Towards the Decentralised Cloud: Survey on Approaches and Challenges for Mobile, Ad hoc, and Edge Computing

ANA JUAN FERRER, JOAN MANUEL MARQUÈS, and JOSEP JORBA, Universitat Oberta de Catalunya, Spain

Cloud computing emerged as a centralised paradigm that made "infinite" computing resources available on demand. Nevertheless, the ever-increasing computing capacities present on smart connected things and devices calls for the decentralisation of Cloud computing to avoid unnecessary latencies and fully exploit accessible computing capacities at the edges of the network. Whilst these decentralised Cloud models represent a significant breakthrough from a Cloud perspective, they are rooted in existing research areas such as Mobile Cloud Computing, Mobile Ad hoc Computing, and Edge computing. This article analyses the pre-existing works to determine their role in Decentralised Cloud and future computing development.

CCS Concepts: • **General and reference → Surveys and overviews**; • **Computer systems organization → Cloud computing**; *Embedded and cyber-physical systems*; • **Computing methodologies →** *Distributed computing methodologies*;

Additional Key Words and Phrases: Cloud computing, decentralised cloud, fog computing, edge computing, Mobile Cloud Computing, Mobile Ad hoc Cloud Computing

ACM Reference format:
Ana Juan Ferrer, Joan Manuel Marquès, and Josep Jorba. 2019. Towards the Decentralised Cloud: Survey on Approaches and Challenges for Mobile, Ad hoc, and Edge Computing. *ACM Comput. Surv.* 51, 6, Article 111 (January 2019), 36 pages.
https://doi.org/10.1145/3243929

1 INTRODUCTION

Computing is increasingly pervasive in all aspects of our life. Mobile phone capacities have expanded from a few Mbs of memory and MHz processors 15 years ago, to the stimulating mobile computing environments we currently have at our disposal. Present-day smartphones are equipped with touch screens, multiple sensors, diverse networking capacities, massive storage, and high-end multi-core processors comparable to a room-sized supercomputer from the 1980s [91]. Computing is not solely present on our smartphones but also in our cars, televisions, cameras, and not to mention refrigerators. This trend is not expected to be reversed in the short term. Along with advent of IoT smart fabrics, autonomous cars, and connected roads, diverse forms of

Ana Juan Ferrer is Also with Atos Research and Innovation.
Authors' addresses: A. J. Ferrer, J. M. Marquès, and J. Jorba, Universitat Oberta de Catalunya, 156 Rambla del Poblenou, Barcelona, 08018, Spain; emails: anajuanferrer@gmail.com, {jmarquesp, jjorbae}@uoc.edu.

0360-0300/2019/01-ART111 $15.00
https://doi.org/10.1145/3243929

nano-computing, smart cities, smart homes, and robots are expected to be a predominant part of our everyday life. As these smart connected things are increasingly enriched with computing power capabilities, there will be the need for these to go beyond rigid basic programming models to become fully networked computing systems capable of delivering advanced features and behaviours on the move and while interacting with their surroundings. Soon sensors, actuators, robots, drones, routers, servers, and cars will only be seen as particular forms of infrastructure elements in the so-called Decentralised Cloud. In the Decentralised Cloud, computing will no longer be constricted to specific devices but will be virtually embedded and pervasive to everything, enabling an unprecedented computing continuum [3].

Cloud computing initially emerged in the space in which "we transitioned from an era in which underlying computing resources were both scarce and expensive, to an era in which the same resources started to be cheap and abundant" [54]. Current approaches for Cloud computing are based on dedicated Data Centres managed by enterprises where resources are perceived as unlimited, in which everything is delivered as a service in stationary resources set-ups. Cloud computing has enabled the democratization of computing. It has provided the illusion of infinite computing and allowed for the radical acceleration of commoditization of computing by making the concept of utility computing a reality. Existing Cloud computing developments emerged as part of a centralized paradigm in which large and fully equipped Data Centre concentrate the available computing power. Gartner's Edge Manifesto [31] has demanded "the placement of content, compute and Data Centre resources on the edge of the network, closer to concentrations of users. This augmentation of the traditional centralized Data Centre model ensures a better user experience demanded by digital business."

Initial steps toward decentralisation of Cloud Computing are being realised through the emergence of Fog [13] and Edge computing [30]. These are recognised to be rooted in the Cloudlet concept in Mobile Computing [8, 83]. Edge and Fog computing are currently being developed under the premise of static computing devices (or sets of them), which serve as computing environments located in the vicinity of data generation areas to avoid latencies generated by application of Cloud computing to IoT scenarios. In this context, IoT devices are solely considered as mere sources of data presenting minimal actuation capacities.

Nevertheless, the expected gains in complexity of IoT devices becoming "Intelligent things" anticipate a future in which connected things go beyond existing basic data gathering and actuation and offer enhancements to execute deep learning and AI processing. Therefore, this evolution will bring about novel opportunities to future evolution of Edge computing by summing up ever increasing computing capacities available in "Intelligent Things" at the edge of the network.

"Intelligent things" are assemblies of a set of computing and storage resources with diverse actuators and sensors, conceptually similar, to Mobile Devices. Connections among Mobile Cloud Computing and evolution of Edge Computing do not end here: the fact that "Intelligent things" have a number of constraints in their size and energy harvesting is also shared with Mobile Cloud computing works.

Additionally, the fact that "Intelligent things" are capable of moving raises novel challenges with regard to resource reliability, unstable connectivity, and overall computing environment dynamicity, which for a number of years have been deeply analysed in the context of Mobile Cloud Computing. This reinforces the idea that future evolution of Edge computing has an intrinsic relationship with Mobile Cloud Computing.

Beyond existing Mobile Cloud Computing Cloudlet and Edge computing concepts relation, we claim that Mobile Cloud and Ad hoc Computing concepts create novel forms of distributed and opportunistic computing, which will become a key building block for the evolution of existing Cloud and Edge computing towards the Decentralised Cloud. As illustrated in Figure 1, evolution paths

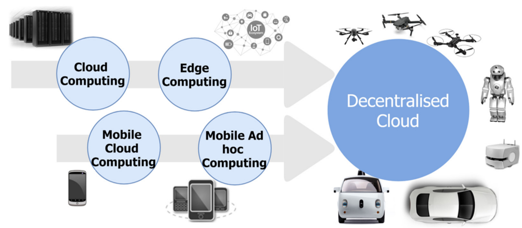

Fig. 1. Cloud, Edge, Mobile Cloud, and Ad hoc Cloud computing evolution paths.

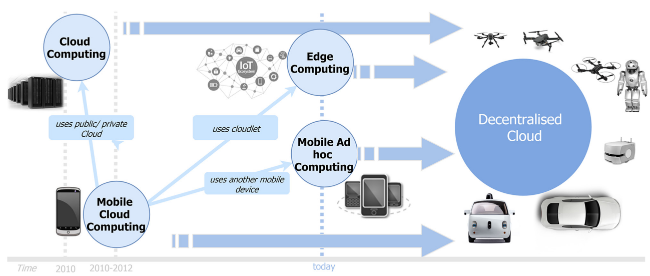

Fig. 2. Relations among Decentralised Cloud models.

of these technologies have so far occurred in parallel; however, we anticipate their convergence in the Decentralised Cloud Concept.

This work aims at making a novel contribution by providing a systematic literature review of works in the areas of Mobile Cloud Computing, Mobile Ad hoc Computing and Edge computing helping to identify their relations and existing developments as potential contributions to further evolution of Decentralised Cloud concept. In Section 2, we identify the diverse models of decentralised cloud we encounter in today's literature including the specific relations among them. Then, Sections 3, 4, and 5 elaborate on the details of these different approaches. Namely, Mobile Cloud, Mobile Ad hoc Cloud and Edge Computing. For each of these, we define existing challenges and approaches; and we analyse existing works according to the defined taxonomies. To conclude, we observe significant gaps still to be covered by research to make the decentralised Cloud vision a reality.

2 BACKGROUND

The movement towards Cloud decentralisation is a novel approach from a Cloud perspective. There is extensive research in the areas of Mobile Cloud Computing (MCC), Mobile Ad hoc Computing (MAC), and Edge computing, which can be explored to gain understanding of existing approaches and challenges this poses. Figure 2 provides a high-level view identifying relations among these technologies. Table 1 details conceptual differences and similarities in their current approaches. Cloud computing [65] initial developments revolved around on the Infrastructure as a Service, having AWS EC2 as its main representative. Nowadays, Cloud computing is considered both a business and delivery model that permits the acquisition of a wide range of IT capabilities encompassing from infrastructure, development environments, and security features to final user applications. Thus, making truly real the idea of an infinite all-purpose elastic IT utility in which everything is open to being consumed by anyone, from anywhere "as-a-Service." Cloud Computing supports elastic delivery of services, which in the case of major Cloud providers are delivered from centralised Data Centres distributed across diverse regions all around the world.

Table 1. Cloud, MCC, MAC, Edge Computing Concepts Comparison

	Cloud Computing	**Mobile Cloud Computing**	**Mobile Ad hoc Computing**	**Edge Computing**
Motivation	To provide IT services on-demand	To provide additional capacity to resource constrained mobile devices	To provide additional capacity to resource constrained mobile devices	To reduce latency in computation tasks of Data Generated by IoT
Client	Any application	Mobile applications	Mobile applications	IoT applications
Resource Nature	Steady servers in Data Centres	Mobile devices complemented with capacity on Cloud computing environments, Steady servers located in the vicinity (cloudlet) or other Mobile devices	Mobile devices complemented with capacity of other Mobile devices	IoT devices complemented with capacity of steady servers located in the vicinity of IoT data generation areas
Means to acquire additional capacity	Federation with other Clouds	Cloud, Cloudlet and other mobile devices	Other mobile devices	Cloud
Optimisation problem	Capacity, QoS	Energy, Capacity	Energy, Capacity	QoS (latency)
Representative Comm. Offerings	AWS, Azure, Google Cloud	None	None	AWS Greengrass, Azure IoT Edge, FogHorn
Standardisation	NIST, ETSI, SNIA, DMTF, OASIS, etc.	None	None	OpenFog Consortium, ETSI

While this approach has been proven to be powerful for a very large number of scenarios, Internet of Things (IoT) and massive number of connected Things bring novel challenges to its development addressed through Edge and Fog Computing.

The emergence of Internet of Everything—the networked connection of people, process, data and things—is expected to exponentially grow the number of connected devices worldwide, from billions of units available today, to orders of magnitude of tens of billions of units expected to be deployed in the coming years. At present, we are observing evolutionary forms of Cloud Computing, such as Edge and Fog, starting to break the Data Centre barriers to provide novel forms of computing embracing computing power and data resources increasingly obtainable everywhere. These are forcing existing Cloud computing environments, which emerged as part of a centralisation paradigm, to evolve to decentralised environments avoiding drawbacks of large data movements and latency, specifically found in IoT scenarios [19]. These new forms of Cloud are making the Cloud concept create a more distributed approach to lead to better performance and enabling a wider diversity of application and services, complementarity to traditional X-as-a-service cloud models, which is used as resource rich environment.

Major cloud providers such as AWS [84] and Azure [4] are increasingly featuring Edge Computing services, as a way to extend their offerings for IoT scenarios. In doing so, Edge computing has become an evolution of well-established Cloud offerings.

Both for Edge, Fog and Mobile Cloud Computing traditional Cloud models are perceived as the resource rich environment to be used to extend limited capacities of these environments.

In parallel to the hype around Cloud computing, mobile technologies experienced an unprecedented growth both in development and adoption. Mobile Devices and Cloud Computing have increasingly evolved in the concept of MCC. MCC is a research area that "aims at using cloud computing techniques on storage and processing of data mobile devices" [35]. In traditional approaches to MCC, Cloud computing environments are used to overcome Mobile devices limitations. To do so, three different approaches can be found in literature to augment limited mobile devices capabilities:

(1) approaches that boost mobile devices capabilities with resources from Cloud environments, by means of public or private environments;
(2) approaches that rely on servers located close to the mobile device position, called Cloudlets and
(3) approaches that are dependent on other mobile devices to increment their capacities, recently coined under the term MAC.

MCC Cloudlet concept [80] is a precedent to Edge and Fog computing. It defines the concept of a proximal cloud that brings closer computing capacities to avoid latency to the mobile devices its serves. Diverse authors have drawn on this connection. Examples of these are Bilal [8], Satyanarayanan [79], and Satyanarayanan [83].

The forms of MCC that consider other Mobile Devices to make use of their available resources recently have been classified as Mobile Ad hoc Cloud Computing [97]. The concept of MAC develops a common umbrella term for a number of works both in MCC and other research environments that consider mobile devices as valid execution resources [97]. Historically, MCC motivation has been the need to extend Mobile Devices limited resources to richer execution environments. Fuelled by the increased capabilities of Mobile Devices, this research area aims to go beyond these approaches considering the Mobile Device a valid Cloud resource and therefore capable of taking part in Computing infrastructures. Although the concept had already been addressed in previous MCC works, it presents the characteristic of the opportunistic behaviour of the environments very much of interest for the development of decentralised Cloud concept.

The call towards the decentralisation of Cloud computing is present in a wide variety of works and under diverse terms. Satyanarayanan [79] contextualises the current trend towards Cloud computing decentralisation in the context of alternating waves of centralisation and decentralisation, which have affected computing since the '60s. In these, centralisation of computing has been prevalent in '60s and '70s through batch processing and timesharing and in the 2000s employing traditional centralised Cloud computing models; whereas alternating with decentralisation in '80s and '90s via the emergence of personal computing and in which Edge computing presents the last episode of this on-going trend.

