systems. *Ad Hoc Networks* 72 (2018), 1 – 13.

[17] A. Jonathan, M. Uluyol, A. Chandra, and J. Weissman. 2017. Ensuring reliability in geo-distributed edge cloud. In *2017 Resilience Week (RWS)*. 127–132.

[18] R. Mennes, B. Spinnewyn, S. Latré, and J. F. Botero. 2016. GRECO: A Distributed Genetic Algorithm for Reliable Application Placement in Hybrid Clouds. In *2016 5th IEEE International Conference on Cloud Networking (Cloudnet)*. 14–20.

[19] M. Mukherjee, R. Matam, L. Shu, L. Maglaras, M. A. Ferrag, N. Choudhury, and V. Kumar. 2017. Security and Privacy in Fog Computing: Challenges. *IEEE Access* 5 (2017), 19293–19304.

[20] Kennedy Chinedu Okafor, Ifeyinwa E Achumba, Gloria A Chukwudebe, and Gordon C Ononiwu. 2017. Leveraging fog computing for scalable IoT datacenter using spine-leaf network topology. *Journal of Electrical and Computer Engineering* 2017 (2017).

[21] OpenFog Consortium Architecture Working Group. 2017. OpenFog Reference Architecture for Fog Computing. *OpenFog* February (2017), 1–162.

[22] O. Osanaiye, S. Chen, Z. Yan, R. Lu, K. R. Choo, and M. Dlodlo. 2017. From Cloud to Fog Computing: A Review and a Conceptual Live VM Migration Framework. *IEEE Access* 5 (2017), 8284–8300.

[23] Lilia Paradis and Qi Han. 2007. A Survey of Fault Management in Wireless Sensor Networks. *Journal of Network and Systems Management* 15, 2 (01 Jun 2007), 171–190.

[24] B. P. Rimal, D. Pham Van, and M. Maier. 2017. Mobile-Edge Computing Versus Centralized Cloud Computing Over a Converged FiWi Access Network. *IEEE Transactions on Network and Service Management* 14, 3 (Sep. 2017), 498–513.

[25] José Santos, Tim Wauters, Bruno Volckaert, and Filip De Turck. 2017. Fog computing: Enabling the management and orchestration of smart city applications in 5G networks. *Entropy* 20, 1 (2017), 4.

[26] M. T. Saqib and M. A. Hamid. 2016. FogR: A highly reliable and intelligent computation offloading on the Internet of Things. In *2016 IEEE Region 10 Conference (TENCON)*. 1039–1042.

[27] Subhadeep Sarkar and Sudip Misra. 2016. Theoretical modelling of fog computing: A green computing paradigm to support IoT applications. *Iet Networks* 5, 2 (2016), 23–29.

[28] W. Wang, H. Chen, and X. Chen. 2012. An Availability-Aware Virtual Machine Placement Approach for Dynamic Scaling of Cloud Applications. In *2012 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing*. 509–516.

[29] T. Wiss and S. Forsström. 2017. Feasibility and performance evaluation of SCTP for the industrial internet of things. In *IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society*. 6101–6106.

[30] Y. Xiao, Z. Ren, H. Zhang, C. Chen, and C. Shi. 2017. A novel task allocation for maximizing reliability considering fault-tolerant in VANET real time systems. In *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 1–7.

[31] Wai-Leong Yeow, Cédric Westphal, and Ulaş Kozat. 2010. Designing and embedding reliable virtual infrastructures. In *Proceedings of the second ACM SIGCOMM workshop on Virtualized infrastructure systems and architectures*. ACM, 33–40.

[32] Xiao Yuan, Chimay J. Anumba, and M. Kevin Parfitt. 2016. Cyber-physical systems for temporary structure monitoring. *Automation in Construction* 66 (2016), 1 – 14.

[33] B. Zhou, A. V. Dastjerdi, R. N. Calheiros, S. N. Srirama, and R. Buyya. 2017. mCloud: A Context-Aware Offloading Framework for Heterogeneous Mobile Cloud. *IEEE Transactions on Services Computing* 10, 5 (Sep. 2017), 797–810.