Shi [86], among many authors, has explained the need of decentralisation motivated by the development of richer IoT devices, which have changed their role from simple data consumers to rich data providers. Overall rich IoT devices are expected to generate such amounts of data that in the longer term it will become impractical to centralise all their processing.

Garcia-Lopez [30] further elaborates the factors that call for placement of computing at the edge with the help of five elements: Proximity, bringing facilities to distribute and communicate information; Intelligence, due to the fact that IoT devices increase computing capacities at a rapid pace; Trust and Control, by permitting data sources to remain in control of generated data and application management; and Humans, making them the centre of all interactions. In addition, Garcia-Lopez recognises further research challenges to be addressed in Cloud computing for realising novel highly distributed Edge architectures and middleware, which go beyond Hybrid Cloud developments and coping with specific challenges of decentralisation and "computation trade-offs between mobile terminals and cloud servers." These are expected to have to deal with issues affecting stability on the availability of edge devices, such as devices' churn, fault tolerance and elasticity aspects; all of them being core aspects of research in Mobile Cloud Computing in the last years.

A similar approach is taken by Varghese when analysing the future of Cloud computing in the next decades in Varghese [94]. It precisely identifies MCC Cloudlet and MAC concepts as foundations for the evolution of Cloud Computing infrastructure towards the decentralised computing infrastructure in which resources are away from the Data Centre boundaries.

At the time of writing, there is still not a term that delimits the above mentioned highly decentralised computing infrastructure. Some authors, such us [23], refer to this just as Edge Computing, declaring that existing Edge Computing development just reflect an embryonic evolution stage of what it can become by utilising the incorporation to the concept of “smart phones, sensor nodes, wearables, and on-board units where data analytics and knowledge generation are performed, which removes the necessity of a centralised system.”

Other authors prefer to define a specific term for this foreseen Edge capacity advancement. This is the case for Villari [95] who defines Osmotic Computing as “a new paradigm to support the efficient execution of IoT services and applications at the network edge.” Osmotic Computing considers again distributed across Edge and Cloud application execution elaborating on MCC concepts to define its evolution requirements while acknowledging the need of reverse “mobile (cloud) offloading” mechanisms, which move functionalities from Cloud computing to Edge devices. Bojkovic [11] has coined the term Tactile internet for the evolution of Fog (Edge) computing that combined with developments in SDN and NVF able address requirements for ultra low latency and high availability required in scenarios such as “autonomous vehicles, haptic healthcare, and remote robotics,” among others.

Back in 2014, Lee [56] invented the TerraSwarm concept, as a set of technologies able to integrate cyber and physical worlds in a way that “Mobile battery-powered personal devices with advanced capabilities will connect opportunistically to the Cloud and to nearby swarm devices, which will sense and actuate in the physical world.” These herald the beginning of a close link, which can be detected among MCC and MAC and the future of Cloud and Edge Computing. It is interesting to note that the consideration of Mobile device in TerraSwarm was also surpassing existing smartphone technology but also considering Autonomous vehicles and Unmanned aerial vehicles (UAVs). While these, still today, seem futuristic scenarios, analysis of UAVs as “near user edge devices which are flying” was provided in Loke [57]. This was anticipating the use of mobile cloud computing cloudlet servers in the air on drones as “Data mules,” able to bring data where it can be better processed, or by means of the development of “Fly-in, Fly-out infrastructure,” able to provide punctual computing services in a specific location.

However, today specific implementations of these are starting to emerge, showing their potential to develop in the medium term. Some of the most noteworthy examples are as follows: Jeong [44] provides a Cloudlet mounted in a UAV that provides offloading capabilities to a series of static mobile devices, and Valentino [92] develops an opportunistic computational offloading system among UAVs. All these works evidence that the nature of UAVs, and generally speaking robots and autonomous vehicles, share device characteristics with traditional mobile devices in the form that they present constraints in terms of computational and storage capacity, battery and energy supply limitations. Together with the fact of relying on unstable network links due to mobility, which drives to specific device reliability and volatility issues not yet explored in stationary resource environments present in Edge and Cloud computing today.

While specific needs of Smartphones have driven the development of MCC, we anticipate that the emergence of rich IoT devices in the form of “intelligent things” will push towards the development of Decentralised Cloud.

Whereas it is widely recognised that MCC Cloudlet concept is the precursor of Edge computing, further evolution of this concept will be rooted in other forms of Mobile Computing, which has relied on the interconnection of constrained devices to resource richer environments in traditional clouds, and more importantly, in the opportunistic formation of computing infrastructures among mobile devices and MAC.

This will be motivated by the on-going trend towards decentralisation but also by the increasing pressure to take advantage of all available computing capacity. As the evolution of Moore's law is progressively reaching its limits and computing demands will solely increase with the advent of more complex IoT devices and their expected data deluge generation. Parallel advances in Deep learning and artificial intelligence will intensify this need by multiplying the requirement for complex processing at the Edge.

Altogether it is evidence of the need for Cloud and Edge computing to draw inspiration from and explore in-depth evolutions that have happened in the context of MCC and MAC to address novel challenges that Decentralised Cloud is bringing to this context, removing the boundaries that have existed up to this point among these technologies employing resources that are analogous in nature.

A clarification should be made about the terms used in the rest of this article. At the time of writing, there is still much controversy regarding the use of Fog and Edge computing terms. OpenFog consortium [67] in its reference architecture [34] alludes to the fact that Fog Computing is often erroneously named Edge Computing and argues about the differences at levels of Cloud interaction, hierarchy, layers, and aspects addressed. In particular indicates that "Fog works with the cloud, whereas edge is defined by the exclusion of cloud. Fog is hierarchical, where edge tends to be limited to a small number of layers. In addition to computation, fog also addresses networking, storage, control, and acceleration." [34] Fog Computing is a term coined by CISCO in its enlightening paper "Fog Computing and Its Role in the Internet of Things" [13]. In this publication, Fog computing is defined as a "highly virtualized platform" between end-devices and Data Centre clouds, which provides compute, storage, and networking services (see Section 4 for details on definition). Recent publications of OpenFog Consortium blog [52] extends this definition, to consider Fog Computing "a continuum or a range of computing that goes from the cloud, to the edge, to the devices."

Currently, many authors are considering "Fog Computing" a vendor-specific term, and therefore opt for using "Edge Computing." ETSI has also coined the term "Mobile-edge Computing" [25], which explicitly focuses on the Network aspects of the technology. While the research and standardisation communities are currently debating the appropriate term to use, major cloud and technology providers have released related products, tagged as Edge Computing, to the market. These commercial products do not adjust to differentiation levels provided by OpenFog Consortium, instead, they consider Edge computing all computing environments outside Data Centre boundaries. The growing popularity of these products, evidenced by Google Trends "Fog Computing" and "Edge Computing" comparison of terms [33], makes us opt for using the Edge computing term throughout this article. However, as our work is a literature survey, both terms Fog and Edge Computing will be used as synonyms, making use of the term used by the referenced author in the different analysed studies.

3 MOBILE CLOUD COMPUTING (MCC)

MCC is an area of research meant to connect Mobile Computing [77, 78, 90], Cloud computing [65], and even certain aspects of networks management [75]. There are manifold approaches and definitions, yet in general they all have the same principle at their core, which is to apply to mobile devices' compute and storage process techniques from cloud computing [35]. Some examples of these definitions are provided below:

- Sanaei [74] defines MCC as "a rich mobile computing technology that leverages unified elastic resources of varied clouds and network technologies toward unrestricted functionality, storage, and mobility to serve a multitude of mobile devices anywhere, anytime through the

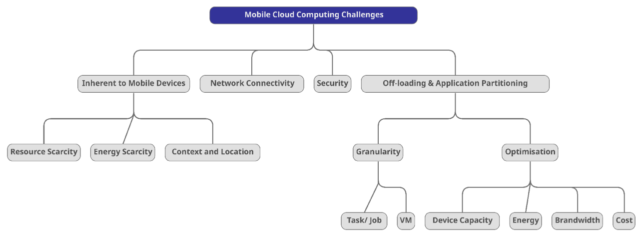

Fig. 3. Mobile cloud computing taxonomy.

channel of Ethernet or Internet regardless of heterogeneous environments and platforms based on the pay-as-you-use principle."

- For Chang [16], MCC represents "an emergent mobile cloud paradigm which leverage mobile computing, networking, and cloud computing to study mobile service models, develop mobile cloud infrastructures, platforms, and service applications for mobile clients. Its primary objective is to delivery location-aware mobile services with mobility to users based on scalable mobile cloud resources in networks, computers, storages, and mobile devices. Its goal is to deliver them with secure mobile cloud resources, service applications, and data using energy-efficient mobile cloud resources in a pay-as-you-use model."
- Kovachev [51] describes MCC as "a model for transparent elastic augmentation of mobile device capabilities via ubiquitous wireless access to cloud storage and computing resources, with context-aware dynamic adjusting of offloading in respect to change in operating conditions, while preserving available sensing and interactivity capabilities of mobile devices."

MCC has been recognised as a beneficial technology for diverse fields of mobile applications in Hoang [40]. By means of concrete application examples, it details mobile applications that take advantage of MCC in areas that comprise: Mobile commerce, using MCC as the mechanism that allows handling mobility in operations such as "mobile transactions and payments and mobile messaging and mobile ticketing" [40]; Mobile learning, applying MCC to overcome shortcomings in terms of devices costs, available network, computing and storage resources, as well as, access to limited educational resources; Mobile healthcare, in which MCC is employed as a tool that permits efficient access to information making specific emphasis in the necessary security and data protection aspects; Mobile gaming, enabling these kind of applications to access resource richer environments. In addition to these, MCC is considered admittedly useful for content sharing, searching services and collaborative applications [40].

3.1 MCC Challenges

Challenges in the scope of Mobile Cloud fall into four groups: First, we can mention the ones inherent to the use of mobile devices. These are related to the limitations mobile devices in resources and battery and ability to perceive context and location. In addition to these, challenges related to the different approaches favoured to deal with these constraints such as Network Connectivity, Security and Off-loading & Application Partitioning are detailed. These are represented in Figure 3 taxonomy.

Inherent Mobile Devices Challenges

Resource and Energy scarcity: While initial works in the area of Mobile Computing considered overcoming devices' limitations as the major issue for performance associated to resources hardware characteristics [15, 78]. Authors today acknowledge the substantial augmentation of devices

capacities in terms of CPU, memory, storage and others, such as the size of screen or associated sensors [70]. Nevertheless, battery lifetime is still often perceived as a main roadblock due to the effect it has on mobile resource availability. With this regard, Sanaei [74] acknowledges existing efforts to optimise, by means of applying offloading technics, energy utilisation on the mobile device and the fact that this cannot always reduce energy. Other authors do not regard energy management and battery restriction as an issue for present-day Mobile Cloud Computing [40, 102]. Specifically, Hoang [40] presents MCC as a promising solution that can help to reduce power consumption in mobile devices without having to perform changes into the devices structure or hardware and taking advantage of software off-loading techniques.
Context and Location: Context and Location: Guan [35] underlines the fact that mobile devices allow the assessment of certain information from the device itself without the user's interaction. Two types of contexts are identified: spatial context, related to location, position and proximity; as well as, social context, context extracted from user's or groups interactions. Reference [74] describes obstacles that radiate from the management of this context owing to the exponential growth of context and social dynamism related to context storage, management and processing on resource constrained mobile devices.

Network Connectivity. The nomadic nature of Mobile devices and the fact that they rely on wireless networks is a challenge for Mobile Cloud [70]. Wireless networks are "characterized by low-bandwidth, intermittent and lower reliable network protocols is considered and as a factor that affects latency and therefore, unfavourably affects energy consumption and response time" [74]. Hoang [40] adds to this list availability issues and heterogeneity among different wireless networks interfaces applied. Hoang [40] explicitly cites as sources for availability issues, the aspects of traffic congestion, network failure and signal loss. In terms of heterogeneity, Hoang [40] considers diversity on the radio access technologies, precisely determining the MCC needs with regards to continuous connectivity, on-demand scalability and energy efficiency. To address all these issues, approaches based in local clouds or cloudlets have been developed. These are examined in detail in Fernando [26], Gkatzikis [32], Sanaei [76], and Satyanarayanan [80]. In this context, Reference [101] tackles the aspect of wireless intermittent connectivity among mobile devices and cloudlet environments, as a MCC key distinctive aspect. It develops a dynamic offloading algorithm that regards user's mobility patterns and connectivity to diverse geographically disperse cloudlets. In addition, it examines cloudlet's admission control policies based on user's distance to the cloudlet and cloudlet's coverage areas.

Security. Fernando [28] concedes the fact that although many authors cite the need to provide the appropriate security context for Mobile Cloud Computing Services execution, the issue has been barely touched upon thus far. Specific analysis of Authentication and Privacy and Security issues are exhibited in Alizadeh [2] and Mollah [62].

Shiraz [88] underlines that fact that "privacy measures are required to ensure execution of mobile applications in isolated and trustworthy environments while security procedures are necessary to protect against threads, mainly at network level." The analysis of privacy and security issues featured in Gao [29] does not specifically concentrate on Mobile Cloud Computing issues but rather reports well-known issues in the context of Cloud computing, which involve the providers access to to user's virtual infrastructure or mobile physical threads associated with lending, lost or thieve of mobile devices or connection to public open network infrastructures.

Conversely, Khan [47] provides a careful analysis and draws a detailed comparison of existing Mobile Cloud Computing security frameworks. Conclusions point to the fact that the majority of security frameworks overlook the trade-off between energy consumption and security requirements. It identifies hurdles that can be surmounted at the level of "data security, network

Fig. 4. Classification of Mobile Cloud Computing models.

security, data locality, data integrity, web application security, data segregation, data access, authentication, authorization, data confidentiality, data breach issues, and various other factors" [47].

Off-loading & Application Partitioning. Many of Mobile Cloud Computing perspectives today revolve around application offloading and partitioning techniques to augment mobile device capacities [28]. Off-loading consists of moving part of the mobile computational workload to more resource-rich servers in heterogeneous Cloud models [53].

Research in Off-loading techniques [55] often contemplates a set of well-delimited phases, which include:

- Decision to offload: Offloading has been viewed as a means to save energy and /or improve performance of mobile devices; however both feasibility and acquired benefits depend on factors such as available network link and amount of data to be transmitted. Considering the trade-off between offloading costs (commonly in terms of time, performance and energy) versus local processing costs, plays a key role in reaching offloading decisions.
- Decision of application parts to off-load: The offloading granularity can be taken statically; this is pre-determined in the mobile application execution flow at application development time; or dynamically, determined at runtime based on the execution context at a given time [53, 55]. The granularity of parts of the application candidates to be offloaded ranges from offloading the complete application, so called coarse-grained methods; to fine-grained methods, which consider specific application parts at levels of object, method, class, function, and even tasks [24].
- Selection/Definition of infrastructures to off-load: Both specific framework analysis and literature surveys in the offloading topic consider this step, not as the selection of a computing infrastructure, but a specific server or surrogate selection in a pre-defined infrastructure [24, 53, 55]. Very frequently, application partitioning mechanisms include mechanisms to optimize mobile device at the levels of processor augmentation, energy savings, execution cost, and bandwidth utilization [28, 29, 70, 88].

3.2 MCC Models

Multiple papers tackle the issue of workload offloading from mobile devices to resource richer environments. These can be classified according to four main perspectives, depicted in Figure 4:

- Off-loading to Server: By means of offloading to specific servers, which can be located or not in a Cloud environment, it provides resources to alleviate mobile resource constraints;
- Off-loading to Cloud: through use of private or public Cloud computing infrastructures; this considers the execution of off-loaded application parts often to virtual machine executing in a IaaS provider [1].
- Off-loading to Cloudlet: By means of using of Local computing infrastructures or Cloudlets [80]. These aim to reduce the overhead network latency derived from the use of distant

traditional cloud infrastructures by using local infrastructures, cloudlets, closer to the mobile device location. Satyanarayanan [82] further develops this concept by regarding the cloudlet as an intermediary step between the mobile device and the cloud, in a three-tier hierarchy in which the cloudlet is deemed to be a "Data Centre in a box" set-up to "bring the cloud closer" to the device [81], therefore reducing latency. Conceptually, the idea of Cloudlet is the building block that sustains Fog Computing. This is further developed in Section 5.

- Off-loading to mobile device/Ad hoc Cloud: By using additional mobile devices capacity, commonly labelled as surrogates. Recently, under this standpoint a novel concept has been formulated through the development of Ad hoc mobile cloud concept.

3.3 Analysis of Existing Works in MCC

Drawing on the previously identified Mobile Cloud Computing Challenges and Models, we define the taxonomy that is depicted in Figure 3. In the sections that follow, we employ the Mobile Cloud Computing models for categorisation of existing works.

3.3.1 Approaches Based on Server Off-Loading. The model considers off-loading from Mobile device to a fixed external server, which can be or not hosted in a cloud environment.

"MAUI: Making Smartphones Last Longer with Code Offload." MAUI [21] targets reducing energy consumed by mobile devices while executing resource intensive applications. It offers fine-grained application off-loading at level of method. MAUI was defined by Microsoft research in 2010, being one of Mobile Cloud Computing precursor works, role in which is commonly referenced [28, 40]. MAUI's design goal is to overcome battery limitations of mobile devices. This work identifies the three most energy voracious categories of applications: video games and streaming, as well as, applications that focus on analysing data streams coming from mobile device's sensors. By means of .Net code portability features, MAUI maintains two versions of the application to offload; one executing at the mobile device (equipped with Windows mobile in an ARM architecture) and one running at the server (x86 CPU). MAUI architecture presents components that execute both on the mobile device and the server. The MAUI programming model, based on C# and Microsoft .NET Common Language Runtime (CLR), allows developers to annotate methods as remotable. These annotated methods are instrumented at compilation time with the aim of allowing application state transfer when offloading. To minimize the amount of transfer of serialized application state, it uses an incremental approach, solely engaged in transmitting differences between mobile and remote states in different method invocations. At runtime the MAUI determines for all instrumented methods whether to execute it on the mobile device or remotely in the server before each execution.

Experimentation over MAUI's performance has been performed using four distinct applications, three of them pre-build and are currently running on Windows mobile phones (face-recognition, interactive video game and chess game applications), whereas a forth application was developed from scratch, a voice-based Spanish-to-English translator. For the first three, analysis of energy consumption and performance has been performed by comparing standalone execution of the application of the mobile application versus remote server execution based on a set of application metrics defined per each one of the mobile applications and considering several network conditions.

"Cuckoo: a Computation Offloading Framework for Smartphones." Cuckoo framework [46] targets application offloading for Android platform. Cuckoo design goals focus on providing a framework for mobile phones computation offload, which allows energy consumption reduction, along with increased speed on execution of compute intensive applications for the Android mobile platform. The framework includes a programming model based on Java, conjoined with a Runtime environment. It allows mobile-to-server fine-grained method offloading, which presents two optimization models: minimizing computation time and mobile device energy consumption. Server side execution requires any environment running a complete Java Virtual Machine, whether it is a dedicated server, a local cluster, a VM in a Cloud environment or any other capable environment.

To a certain extent, Cuckoo can be considered an analogous work in Java and Android platforms to previous .Net and Windows Mobile developments in MAUI. Having its main difference in the fact that Cuckoo permits to distinguish among code versions to be executed in the mobile device and the server [46] bring new capabilities for user system configurability. This mechanism while being powerful in some cases has been identified as a drawback in previous mobile cloud computing surveys due to the need of providing two versions of the same application code [28]. By the use of Ibis High performance computing system Cuckoo acquires new capabilities for remote server configurability compared to other Server Off-loading existing works. Cuckoo permits dynamic deployment and interoperability with remote servers in diverse execution environments. This way, Cuckoo is able to consider Off-loading to Server model, in remote servers in diverse execution environments (dedicated server, a local cluster, a VM in a Cloud environment, etc.) in a transparent manner enabled by the interoperability layer that Ibis facilitates.

Cuckoo has been validated using two example applications: First, eyeDentify, an application that performs image pattern recognition, and simultaneously computing and memory intensive. eyeDentify was re-factored to use the Cuckoo programming environment. The second application was Photoshoot, which is a distributed augmented reality mobile application.

3.3.2 Approaches Based on Public/Private Cloud Computing. These approaches focus primarily on augmenting mobile device capabilities enabled by the use of more powerful resources in traditional date centre Clouds, both considering in private or public cloud environments and different levels of the Cloud stack (IaaS, PaaS, and SaaS).

"CloneCloud: Elastic Execution Between Mobile Device and Cloud." CloneCloud presents a system that aspires to "augment mobile devices" [18] capabilities by means of offloading methods to device clones executed in a computational cloud. Vision was presented in Chun [18], while its implementation is reported in Chun [17]. CloneCloud design goal is enable automatic transformation of mobile applications to profit from Cloud.

Significantly different from previous works Cuckoo and MAUI, CloneCloud is not dependent on the programmer to create application partitions. Instead, its purpose is to make application partitioning seamless and automatic for the programmer. To do so, it applies an offline method in which both static program analysis and dynamic program profiling are performed to define application partitions. Application partitions, in this case, are a choice of execution points where the application migrates a part of its execution and state from the device to the clone. Analyses can be executed considering several execution characteristics (considering CPU, network, and energy consumption) leading to the creation of diverse partitions for the same application.

Static program analysis aims to identify "legal" choices whereby migration and re-integration execution between the device and the cloud are made possible. The system defines these migration points as method entry and exit points. "Legal" partitions are pre-computed and stored in a database. These are used in combination with dynamic application profiler to manage the distributed execution of the application across the mobile device and the device clone in the cloud.

CloneCloud is reliant on the concept of Application layer VMs, specifically in the Java VM available on Android devices, DalvikVM. This supports the migration of application pieces between the mobile device and the clone despite the differences in the CPU instruction set architectures, ARM and x86. Migration in CloneCloud is at level of thread and relies in a private Cloud environment based on VMware ESX.

"ThinkAir." ThinkAir's [49, 50] ambition is to simplify developers tasks in migrating their applications to Cloud. To do so, it presents a framework that facilitates method-level computation offloading to Cloud environments. The main novelties provided by ThinkAir adopt a more sophisticated use of Cloud computing environment directed at exploiting Cloud potential with regard to elasticity and scalability for Mobile Cloud benefit. ThinkAir provides on-demand cloud resource allocation to comply with specific requirements of mobile applications to offload at level of CPU and memory resources. Unlike CloneCloud, ThinkAir makes use of public commercial Cloud offerings and does not store pre-defined off loadable code partitions. ThinkAir relies instead on annotations provided by the developer to identify parts of code candidate to be off-loaded. Furthermore, it enables parallelization by dynamically managing virtual infrastructure in the Cloud environment, therefore reducing both cloud server's side and overall application's execution time and energy consumption. The primary server in the ThinkAir architecture is a VM that clones of the mobile device replicating both data and applications (additional information about how these clones are synchronized and kept up-to-date is not present in the analyzed works). This primary server is always set-up ready to be contacted by the mobile device. Other VMs distinct from the primary server, called secondary servers, are instantiated on-demand by the user. The primary server manages communications from the mobile, the life-cycle of these secondary servers, as well as task, allocation in case of parallelization; however, no concrete details about this mechanism are readily available.

3.3.3 Approaches-Based Cloudlets. Satyanarayanan [80] formulated the concept of cloudlet as "a trusted, resource rich computer or cluster of computers that is well-connected to the internet and it is available for nearby mobile devices." In this concept, the mobile device acts as thin-client to services deployed in the cloudlet by means of VMs and that are accessible by wireless LAN.

As opposed to previously described approaches subject to distant servers or clouds, the overall aim of these models is to decrease the overhead network latency derived from the use of distant traditional cloud infrastructures. This is achieved by using local clouds or infrastructures, cloudlets, closer to the mobile device location. Proximity intends to ensure the predictability of the cloudlet's response time in order of magnitude of milliseconds. Generally speaking, the cloudlet vision constructs scenarios where cloudlets shape "decentralised and widely disperse" computing infrastructures spread over the Internet. It is similar to enriching WIFI access points today with an easily deployable, long-lasting and self-managing "datacenter-in-a-box" resource.

"The Case for VM-Based Cloudlets in Mobile Computing." While Satyanarayanan [80] defined the concept of cloudlet, it also provided an architecture to turn the concept into reality. Several authors, including the recently developed Edge Computing area, do not dramatically differ in the concept articulation but in its realization. Some of these works are described in the sections that follow.

Overall design ambition of this work is to unveil potential of mobile computing as a mechanism that "seamlessly augments cognitive abilities of users using compute-intensive capabilities such as speech recognition, natural language processing, computer vision and graphics, machine learning, augmented reality, planning and decision-making." This ambition, articulated more than a decade ago, is today demonstrated ahead of its time and visionary by dint of existing Edge computing and Decentralised Cloud foreseen evolution. The architecture proposed in this article is contingent upon "transient customization of cloudlet infrastructure" [80], in which VMs are temporarily created, used, and, afterwards, discarded from the cloudlet infrastructure in a dynamic manner and to provide a specific service to a mobile device located nearby. VM technology creates the necessary isolation and compatibility for cloudlet sustainability.

"Gabriel." Following the example described in previous work [80], Gabriel [36] applies the Cloudlet concept to wearable devices to exploit its potential in Cognitive assistance processes.

Gabriel relies on Cloudlets, with a view to reduce end-to-end latency while addressing battery and processing constraints of these wearable devices. The concept is developed for the cognitive assistance scenarios employed include applications such as Face, Object, and Optical character recognition and Motion classifier. These require the interaction with wearable device (Google Glasses, in this case) while placing high demands on both computation level capacity and latency requirements. The system design considers offloading from wearable devices to cloudlets and considering transiency among diverse cloudlets. Also it takes the account that in cloudlets could have interactions to public/private clouds. Another notable aspect is that Cloudlets could be implemented with resource richer (not smartphones) movable devices such as laptops and netbooks. These bring different options for deployment of Gabriel framework itself, however are not developed in its architecture so to enable interoperability and seamless integration with a variety of execution environments. In Gabriel, offloading normally occurs between the wearable device and the cloudlet located nearby. The wearable device discovers and associates to it. In the absence of a cloudlet set-up, a proposed solution is to offload to cloud, but this approach penalizes because of WAN latency and bandwidth issues initially avoided with cloudlets. In the case the internet connection is not accessible, an alternative solution is the use of a mobile device or a laptop carried by the user as a direct device to offload. The vision proposed is that as smartphones increasingly come with more processing power, they can morph into viable offloading devices in the near future. Gabriel deploys each cognitive application in a separated VM in the cloudlet cluster. This cluster is also utilized to perform computational task parallelization required by the various applications.

3.3.4 Approaches Based on Mobile Devices Cloud Computing Infrastructures. Hitherto, approaches regarding the mobile device part of the cloud are the least explored ones. The works under this classification significantly differ from previous MCC presented works. For Server-, Public/Private Cloud-, and Cloudlet-based MCC approaches, the mobile resource acts as a thin client and main motivation is to extend its limited capacities by acquiring additional capacity in resource richer environments. These resource richer environments are witnessed as infinite, in terms of the resources they can bring to the mobile application execution, neither presenting limitations in terms of battery and network instability under these approaches consideration. Here, the perspective changes. First, due to the consideration of mobile devices, which changes perception from just been seen as a thing client, to be considered a valid execution environment to complement capacity of other resources in its network. But also from the view that the resource in which workload is offloaded presents the same volatility and instability characteristics as the resource that has originated the workload. The notable evolution that commenced a decade ago thanks to Moore's Law, has led to the increase of power and functionality of mobile phones and in general any electronic device. Specialists expect this trend to continue up to a certain limit, as previously presented. Mobile battery is also an extended area of research both at industry and academia driven by requirements generated by the developments of wearable technologies. Initial works driving this approach were presented in 2009–'10 and coined under the MCC term. Recently (2018) authors have used the term Mobile Ad hoc Cloud computing to refer to similar approaches. These are presented below in the following sections.

"Hyrax: cloud computing on mobile devices using Map Reduce." Hyrax [58] is "a platform derived from Hadoop that supports cloud computing on Android devices." Hyrax is constructed on the basis of a vision in which mobile computing is "an extension of cloud computing for which foundational hardware is at least partially made up of mobile devices" [58]. Hyrax's overall goal is to evaluate feasibility of mobile devices' hardware and network infrastructure to become a sort of cloud provider that uses local data and computational resources, analogous to traditional clouds. The envisaged type of clouds would be made of the opportunistic creation of networked connections of smartphones in which smartphones perform individual local computations in support of a larger system-wide objective, which aggregates smartphone's local computations to meet goals of an overall application. The following principles guide the proposed mobile cloud computing infrastructure: "(a) each node is owned by different user; (b) each node is likely to be mobile; (c) each node is battery powered, and (d) network topology is more dynamic" [58]. The following are understood as advantages of the approach: avoidance of large data transfers to centralized remote services to perform computational jobs, instead of using local or vicinity capacity processing mobile multimedia and sensor data immediately; enablement of more efficient access and sharing of data stored on smartphone devices through local area or peer-to-peer networks; as well as distributed hardware ownership and maintenance.

Hyrax has based its work on porting Apache Hadoop to be executed in the proposed Mobile Cloud infrastructure, rather than traditional commodity hardware as it is by definition intended. It is important to note that although mobile nodes are intended to be distributed, implementation of Hyrax utilizes an approach based on centralised management. Additionally, Hyrax to some extend oversimplifies the problem by relying solely on existing Hadoop fault tolerance mechanisms to overcome issues derived from use of mobile resources of the infrastructure. In addition, Hyrax does not take into account any of the application offloading and partitioning techniques for mobile application in previous works, instead it focuses on providing a already existing data analytics infrastructure in which worker nodes are mobile devices that offered functionality is equivalent to traditional clouds. Thereby, Hyrax is significantly divergent to previous MCC works however, completely in line with on-going and expected developments of Edge and Decentralised Cloud approaches, which almost a decade after still ambition similar goals.

"A virtual cloud computing provider for mobile devices." Huerta-Canepa work on "virtual cloud computing provider for mobile devices" is described in Huerta-Canepa [42]. Its overall ambition is to overcome mobile resource limitations by simulating a cloud environment with other mobile resources available in the vicinity for situations in which connection to cloud is inaccessible or too costly. This work is unique in MCC field by defining an infrastructure that is solely created out of mobile devices as an ad hoc p2p cloud. The work provides remarkable inputs in relation to context management adapted to particularities of mobile devices. Specifically in this work, partitioning of an application takes into account local resource availability and application resource needs. The selection of subrogates to which to offload and assign application partitions uses the amount and type of resources requested by the application execution and the amount of these resources available at candidate surrogates. This takes into account the mobile devices context defined as: social context, including relationships among users; location; and number devices in the vicinity. In addition, the works put forward a model for application partitioning that considers energy and time constraints; a failure prevention mechanism based on context; plus an adaptable trust mechanism that enables to open the platform to unknown nodes. Huerta-Canepa [43] depicts the set of policies and processes involved in the proposed Context-aware offloading policy schema. The schema details the following steps: Monitoring, Partitioning, Selection of surrogate candidate, and Offloading. Implementation of this architecture is reported to be based on Hadoop running on top of PhoneME. PhoneME is Sun Microsystems project to provide a JVM and Java ME reference implementation.

3.4 Features Comparison

Table 2 provides a feature comparison using the concepts defined in Mobile Cloud Computing Taxonomy, adding information about implementation status, maturity, and use cases. In previous subsections, we have analysed existing MCC works according to the MCC defined models for offloading: to server, cloud, cloudlet, and mobile device. Independent of this system, architectural

Table 2. MCC Frameworks Feature Comparison

Works	MAUI [21]	Cuckoo [46]	Clone Cloud [18]	Think Air [49]	Cloudlets [80]	Gabriel [36]	Hyrax [58]	Virtual Cloud [42]
Resource Scarcity	X	X	X	X			X	X
Energy Scarcity	X	X	X	X				
Context and Location			X	X		X		X
Network	X		X	X	X	X	X	
Security								
Off. granul.	Method	JVM	JVM	Method	VM	VM	JVM	
Off. Optim.	Energy	Energy	Energy	Cost		Latency		Energy, Time
MCC Model	Off. to Server	Off. to Server	Off. to Cloud	Off. to Cloud	Off. to Cloudlet	Off. to Cloudlet	Off. to Mobile Device	Off. to Mobile Device
Prog. Model/ Language	Windows Mobile /.Net	Android / Java	Android / Java	Javascript / Java, C#		Android / Java	Java / Hadoop	PhoneMe, Java / Hadoop
Maturity	Prototype	Prototype	Prototype	Prototype	Architecture	Prototype	Prototype	Prototype
Use cases	Image Recog, Gamme	Image conv., Aug. Reality	Virus scan, image search, adver.	Image proc., aug. reality and video		Object recog., OCR	Video search and sharing	

approach of all analysed studies, except of Hyrax, build on top of two main concepts: overall aim to optimise mobile device constrained resources and subsequent need for workload off-loading. From the analysed works, only Gabriel (by use of wearable devices) is exploiting the MCC optimisation models and techniques for other available constrained devices than mobile devices, while these have huge potential for development in IoT and Decentralised Cloud context. Reinfurt [72] provides a systematic classification of IoT devices in form of patterns. In this, it is recognised that many IoT devices are mobile and are located off the power grid and recognises the need for these to optimise energy use, similarly to mobile devices addressed by MCC works. We claim that similarly to how MCC Cloudlet concept has recently been conceptually used in the development of Edge Computing concept. Tools and techniques for task off-loading and energy optimisation developed in the context of MCC will soon have to be employed in IoT and decentralised cloud context, together with the need of optimising IoT devices resources and taking advantage of all existing computing capabilities at the Edge.

According to this analysis, we observe that the criteria most often used for optimisation of offloading decision are Energy and Execution time. The consideration of the Energy criteria is devoted to MCC traditional overall approach to preserve mobile devices resources. We foresee this need will remain with the application of MCC techniques to IoT context. It is noteworthy that so far consideration of security in MCC has been only marginally addressed. This is particularly critical while considering more advanced scenarios for MCC in Decentralised Cloud context, as mobile devices and, generally speaking, IoT devices, act as sources of data that will soon become critical to protect.

4 MOBILE AD HOC CLOUD COMPUTING (MAC)

The concept of MAC has been only recently coined in Yaqoob [97], in which it is recognised as a novel area of research that is still in its infancy. In this work, MAC is understood as a new research domain that aims to "augment various mobile devices in terms of computing intensive tasks execution by leveraging heterogeneous resources of available devices in the local vicinity."

A more concise definition is provided in Yaqoob [98]: "MAC enables the use of a multitude of proximate resource-rich mobile devices to provide computational services in the vicinity." Balasubramanian [6] further extends MAC definition by adding cooperation factors among participant mobile devices "A MAC is a pool of devices with high computational capabilities and is closer to the user. This low-cost computational environment is deployed over a network where all nodes cooperatively maintain the network." To the best of our knowledge, there is not yet a formal definition of MAC.

MAC motivation is to address situations in MCC for which connectivity to cloud environment is not feasible, such as absence or intermittent network connection [97]. This motivation was already the driver for MCC "offloading to mobile device" works, specifically central to Huerta-Canepa [42, 43]. It has to be noted that neither motivation nor MAC definitions denote substantial differences with previous MCC works instead; MAC appears as a novel term to denominate more recent works.

MAC is recognised to have its roots into MCC but also in opportunistic computing [97]. The definition of opportunistic computing [20] provides additional considerations relevant for a system solely constituted by mobile devices. These are the concepts related to resource volatility and churn that can support further formal definition of MAC: "Opportunistic computing can be described as distributed computing with the caveats of intermittent connectivity and delay tolerance. Indeed, mobile and pervasive computing paradigms are also considered natural evolutions of traditional distributed computing. However, in mobile and pervasive computing systems, the disconnection or sleep device situations are treated as aberrations, while in opportunistic computing, opportunistic connectivity leads to accessing essential resources and information" [20].

Kirby [48] develops the desired features for ad hoc clouds: "An ad hoc cloud should be self-managing in terms of resilience, performance and balancing potentially conflicting policy goals. For resilience it should maintain service availability in the presence of membership churn and failure. For performance it should be self-optimizing, taking account of quality of service requirements. It should be acceptable to machine owners, by minimizing intrusiveness and supporting appropriate security and trust mechanisms" [48].

Shila [87] provides a distinction among mobile and static ad hoc clouds. The latter, are including Edge computing and cloudlet environments and elaborating links among these novel cloud models and volunteer computing, as a way to optimise use of spare devices in mobile and other edge devices. Similar consideration is made by Varghese [94], considering this as a major trend for changing cloud infrastructures.

4.1 MAC Challenges

Challenges in MAC are inherit from MCC. However, the consideration of Mobile devices as the single source of resources brings specific challenges to be considered in the context of MAC. These are depicted in Figure 5.

QoS and Fault tolerance: As described in Shiraz [88] mobile devices present specific characteristics with regards to resource availability (connectivity instability, battery limitation, communication bandwidth, or location variations). This makes it specifically relevant in the context of MAC the consideration of service management issues related to fault tolerance, availability and performance aspects. Shiraz [88] highlights the importance of Fault tolerance mechanism considering the nature of mobile devices and its volatility. In addition, it remarks the need of incorporating additional aspects for QoS management in Mobile Cloud Computing, which entail frequent loss of connectivity and low bandwidth and computational resources. Management of volatility of mobile resources and the availability issues derived from this fact is as a result the main

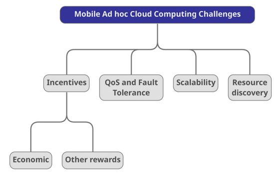

Fig. 5. Mobile Ad hoc Cloud Computing taxonomy.

identified challenge. Related work in the area for Mobile Cloud Computing based on Cloudlets recognizes as main problems limited and highly demand resources and mobility of users. Yaqoob [97] reinforces the need of additional research of stability issues related to ad hoc and distributed clouds. Similarly to some aspects of Service Management, few authors so far have analysed the problem of Admission control (the mechanisms to decide whether to accept or not a service to be executed on a cloud infrastructure). It is likely due to the fact that it is solely applicable in the context of MAC. In addition to this, expected autonomic nature of MAC calls for management procedures that are self-managed. This autonomic management has to consider self-healing mechanisms to optimize provided QoS taking into account levels of fault tolerance and device's churn.
Scalability: Mobile Ad hoc Clouds could potentially sustain the provision of services over a massive number of resources with limited availability. Specifically on this aspect, some authors such as Hoang [40] only contemplate network QoS factors relevant for Mobile Cloud Computing, relying on local clouds and cloudlets as the simple solution for these issues. Particularly, Hoang [40] identifies challenges in this area such as the distribution of processing, networking and storage capacity, in addition to the trade-offs management among cost and quality of experience. Both aspects, when extrapolated from MCC to MAC context, become critical to further develop this technology at scale.
Incentives for participation: Incentives for participation represent a key aspect for MAC and generally speaking to any volunteer computing system. Previous research in the area of volunteer Computing has demonstrated that temporal and voluntary resource donation is linked to different types of social, cultural and economic incentives with respect to service and data exchange, financial and collaboration aspirations. User's willingness to contribute is a key aspect for any contributory system. Although this area has been barely analysed, Nov [66] presents motivations to contribute in the eScience area where most of the Volunteer computing work has been developed. Findings relate motivations mainly to be "do good" and social contribution.
Resource Heterogeneity: Generally speaking, Ad hoc Clouds are particularly susceptible to the heterogeneity of devices. As resources set-in up the Ad hoc Cloud environment are not confined to the data centre boundaries, but instead are extracted from sets of available resources, management frameworks have to consider device heterogeneity as key enabler.
Resource Discovery: Very linked to mobile devices churn in MAC, processes for resource discovery in MAC have specific requirements with regard to the need to manage the environment dynamicity as well as to act in close relation with incentives mechanisms.

4.2 MAC Models

The analysis of existing literature in MAC, as well as general ad hoc Cloud and opportunistic computing enables the definition of the following potential models for MAC (depicted in Figure 6).

- Distributed: Similarly to existing works in Contributory or Voluntary Computing, MAC could be based on the temporary resource donation, which is voluntarily contributed to set-up the ad hoc mobile cloud. In this case, mobile resources would act at the same time

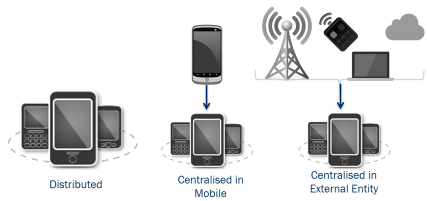

Fig. 6. Classification of Mobile Ad hoc Cloud Computing models.

Table 3. MAC Frameworks Feature Comparison

Works	**Job Sharing** [26]	**MOCCA** [61]	**Ad hoc Cloud** [100]	**MobiCloud** [37]	**mClouds** [60]	**Aura** [38]
QoS and Fault Tolerance						
Scalability						
Incentives	Economic				Economic	
Res. Heterogeneity						
Res Discovery			X		X	X
MAC model (Centralised)	in Mobile	in External Entity	in Mobile	in External Entity	in Mobile	in External Entity
Prog. Env.	Java	Java				Android / Java
Maturity	Prototype	Prototype	Simulation	Simulation	Model	Prototype
Use Cases	Distributed Mandelbrot Set Generation	Cholesky Decomposition Fast Fourier Transform				MapReduce Word count

as resource contributors and as resource users, by executing tasks or jobs in the MAC environment. Rooted in the contributory approach, mobile devices capacity is expected to be bestowed for an undetermined time period and can be disconnected at any time; as well as, it is decentralised and purely distributed, we can note the absence of any dedicated resource to its management.

- Centralised in Mobile: These represent models in which one of the mobile devices taking part in the MAC does act as a Master for the Ad hoc cluster, having the rest of devices as "surrogates." This model is inherited directly from previous work in MCC in which the concept of surrogate was described.
- Centralised in External entity: This model categorizes these cases in which the mobile device is deemed not to have sufficient resources to perform the ad hoc cloud management, and other resource richer entities are selected as master. This model therefore only views mobile devices as "workers" or "surrogates." In the observed cases, the master election is a static decision, and not considering operational environments.

The taxonomy in Figure 5 for MAC challenges defines previously described characteristics for MAC. This taxonomy is used in Table 3 to classify existing works presented in next section.

4.3 Analysis of Existing Works in MAC

This section presents a detailed analysis of previous works in MAC.
"Dynamic Mobile Cloud Computing: Ad Hoc and Opportunistic Job Sharing." Fernando [26, 27] elaborate on various aspects of dynamic mobile cloud computing framework. This framework aims to exploit the cloud when it is defined as "a cloud if local resources utilized to achieve

a common goal in a distributed manner." The aim of this work is to explore the feasibility of such local cloud to support mobility in mobile computing and associated concerns such as sparseness and hazardousness of the resources, in addition to limited energy source and connectivity. This framework aspires to respond to the following characteristics, being: "(a) dynamic, in the way it can handle different resources and connectivity changes; (b) proactive, so that costs can be pre-estimated; (c) opportunistic, it makes use of resources as they are encountered; (d) cost-effective, in a manner that allows task distribution based on a cost model benefiting all participant resources; (e) not limited to mobile devices, but able to manage low end devices such as sensors" [27]. As opposed to previous works analysed it considers parallel task execution using simultaneously diverse surrogate devices, however details on the approach to do so, are not provided.

The system architecture is organised in a cluster, in which one of the end-user devices acts as master, with a set of associated surrogate mobile devices performing slave tasks. Although authors intention in this set of works is to handle diverse end-user devices in the IoT spectrum, experimentation performed focus on PCs and mobile devices.

"MOCCA: A mobile Cellular Cloud Architecture." MoCCA [61] is described as a "cellular Cloud architecture for building mobile clouds using small-footprint microservers running on cell phones." MoCCA's objective is to avoid costs incurred in the set-up of traditional cloud data centres by taking advantage of already existing infrastructure elements. MoCCA advances the idea of benefiting from already existing telecommunications and networking elements in GSM cellular systems to build its architecture. Thus, the resources included in the architecture are smartphones, base stations, base stations controllers and mobile switching centres. Five aspects are identified as main concerns for Mobile Cloud design in this work: (1) Connectivity, bandwidth limitation, lack of direct connectivity among mobile devices, and the need to consider frequent network disconnections; (2) Computational limitation, due to mobile device resource limitations; (3) Churn, due to users mobility and devices' volatility; (4) Energy, with the approach of conserving energy in the mobile device; (5) and Incentives to users to participate with their mobile device in the Mobile Cloud infrastructure. The architecture proposed consists of two main parts: MoCCA Client and MoCCA manager. The latter, provides centralized control from the base station controller resource and executes from the base station. The MoCCA client is powered with an execution sandbox with stores function codes to be executed, in addition to Client controller and Audit and logging functions. MoCCA has been evaluated with computer bound applications. The only notable issue regarding Mobile Cloud Design that has been evaluated is Energy consumption from data reception and transmission. The remaining concerns (connectivity, churn, computational limitations, and incentives) have yet to be considered in their architectural design and evaluation.

MoCCA's differentiation aspect from previous MCC and MAC works is that MoCCA adopts GSM cellular network infrastructure as part of the MAC. This fixed infrastructure acts as the MAC coordinator. The idea of using network equipment as part of the computing infrastructure at the Edge is now intensively examined as part of Edge computing research.

"Ad hoc Cloud as a Service." Zaghdoudi [99, 100] present a "protocol and a preliminary architecture for the deployment of Ad hoc MCC on top of MANET Ad hoc networks." They address the need of solving dependence of mobile devices with remote cloud by exploiting capacities of surrounding devices. In these, two main entities are considered: Providers, offering nodes acting as resource providers; and Customers, which request resources. The resultant protocol, C-Protocol "governs the interaction and the communication among Ad hoc nodes and provides the dynamic management of providers and customers" [99, 100]. The proposed architecture presents two layers: The C-protocol layer, a meta-layer intended to provide required network services; the CloudSim layer: a simulation layer using CloudSim simulation aiming to model and simulate a data centre environment and virtualized infrastructure based on mobile devices. The protocol considers adding and members departure processes, as well as Customer inclusion. No specific details about potential implementation of these, such as mechanisms for customer or provider registry or fault tolerance, monitoring mechanisms, workload considerations are constituent of this work.

The originality of this work lies in the joint consideration of network and compute aspects (although the latter are not developed with full details) and specifically the joint consideration of MAC and spontaneous networks such as MANETs. Initial experimentation has used nine laptops equipped with Windows and Linux operating systems simulating mobile nodes connected over WIFI Adapters. The objective of the experimentation was to analyse the feasibility of three metrics: time to set-up, time for customer to join, and time to add a provider in the MAC system.

"MobiCloud." MobiCloud [37] is presented as a "reliable collaborative mobilecloud management system," which enables the efficient and collaborative use of available mobile phone resources. This work coins the novel term mobilecloud to refer to the overall objective of exploitation of computing capacities of mobile and field devices even when no internet connectivity is available. The detailed architecture comprises two types of nodes: a field control node, named Cloud Agent and participant nodes (mobile or field nodes). The Cloud Agent is the agent requesting to form a Cloud and provides centralized Cloud controller functionalities. When an application is submitted to the CloudAgent, it localizes from the set of available registered resources those that match the defined application requirements.

The reliability of the resources is assessed by means of a Trust management system that takes into account QoS offered by the participant nodes.

Available nodes are prioritised resting on, first, number of available CPUs, and then, time employed in data transmission. The differentiation aspect of this work is declared to rely on the node reliability mechanism and its reputation system based on user's feedback. Other works in the past [42, 43] have provided fully automated processes built upon collection of historical node behaviour. Evaluation of MobiCloud has been performed using an extension of CloudSim simulation and has included the homogeneous computing capacities of nodes, complete availability of all nodes and uniform distribution of connectivity speed. The metric evaluated in simulation has been application execution time.

"mClouds." mClouds [60] build on the vision future mobile devices will become core components of mobile cloud computing architectures and not just thin clients to cloud environments. It particularly elaborates in the assumption that "computation and memory will likely increase considerably while battery and network capacity will not grow at the same pace" with the overall aim of reducing saturation of cellular data networks [60]. The initial analysis of mClouds architecture is divided into two main aspects: distributed mCloud processing and specific resource discovery procedures; and incentives management.

Distributed mCloud processing architecture comprises mDevs, mobile devices able to execute mTasks. An mTask is a part of a larger computing task that can be parallelised. Distributed mCloud processing advocates on a simple initial principle, execute locally whenever possible. For cases in which this is not feasible due to lack of resources in the task originator device (master), look for mobile resources to form a mCloud.

This work presents the interesting novelty of elaborating in incentives strategies for mCloud participation. Incentives mechanisms consider the mobile carrier as clearing house, to reduce network congestions at certain locations. mClouds is conceived as a commentary approach to previous MAC and MCC works developing tools and mechanisms for application partitioning and offloading.

"Aura." Aura [38] aims at providing IoT-based Cloud computing models in which mobile devices, acting as clients, are able to offload computation tasks to nearby IoT devices. Therefore, creating ad hoc cloud out of low-power IoT devices in a specific location to which proximal mobile devices can outsource computation tasks. Motivation for this approach is twofold: first, to provide a local computation environment that reduces latency and keeps data privacy; and second, with the intention of avoiding the costs of deploying data centre clouds located near to the client. The use of Aura is exemplified in a Smart building scenario. Compared to previous works, Aura brings the innovation of already considering IoT devices in the Smart Building scenario as part of the MAC system considering them not only as data sources but as valid MAC resources, depending on their specific characteristics. A proof of concept of the approach has been developed for Aura with an Android mobile application for Mobile Agent implementation; Controller as a Desktop Java application; and IoT devices capabilities represented by MapReduce ported to Contiki IoT platform. A number of IoT devices were simulated with Cooja framework. The experimentation was conducted by offloading wordcount implemented in MapReduce for optimisation of execution time.

4.4 Features Comparison

Table 3 provides a feature comparison using the concepts outlined in Mobile Ad hoc Cloud Computing Taxonomy, introducing additional information about implementation status, maturity, and use cases. At model level, we observe that so far the preferred model in existing works is to provide ad hoc mobile cloud functionality from an external entity. This external entity in the analyses works is offered from Cloud environments, IoT devices and even, Network equipment. Centralised management in a mobile that manages ad hoc clouds in other mobiles acting as "surrogates" is also a model that is gaining popularity emerging together with the increment of computing capacities of mobile devices. In both cases, there is a single point of failure for these architectures due to centralised design. Complete decentralisation and distribution has been an area of study in Volunteer and P2P systems in the past. This model of management is feasible and performant, as demonstrated in previous volunteer and p2p computing works, and provides interesting features at levels of mechanism for handling complexity of volatile resources, high scalability and self-management foreseen as specifically of interest for the evolution of mobile and steady ad hoc clouds.

Until now, only some specific MAC works have gone beyond the smartphone as main source of resources. Tools such as Aura describe initial steps towards the inclusion of IoT in mobile ad hoc architectures. In our view, future evolution of MAC in Decentralised Cloud will not only reinforce existing initial works addressing IoT devices but to focus its evolution on them. The exceptional forecasted development on the number and complexity of IoT connected devices will force this evolution as a mandatory requirement. The overall computing available at the Edge of the network is growing in number of devices but also in their capacity, coming from diverse and heterogeneous sources in form of robots, drones, and autonomous vehicles. At the level of challenges addressed, we observe consideration of location is yet to be addressed, as well as, QoS and massive scalability necessary in this context. As observed in Table 3, the attention to hardware heterogeneity in the management of MAC is not yet a reality in any of the analysed MAC works. This is, in our view, another clear source of evolution in the coming years for MAC and Decentralised Cloud in general. Over the past decades, Moore's law has enabled the substantial computing capacity growth in processors. Recently, we are witnessing the emergence of built-in artificial intelligence processing units into mobile devices, which are expected to soon power many other IoT devices. The foreseen slow-down progress expected for Moore's Law in the future will call for taking better advantage of all available compute resources, therefore forcing MAC systems and Decentralised Cloud to manage heterogeneity so to exploit all available compute sources.

5 EDGE COMPUTING

Cloud computing today has transformed into a massive centralized infrastructure acting as a central keystone for compute power, storage, process, integration, and decision making in numerous environments. Following the pattern we have thus far in the existing IoT set-ups, generated sensor data would have to be transmitted over the network to be centralized, processed, and analysed in the Cloud.

With a view to cope with IoT proliferation this scenario has to change, providing an infrastructure that takes into account billions of devices connected at the edge and offering more rapid processing and decision making. Therefore, the idea under Edge Computing is to enable the decentralisation of the cloud, approximating computation and storage to the sources, at the edge of the network: avoiding unessential network transmission and getting data and computation at the right place and right time.

Edge computing (also known as Fog Computing) paradigm [13] "extends Cloud Computing to the Edge of the network." Both Edge and Cloud manage computation, network and storage resources applying similar techniques such as virtualisation and multi-tenancy [12]. However, Edge computing's main aim is to address the latency issues detected in the application of Cloud Computing to large IoT scenarios [96].

Edge computing is defined by Shi [86]: "Edge computing refers to the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services." This work frames Edge "as any computing and network resources along the path between data sources and cloud data centres" [86].

The term Fog Computing has been instead proposed by Cisco [19]: "Fog Computing is a paradigm that extends Cloud computing and services to the edge of the network. Similar to Cloud, Fog provides data, compute, storage, and application services to end-users. The distinguishing Fog characteristics are its proximity to end-users, its dense geographical distribution, and its support for mobility." Also from CISCO, Bonomi's in its introductory work "Fog Computing and its role on the internet of Things" [13] proposes the following definition for Fog computing: "Fog Computing is a highly virtualised platform that provides compute, storage, and networking services between end devices and traditional Cloud Computing Data Centres, typically, but not exclusively located at the edge of network."

The definition provided by Vaquero [93] does not confine technology choices to virtualisation and adds a cooperation factor: "Fog computing is a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralised devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third-parties. These tasks can be for supporting basic network functions or new services and applications that run in a sandboxed environment. Users leasing part of their devices to host these services get incentives for doing so."

Overall, Bonomi's approach refers to the fact that IoT platforms will, in the short term generate large volumes of data, which will stand in need of analytics platforms to be geo-distributed; in a way of "moving the processing to the data." Therefore, creating the need for "distributed intelligent platform at the Edge Computing that manages distributed compute, networking and storage resources."

Edge and Fog Computing are not devised as competitors to Cloud; quite the contrary, it is conceived as the perfect ally for use cases and applications for which traditional Cloud Computing is not sufficient. Further extended in Bonomi [12] the Edge vision was created to "address applications and services that do not fit well the paradigm of the Cloud." Edge approach is very much aligned with Mobile Cloud Computing works, as recognized in Garcia Lopez [30], Satyanarayanan [82], and Yannuzzi [96]. When observing evolution of the market, again, the major Cloud provider, Amazon Web Services (AWS) appears as a pioneer in the area of Edge computing by its AWS Greengrass product [84]. This has recently being followed by MS Azure Edge platform [4], as will be presented in Section 5.3.2.

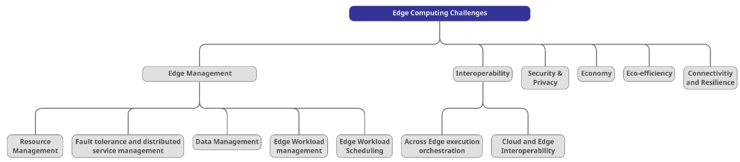

Fig. 7. Edge computing taxonomy.

5.1 Edge Computing Challenges

Below, we elaborate on a series of Edge computing challenges and characteristics necessary to be developed to make the described concepts a reality. These are also represented in Figure 7.

Edge management

Resource Management: Management of massive number of small diverse devices and sensors in Edge computing set-ups will necessitate new management styles, potentially decentralised and able to scale to degrees that nowadays are unprecedented in existing architectures [93].

Fault tolerance and distributed service management: Resource heterogeneity, scalability, fault tolerance, availability and performance are service management aspects still to be addressed in Edge computing. These are of specific interest due to the nature of devices and their volatility in addition to this need of including supplementary aspects for QoS management, scalability and heterogeneity in resources, integration of special devices, including hardware accelerators, FPGAs, and GPUs.

Edge Workload management: Encapsulation of edge workloads on top of Edge systems will have to accommodate diverse workload typologies and the different processors types where these workloads can be computed, for which the final encapsulation solution may vary. A system able to deal with various encapsulation approaches will be required to prepare the workloads depending on the final execution environment. Mechanisms adapted to balance between high-performance processor and low-power processor according to the final objectives of the workload should shortly be taken into consideration.

Edge Workload Scheduling: Workload or task scheduling in Edge and Fog computing has to take into account specificities of the Edge devices, such as energy constraints and QoS (usually in terms of latency optimisation). Diverse works have recently analysed the problem from diverse perspectives. Some works handle it as a joint optimisation problem among the Edge and Cloud resources: with the aim of addressing different application classes [10]; focusing on performance and cost optimisation [68]; and aiming to optimise delay and power consumption [22]. Others, such as Bitam [9], devise it with the innovative approach of bio-inspired optimization.

Data management: Hitherto, data intensive applications have been the key motivation spreading Edge computing need. Novel systems able to manage data scattered on an Edge heterogeneous and distributed environment needs to deal with the intricacies of the underlying complex infrastructure composed by smart devices, sensors, as well as traditional computing nodes. Conversely, developers must focus on establishing the relevant data, which is the necessary to keep, their format and quality, and how to process them, avoiding details concerning how to gather data, where to store or process them [69].

Edge Interoperability

Across Edge execution orchestration: Edge set-ups are envisaged to be spread covering wide geographic areas. For serving applications and services that make use of these distributed set-ups, mechanisms for deployment, provisioning, placement and scaling service instances across execution zones in the distributed Edge set-ups are necessary.

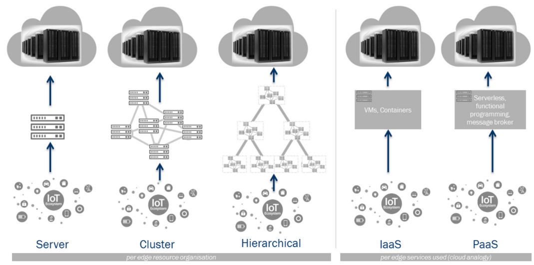

Fig. 8. Classification of edge computing models.

Cloud and Edge Interoperability: Current status of Edge computing developments very much relies on specific vendor solutions. For these to interoperate among them and with traditional clouds, new standards would have to appear to manage the expected scale of edge set-ups and the interoperability of devices and sensors.

Economy. Cloud computing has been recognized as a bridge between distributed systems and economics. Cloud computing providers offer a number of services to users using pricing schemes relying on incurred resource consumption. Existing commercial Edge computing environments, although based on simple devices, are being deployed in complex economic models that combine pay-per-use and are licensed-based. Further investigation is vital for designing models ready to cope with challenges and diversities of existing Edge Cloud models.

Eco-efficiency. A significant challenge associated with Edge deployments is potent power provisioning for locally deployed infrastructure. While substantial advances have been made for data centre and Cloud Energy Efficiency, particular challenges remain to optimise energy consumption and availability of energy sources in edge environments. Another environmental concern linked with Edge computing is the lifecycle of all devices that are disseminated. Approaches for device management of objects that incorporate a battery and matter potentially harmful to the environment would have to be considered in the future.

Security and privacy. Edge computing, similar to traditional cloud, is viewed as multi-tenant, and therefore actual set-ups will require of concrete isolation mechanisms to avoid security and privacy concerns.

Connectivity and Resilience. Resiliency is also a core characteristic required for Edge computing set-ups, notably for mission critical IoT applications. There is the overall need for these applications to continue providing their services from the Edge even when network links to Cloud are down or seriously overloaded. Diverse techniques are being studied to provide lack of connectivity resilience capability, among them fault tolerance systems across diverse Edge installations in a close location and techniques for unconnected Edge limited operation.

5.2 Edge Computing Models

Existing approaches to Edge Computing can be classified according to the following criteria. These different Edge models are depicted in Figure 8.

- Edge Server approaches are those that consider the Edge environment a device, which we name server, that provides computing and storage capacities to a series of Edge sensors and other resource poorer devices that are connected to it in a locally close environment. These so-called "servers" can be represented by devices that range from Raspberry Pis to servers, but so can devices such as connected cars, network equipment, or other rich smart

IoT devices, as long as they provide a minimum computing and storage capacity. In this regard, project HEADS has provided the following classification [39] among devices that comprise Tiny, Small, and Large. These can be described as: Tiny—very limited devices (8- and 16-bit micro controllers with less than 64kB program memory and 4kB of data memory). An example of this type of device is Arduino UNO; Small—devices with a specific OS and restricted hardware characteristics (less than 128kB program memory and less than 64kB data memory); Large—devices supporting general purpose OS. Examples of these are: Raspberry PI and Android. Edge Server approaches are the ones we encounter today in commercial products such as Amazon Greengrass, and Azure IoT Edge using the so-called "Large" devices. Also from equipment vendors such as Dell, we found pure and traditional servers to be deployed (Dell PowerEdge Series).
- Edge Cluster approaches are those considering sets of the previously so-called server devices that are coordinated by a node considered the cluster master. This clustered approach could be considered at diverse granularity levels in view of the nature of the proposed scenario and the compute/storage requirements. An exemplification of the concept could be performed in a smart home scenario considering that all "smart" enough devices, servers, aggregate their capacity to provide compute/storage capacities to other more resource constrained home appliances.
- Hierarchical classification considers layered configurations of Edge clusters. The layered approach could be construed according to diverse criterion. These include: layered approaches based on increasingly resources capabilities or location (aggregating at diverse levels, i.e., resources at home, neighbourhood, and smart city).

Making an analogy with existing Cloud offerings, we could also classify Edge approaches as:

- IaaS: Those offering Compute and Storage capacities in diverse virtualisation formats including VMs and containers.
- PaaS: offering access to programming environments (the more advanced ones providing Serverless and functional programming environments such as AWS Lambda), ML tool-sets as well as software capabilities such as message brokers to facilitate development of applications on top of these environments.

5.3 Analysis of Existing Works in Edge Computing

5.3.1 Existing Works in Research Environment. **"Fog computing: A platform for internet of things and analytics."** Fog computing was introduced in Bonomi [13]. Bonomi [12] enhances this initial work to propose a Fog architecture including new requirements that IoT scenarios pose on Fog Computing with regard to big data analytics. Overall the approach is based on the fact that IoT platforms will, in the short term generate large volumes of data, requiring of analytics platforms to be geo-distributed; in a way that "moving the processing to the data." Thus, creating the need for "distributed intelligent platform at the Edge (Fog Computing) that manages distributed compute, networking and storage resources."

The proposed high-level architecture has the following three key objectives: transparency, heterogeneity (of both resources and applications) and distributed orchestration. Transparency refers to the ability to manage in an abstract manner resource elements at edge, cloud and network. Heterogeneity is related to the diversity of previously mentioned resources but also to need of supporting multiple applications from diverse sectors. Finally, orchestration has to be driven by defined policies that consider scalability at local and global levels. Bonomi's work coined the term Fog computing. Although cloudlet concept is not specifically referenced in this work diverse authors have recognised its direct links in spite of different motivation for decentralisation: IoT infrastructure scalability, for fog computing; versus mobile applications performance for cloudlet [79].

"ANGELS for distributed analytics in IoT." ANGELS stands for "Available Network Gateways in Edge Locations for Sensors" and it is presented in Mukherjee [64]. ANGELS presents on-going work and explores the idea of using smart edge devices (sensor gateways, personal laptops, playstations, and smartphones) as envisaged in the Fog paradigm to perform parallel execution of data processing jobs in IoT, using idle capability of these devices. Overall ambition of this work is to take advantage of unused computing capacity at the edge of the network at homes and around these, to cope with demands for data analytics computation expected from the development of IoT systems. This architecture targets the class of applications that presents a data parallelization approach: namely, applications capable of processing data divisible into several subsets, partitions, which can be processed in parallel, similar to the MapReduce approach.

So far, this architecture is working under the assumption that edge devices are available. The next steps detail the consideration of dynamic availability patterns of edge devices. A new element of ANGELS is the contributory/volunteer computing element it brings, by means of taking advantage of idle of smart edge devices. However, it recognises that due to Edge devices resources constraints and their mobility, edge devices will have to be complemented with fully powered resource richer servers.

"Mobile Fog." Mobile Fog [41] presents a "high-level programming model," or a PaaS, "for applications that are geographically distributed, large scale, and sensitive to latency" [41]. The authors position this work as an alternative for Cloud PaaS that focus on web applications, by developing a solution that specifically addresses needs of data analytics for IoT. The objectives of Mobile Fog Programming model are: to ease application development on highly distributed heterogeneous devices; and to support scalability both at Edge and Cloud. In this work, Edge devices resources considered go beyond typical mobile phones, as we also consider connected vehicles. In Mobile Fog, an application is a group of distributed processes that have to be assigned into a set of disperse computing instances in edge devices and fog or cloud environments. It is considered a physical hierarchy of devices in which a process in an edge device is a leaf, and processes in the edge cloud are intermediate nodes and processes in cloud are considered the root. In this set-up each Mobile Fog Node manages workload from a specific geo-spatial location. Scalability management is performed through scaling policies that determine behaviour reliant on monitoring metrics such as CPU or bandwidth. Scalability mechanisms address instances at the same network level. Further work it is expected in runtime systems implementation and process placement algorithms. This work recognises to be complementary to fog architecture presented by Bonomi [12, 13] by addressing on programmability aspects in Fog context.

"Nebula." Nebula [14, 45, 73] is presented as a "dispersed edge cloud infrastructure that explores the use of voluntary resources for both computation and data storage." Nebula motivations are to reduce data upload to traditional clouds by offering disperse computing environments and to eliminate overhead of virtual infrastructure instantiation in Clouds. Nebula relies on volunteer computing mechanisms as tools that allow widely distributed environment. While supporting distributed data intensive applications, Nebula deems data movement and origination problems, considering geographical distributed execution. To do so, scheduling of computing has to take into account execution time but also data movement costs. Nebula system architecture includes the use of dedicated servers for central platform-level operations, together with a set of donated nodes both providing computation or data storage resources.

Data Nodes donate storage space to store application files. They provide operations to get and store data. Compute nodes, offer computation resources to the environment. With a view to maintaining isolation among the donated resources and applications executed by means of Nebulas, it employs NaCI sandbox provided by Google Chrome browser. By means of this sandbox, Nebulas orchestrates the execution of NaCI executables into the contributed resources. Evaluation has been provided for Nebulas MapReduce Scheduler comparing it to current Volunteer computing models BOINC and MapReduce-tuned BOINC. This evaluation has employed an experimental set-up using 52 Nodes in PlanetLab using a Word Count MapReduce Like application. Similarly to ANGELS [64], NEBULAS develops the idea of volunteer contribution of Edge resources, however, elaborating by-design management of fault-tolerance to edge devices churn and volatility.

"Resource Provisioning for IoT Services in the Fog." Skarlat [89]'s main objective is to provide both theoretical and practical foundations for resource provisioning in Fog environments. It provides a systematic classification of Edge resources. This classification comprises the following classes for resources: fog cells, single IoT devices that control a series of other IoT resources while providing virtualised resources; and fog colonies, described as micro-data centres built-up from a series of fog cells. In the proposed architecture: The Cloud-Fog control middleware is the central unit that supports the management of underlying Fog colonies. The management of fog colonies incorporates execution of fault tolerance processes over fog cells as well as novel device discovery and re-organisation of colonies if needed; Fog Orchestration Control Node supports a fog Colony constituted by diverse Fog Cells; and Fog Cells are software components running on Fog devices. Both the Fog orchestration control node and Cloud-Fog control middleware need to implement placement optimisation for tasks execution. The selected optimisation criterion in this work is twofold: first, to optimise resource utilisation at fog cells, and second, to minimize delays in propagating data to cloud. This hierarchical architecture is more complex than MobileFog's one, developing various Fog levels. Evaluation of the proposed model has been performed using an extension of CloudSim simulation framework for Fog Computing, resulting in 39% delays reduction.

5.3.2 Existing Products in the Market. **"Azure IoT Edge."** Azure IoT Suite Reference architecture [59] considers three central aspects for a typical IoT solution: device connectivity, data processing, analytics and management; and presentation and business connectivity. Recently Azure has announced the availability of Azure IoT Edge [4] as Open Source [5]. The provided open source software can run on Windows and Linux/Mac powered devices. IoT Edge modules are executed as Docker compatible containers. The IoT Edge Runtime provides monitoring and workload execution functionalities at the Edge.

It allows data pre-processing on-premises before sending it to Azure Cloud environments. The Microsoft services that can run on these devices include Azure Machine Learning, Stream Analytics Azure Functions, Microsoft's AI services and the Azure IoT Hub. Azure IoT Hub component contains device registry and identity store, as well as, device-to-edge and edge-to-device messaging features, acting as the entry point to access the rest of IoT suite services at Edge side. Azure IoT Hub presents an SDK that allows interoperability with custom gateways and simplified programming, Stream Analytics component offers real-time event processing so to support stream data analysis by processing telemetry, data aggregation, and event detection. On the Cloud side, Azure Storage offers long term data and object storage. This can be used in conjunction with Azure Web Apps and Microsoft Power BI, to have data visualisation means. At the time of writing, Azure IoT Edge can be used free of charge while associated use of Cloud services is billed based on usage.

"AWS Greengrass." AWS Greengrass [84, 85] offers an Edge computing platform that propounds local computing using AWS Serverless technology (AWS Lambda), messaging, data catching sync and ML inference while providing interoperability with AWS IoT Cloud services. It is a software stack available for any ARM and x86 device with minimum required capacity (1GHz of compute, 128MB of RAM, plus additional resources for workload and message throughput). At the time of writing, AWS Greengrass documentation details that compatibility tests have been validated with more than 40 devices. In addition, it offers direct communication and operation with Amazon FreeRTOS micro-controllers. The software stack is divided into three main pieces: AWS Greengrass Core, AWS GreenGrass SDK, and AWS IoT Device SDK. The Greengrass core allows for: local deployment of applications using lambda functions developed in Python 2.7, Node.JS 6.10, and Java 8; enables secured local messaging based on OPC-UA protocol; provides device management and device clones; and authentication and authorisation in device to cloud communication. AWS Greengrass SDK permits Lambda functions to interact with Core services. The extended IoT Device SDK endowed with Greengrass offers an extension to existing AWS IoT Device SDK to support constrained devices (supporting TLS) to communicate with Greengrass core. In addition, devices can use Greengrass discovery API to locate and manage secure communication to Greengrass core. A very interesting feature added recently is Greengrass ML. This feature allows ML models that have been developed and trained in the cloud, to be deployed and executed locally in the Greengrass core equipped device. This is reported to support GPU utilisation for devices that have it present.

It is worth remarking that AWS Greengrass supports the possibility to work offline (without Internet connection to the Cloud), performing synchronisation process when connectivity is ensured to the device. Logically, this has to be limited to the resources available on the device powering the AWS Greengrass Core, albeit no concrete information is presently found in the product information. Pricing for AWS Greengrass considers a combination of devices installed plus the usage of Cloud services these make. The price for devices can be charged monthly or with a fixed yearly amount.

5.4 Features Comparison

In Table 4, we present a comparison of features among all analysed architectures. The analysis compares features considered by research works and commercial offerings. In this analysis, the observed maturity of market developments possesses a remarkable nature. These today are considering advanced capabilities with regard to Data management, Edge workload execution models adapted to the last trends on the market, and even consideration of machine-learning frameworks. At the same time, analysed works in research elaborate on conceptual approaches and future requirements while existing implemented architectures are yet scarce. It is interesting to note that as previously introduced, OpenFog architecture limits Edge computing to intermediary nodes among IoT devices and Cloud, while considering Fog, as the computing continuum that embraces end-to-end management from IoT devices to Cloud. However, from the provided descriptions, it is clear that commercial Edge computing offerings, and specifically Amazon Greengrass, go far beyond providing an intermediate computing layer. Instead, these develop end-to-end solution for both IoT devices, computing at the edge and rich cloud services, commercial products make reality the computing continuum concept, nevertheless exposing its adopters to strong vendor lock-in.

According to current developments it can be the case that instead of research works feeding industry products with advanced features and ideas, it is research lagging behind industrial developments. This is to some extent corroborated by initial experimentation done in commercial offerings with rich IoT devices in connected vehicles [7, 71], which represents a clear initial step towards the realisation of Decentralised Cloud concept defined by this work. This experimentation while exploiting the inference of ML at the edge recognises the need of edge groups of devices and its communication. As it happened in the area of Grid, and the successful application of Cloud utility models in the market almost a decade ago. Nowadays, opportunities in research are apparently in scheduling, orchestration, and optimisation problems, instead of basic capabilities already being tackled in interesting approaches by commercial developments of major Cloud providers. These commercial offerings are advancing at an impressive rapid pace and getting quickly into quite mature stages that research and standardisation works have yet to achieve.

Table 4. Edge Frameworks Feature Comparison

Works	**Fog computing** [13]	**ANGELS** [64]	**MobileFog** [41]	**Nebula** [73]	**Resource** [89]	**AWS Green.** [84]	**Az.IoT Edge** [4]
Nature	Research	Research	Research	Research	Research	Comm.	Comm.
Resource	X	X	X	X	X	X	X
Fault Tolerance				X	X		
Data Mng.		X		X		X	X
Workload Mng.	X	X				X	X
Workload Scheduling					X		
Across Edge interop.							
Edge and Cloud interop.						X	X
Economy						X	
Eco-efficiency							
Incentives					X		
Sec. and Priv.						X	X
Connectivity and Resilience						X	X
Edge Infra.	Cluster	Server	Hierarch.	Cluster	Hierarch.	Server	Server
Offering	IaaS	PaaS	PaaS	PaaS	IaaS	PaaS	PaaS
Prog. Model/ Language				JavaScript		Lambda, containers	containers
Maturity	Arch.	Arch.	Arch.	Simulation	Simulation	Product	Product

6 CONCLUSIONS

This article has presented the current state-of-the-art and research challenges for future Decentralised Cloud models. In these, we observe that Mobile Cloud Computing has already developed a number of valuable tools and techniques that can significantly influence the future evolution of Cloud models. Specifically, in relation to Cloudlet and Edge Computing and Mobile Ad hoc Cloud.

Building its routes in Cloudlet concepts, we observe that still Edge Computing research is very much in a conceptual state. At the current state of development, multiple works have elaborated on diverse conceptual approaches for it; however, very few architectures do elaborate on management of specific aspects and still research gaps are appreciated in research challenges, such as across Edge execution models, Economy, Connectivity, and Resilience. Interestingly, while the research community is still debating the most appropriate term to use (Edge/Fog), major cloud providers are already launching significantly mature products to the market, even exploiting aspects such as ML inference at the Edge. This gives a clear indication on how promising Edge Computing developments are and the need for future research works to take into consideration commercially developed products in order not to re-invent the wheel.

At the same time, expected gains in complexity of the connected "intelligent things" will designate specific requirements to Decentralised Cloud Computing evolution [3]. Intelligent Things, such as robots and autonomous vehicles, can be viewed as mobile devices that provide complex aggregations of computing and storage resources together with diverse and heterogeneous sensors and actuators, which all together implement a cognitive loop. According to predictions "intelligent things" will become better, faster, and cheaper. Producers will soon be under pressure to provide complex behaviours, cognitive capabilities and skills at competitive costs, while increasing onboard computation and storage of smart machines will raise their costs, increase energy demand

and reduce their autonomy. The self-contained and self-sustaining nature of these novel "intelligent things" resources combined with their size and energy harvesting constrains will require of novel computing and communication architectures beyond state-of-the-art today.

A caveat has to be made in relation to Moore's law development. For the last few decades, overcoming similar challenges has always relied on the application of Moore's law. This has allowed producing ever better and capable hardware. Increasingly, Hardware manufacturers are encountering more difficulties in producing ever miniaturised low-power computing units that are cheaper and faster. This does not probably mean that computing progress will suddenly stall, but can have implications in the nature of that progress. The computing progress could be progressively changing to approaches that take better advantage of available resources while coping with the necessary balance among resources in high demand: computing and energy.

These environments are initially taking form in the evolution of Ad hoc Clouds, enabling smart collaboration among mobile devices. These build their routes in ad hoc networks and opportunistic computing. Further evolution of this concept is expected to enable the creation of dynamic ecosystems, meshes or swarms of "Intelligent Things" in fully distributed and decentralised manner in the so-called Decentralised Cloud.

At the same time, we are witnessing two very significant advances in AI and deep-learning technologies, which fuelled by the unstoppable data availability collected from "Intelligent Things" will soon increase computing demand by several orders of magnitude.

While the relation among these technologies is starting to be tackled by both research and commercial efforts [63, 79, 84], it further calls for development of Decentralised Cloud environments as ecosystems of "Intelligent Things" in which resource capacities are complimented by connection to other objects in the community. These have to be designed to allow the dynamic creation of dynamic device eco-systems, encompassing "Intelligent Things," cyber-physical devices, edge, and clouds, each of these adding to the collective capability and insight in a future computing continuum that will act as the backbone on which to build collective intelligence.

REFERENCES

[1] Saeid Abolfazli, Zohreh Sanaei, Ejaz Ahmed, Abdullah Gani, and Rajkumar Buyya. 2013. Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges. *CoRR* abs/1306.4. arXiv:arXiv:1306.4956.

[2] Mojtaba Alizadeh, Saeid Abolfazli, Mazdak Zamani, Sabariah Baaaharun, and Kouichi Sakurai. 2016. Authentication in mobile cloud computing: A survey. *J. Netw. Comput. Appl.* 61 (2016), 59–80. DOI: https://doi.org/10.1016/j.jnca.2015.10.005

[3] Ana Juan Ferrer, Robert Woitsch, Kyriakos Kritikos, George Kousiouris, Fotis Aisopos, David Garcia, Pierluigi Plebani, and Xavi Masip. 2017. *Future Cloud Research Roadmap, FP9 Cluster Inputs.* Technical Report. Retrieved from https://drive.google.com/file/d/0B4hHTKjZDMXGSGxoYnh4eXhURzA/view.

[4] Microsoft Azure. 2017. Azure IoT Edge. Retrieved from https://azure.microsoft.com/en-us/campaigns/iot-edge/.

[5] Microsoft Azure. 2017. Azure IoT GitHub. Retrieved from https://github.com/Azure/iot-edge/.

[6] Venkatramanan Balasubramanian and Ahmed Karmouch. 2017. An infrastructure as a service for mobile ad hoc cloud. In *Proceedings of the IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC'17).* DOI: https://doi.org/10.1109/CCWC.2017.7868393

[7] Jeff Barr. 2018. AWS IoT, Greengrass, and Machine Learning for Connected Vehicles at CES. Retrieved from https://aws.amazon.com/blogs/aws/aws-iot-greengrass-and-machine-learning-for-connected-vehicles-at-ces/.

[8] Kashif Bilal, Osman Khalid, Aiman Erbad, and Samee U. Khan. 2017. Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers. *Comput. Netw.* (2017). DOI: https://doi.org/10.1016/j.comnet.2017.10.002

[9] Salim Bitam, Sherali Zeadally, and Abdelhamid Mellouk. 2018. Fog computing job scheduling optimization based on bees swarm. *Enterpr. Info. Syst.* 12, 4 (2018), 373–397. DOI: https://doi.org/10.1080/17517575.2017.1304579

[10] Luiz F. Bittencourt, Javier Diaz-Montes, Rajkumar Buyya, Omer F. Rana, and Manish Parashar. 2017. Mobility-aware application scheduling in fog computing. *IEEE Cloud Comput.* 4, 2 (2017), 26–35. DOI: https://doi.org/10.1109/MCC.2017.27

[11] Z. S. Bojkovic, B. M. Bakmaz, and M. R. Bakmaz. 2017. Vision and enabling technologies of tactile internet realization. In *Proceedings of the 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS'17)*. 113–118. DOI: https://doi.org/10.1109/TELSKS.2017.8246242

[12] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. 2014. Fog computing: A platform for internet of things and analytics. In *Big Data and Internet of Things: A Roadmap for Smart Environments*. Springer, 169–186.

[13] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog computing and its role in the internet of things. In *Proceedings of the 1st Edition of the MCC Workshop on Mobile Cloud Computing*. 13–16. DOI: https://doi.org/10.1145/2342509.2342513

[14] Abhishek Chandra, Jon Weissman, and Benjamin Heintz. 2013. Decentralized edge clouds. *IEEE Internet Comput.* 17, 5 (Sep. 2013), 70–73. DOI: https://doi.org/10.1109/MIC.2013.93

[15] Jesse Chang, Rajesh Krishna Balan, and Satyanarayanan Satyanarayanan. 2005. *Exploiting Rich Mobile Environments*. Technical Report December. Retrieved from http://reports-archive.adm.cs.cmu.edu/anon/anon/usr0/ftp/home/ftp/2005/CMU-CS-05-199.pdf.

[16] Ruay-Shiung Chang, J. Gao, V. Gruhn, Jingsha He, G. Roussos, and Wei-Tek Tsai. 2013. Mobile cloud computing research - Issues, challenges and needs. In *Proceedings of the IEEE 7th International Symposium on Service Oriented System Engineering (SOSE'13)*. 442–453. DOI: https://doi.org/10.1109/SOSE.2013.96

[17] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. 2011. CloneCloud: Elastic execution between mobile device and cloud. In *Proceedings of the 6th Conference on Computer Systems (EuroSys'11)*. ACM, New York, NY, 301–314. DOI: https://doi.org/10.1145/1966445.1966473

[18] Byung-Gon Chun and Petros Maniatis. 2009. Augmented smartphone applications through clone cloud execution. In *Proceedings of the 12th conference on hot topics in operating systems (HotOS'09)*. USENIX Association, Berkeley, CA, 8. Retrieved from http://dl.acm.org/citation.cfm?id=1855568.1855576.

[19] Cisco Systems. 2016. Fog computing and the internet of things: Extend the cloud to where the things are. DOI: https://doi.org/10.1109/HotWeb.2015.22 arxiv:arXiv:1502.01815v3

[20] Marco Conti and Mohan Kumar. 2010. Opportunities in opportunistic computing. *Computer* 43, 1 (2010), 42–50. DOI: https://doi.org/10.1109/MC.2010.19

[21] Eduardo Cuervo, Aruna Balasubramanian, Dae-ki Cho, Alec Wolman, Stefan Saroiu, Ranveer Chandra, and Paramvir Bahl. 2010. MAUI: Making smartphones last longer with code offload. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys'10)*. ACM, New York, NY, 49–62. DOI: https://doi.org/10.1145/1814433.1814441

[22] Ruilong Deng, Rongxing Lu, Chengzhe Lai, Tom H. Luan, and Hao Liang. 2016. Optimal workload allocation in fog-cloud computing toward balanced delay and power consumption. *IEEE Internet Things J.* 3, 6 (2016), 1171–1181. DOI: https://doi.org/10.1109/JIOT.2016.2565516

[23] Hesham El-Sayed, Sharmi Sankar, Mukesh Prasad, Deepak Puthal, Akshansh Gupta, Manoranjan Mohanty, and Chin-Teng Lin. 2018. Edge of things: The big picture on the integration of edge, IoT and the cloud in a distributed computing environment. *IEEE Access* 6 (2018), 1706–1717. DOI: https://doi.org/10.1109/ACCESS.2017.2780087

[24] Nur Idawati Md. Enzai and Maolin Tang. 2014. A taxonomy of computation offloading in mobile cloud computing. In *Proceedings of the 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*. IEEE, 19–28. DOI: https://doi.org/10.1109/MobileCloud.2014.16

[25] ETSI. 2016. Mobile-edge Computing. Retrieved from http://www.etsi.org/images/files/ETSITechnologyLeaflets/MobileEdgeComputing.pdf.

[26] Niroshinie Fernando, Seng W. Loke, and Wenny Rahayu. 2011. Dynamic mobile cloud computing: Ad hoc and opportunistic job sharing. In *Proceedings of the 4th IEEE International Conference on Utility and Cloud Computing (UCC'11)*. 281–286. DOI: https://doi.org/10.1109/UCC.2011.45

[27] Niroshinie Fernando, Seng Wai Loke, and Wenny Rahayu. 2012. Mobile crowd computing with work stealing. In *Proceedings of the 15th International Conference on Network-Based Information Systems (NBiS'12)*. IEEE, 660–665.

[28] Niroshinie Fernando, Seng W. Loke, and Wenny Rahayu. 2013. Mobile cloud computing: A survey. *Future Gen. Comput. Syst.* 29, 1 (2013), 84–106. DOI: https://doi.org/10.1016/j.future.2012.05.023

[29] Jerry Gao, Volker Gruhn, Jingsha He, George Roussos, and Wei-Tek Tsai. 2013. Mobile cloud computing research - Issues, challenges and needs. In *Proceedings of the IEEE 7th International Symposium on Service-Oriented System Engineering*. IEEE, 442–453. DOI: https://doi.org/10.1109/SOSE.2013.96

[30] Pedro Garcia Lopez, Alberto Montresor, Dick Epema, Anwitaman Datta, Teruo Higashino, Adriana Iamnitchi, Marinho Barcellos, Pascal Felber, and Etienne Riviere. 2015. Edge-centric computing. *ACM SIGCOMM Comput. Commun. Rev.* 45, 5 (Sep. 2015), 37–42. DOI: https://doi.org/10.1145/2831347.2831354

[31] Gartner. 2017. The Edge Manifesto: Digital Business, Rich Media, Latency Sensitivity and the Use of Distributed Data Centers. Retrieved from https://www.gartner.com/doc/3104121?ref=SiteSearch&sthkw=micro data centres&fnl=search&srcId=1-3478922254.

[32] Lazaros Gkatzikis and Iordanis Koutsopoulos. 2013. Migrate or not? Exploiting dynamic task migration in mobile cloud computing systems. *IEEE Wireless Commun.* 20, 3 (2013), 24–32. DOI: https://doi.org/10.1109/MWC.2013.6549280

[33] GoogleTrends. 2018. Google Trends, fog computing vs. Edge Computing. Retrieved from https://trends.google.com/trends/explore?q=fogcomputing,edgecomputing.

[34] The OpenFog Consortium Architecture Working Group. 2016. Architecture Overview. Retrieved from https://www.openfogconsortium.org/wp-content/uploads/OpenFog-Architecture-Overview-WP-2-2016.pdf.

[35] Le Guan, Xu Ke, Meina Song, and Junde Song. 2011. A survey of research on mobile cloud computing. In *Proceedings of the 10th IEEE/ACIS International Conference on Computer and Information Science (ICIS'11).* 387–392. DOI: https://doi.org/10.1109/ICIS.2011.67

[36] Kiryong Ha, Zhuo Chen, Wenlu Hu, Wolfgang Richter, Padmanabhan Pillai, and Mahadev Satyanarayanan. 2014. Towards wearable cognitive assistance. In *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys'14).* ACM Press, New York, New York, 68–81. DOI: https://doi.org/10.1145/2594368.2594383

[37] Ahmed Hammam and Samah Senbel. 2014. A reputation trust management system for ad hoc mobile clouds. *Intell. Syst. Ref. Library* 70 (2014), 519–539. DOI: https://doi.org/10.1007/978-3-662-43616-5_20

[38] Ragib Hasan, Md. Mahmud Hossain, and Rasib Khan. 2015. Aura: An IoT-based cloud infrastructure for localized mobile computation outsourcing. In *Proceedings of the 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*. IEEE, 183–188. DOI: https://doi.org/10.1109/MobileCloud.2015.37

[39] HEADS Project. 2016. *D3.3. Final Framework of Resource-constrained Devices and Networks.* Technical Report. Retrieved from http://heads-project.eu/sites/default/files/HEADS%20D3.3%20V1.0.pdf.

[40] Ding Thai Hoang, Chonho Lee, Dusit Niyato, and Ping Wang. 2013. A survey of mobile cloud computing: Architecture, applications, and approaches. *Wireless Commun. Mobile Comput.* 13, 18 (2013), 1587–1611. DOI: https://doi.org/10.1002/wcm.1203

[41] Kirak Hong, David Lillethun, Umakishore Ramachandran, Beate Ottenwälder, and Boris Koldehofe. 2013. Mobile fog. In *Proceedings of the second ACM SIGCOMM Workshop on Mobile Cloud Computing (MCC'13).* ACM Press, New York, New York, 15. DOI: https://doi.org/10.1145/2491266.2491270

[42] Gonzalo Huerta-Canepa and Dongman Lee. 2010. A virtual cloud computing provider for mobile devices. In *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services Social Networks and Beyond (MCS'10).* ACM Press, New York, New York, 1–5. DOI: https://doi.org/10.1145/1810931.1810937

[43] Gonzalo Fernando Huerta Cánepa. 2012. *A Context-aware Application Offloading Scheme for a Mobile Peer-to-Peer Environment.* Ph.D. Dissertation. KAIST, South Korea.

[44] Seongah Jeong, Osvaldo Simeone, and Joonhyuk Kang. 2018. Mobile edge computing via a UAV-mounted cloudlet: Optimization of bit allocation and path planning. *IEEE Trans. Vehic. Technol.* 67, 3 (2018), 2049–2063. DOI: https://doi.org/10.1109/TVT.2017.2706308

[45] Albert Jonathan, Mathew Ryden, Kwangsung Oh, Abhishek Chandra, and Jon Weissman. 2017. Nebula: Distributed edge cloud for data intensive computing. *IEEE Trans. Parallel Distrib. Syst.* 9219, c (2017), 1–1. DOI: https://doi.org/10.1109/TPDS.2017.2717883

[46] Roelof Kemp, Nicholas Palmer, Thilo Kielmann, and Henri Bal. 2012. Cuckoo: A computation offloading framework for smartphones. In *Mobile Computing, Applications, and Services*, Martin Gris and Guang Yang (Eds.). Springer, Berlin, 59–79.

[47] Abdul Nasir Khan, M. L. Mat Kiah, Samee U. Khan, and Sajjad A. Madani. 2013. Towards secure mobile cloud computing: A survey. *Future Gen. Comput. Syst.* 29, 5 (2013), 1278–1299. DOI: https://doi.org/10.1016/j.future.2012.08.003

[48] Graham Kirby, Alan Dearle, Angus Macdonald, and Alvaro Fernandes. 2010. An approach to ad hoc cloud computing. *Arxiv preprint arXiv* (2010), 2–5. DOI: https://doi.org/arXiv:1002.4738v1 arxiv:1002.4738.

[49] Sokol Kosta, Andrius Aucinas, Pan Hui, Richard Mortier, and Xinwen Zhang. 2011. Unleashing the power of mobile cloud computing using ThinkAir. *CoRR* abs/1105.3 (2011). arXiv:arXiv:1105.3232v1

[50] Sokol Kosta, Andrius Aucinas, and Richard Mortier. 2012. ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. In *Proceedings of the IEEE INFOCOM.* IEEE, 945–953. DOI: https://doi.org/10.1109/INFCOM.2012.6195845

[51] Dejan Kovachev, Yiwei Cao, and Ralf Klamma. 2011. Mobile cloud computing: A comparison of application models. *CoRR* abs/1107.4 (2011). arxiv:1107.4940. Retrieved from http://arxiv.org/abs/1107.4940.

[52] Sara Kubik. 2017. A Plain Language Post about Fog Computing (That Anyone Can Understand). Retrieved from https://www.openfogconsortium.org/a-plain-language-post-about-fog-computing-that-anyone-can-understand/.

[53] Karthik Kumar, Jibang Liu, Yung-Hsiang Lu, and Bharat Bhargava. 2012. A survey of computation offloading for mobile systems. *Mobile Netw. Appl.* 18, 1 (Apr. 2012), 129–140. DOI: https://doi.org/10.1007/s11036-012-0368-0

[54] Kenji E. Kushida, Jonathan Murray, and John Zysman. 2015. Cloud computing: From scarcity to abundance. *J. Industry Compet. Trade* 15, 1 (2015), 5–19. DOI: https://doi.org/10.1007/s10842-014-0188-y

[55] Hyun Jung La and Soo Dong Kim. 2014. A taxonomy of offloading in mobile cloud computing. In *Proceedings of the IEEE 7th International Conference on Service-Oriented Computing and Applications*. IEEE, 147–153. DOI: https://doi.org/10.1109/SOCA.2014.22

[56] Edward A. Lee, Jan Rabaey, Bjorn Hartmann, John Kubiatowicz, Kris Pister, Alberto Sangiovanni-Vincentelli, Sanjit A. Seshia, John Wawrzynek, David Wessel, Tajana Simunic Rosing, David Blaauw, Prabal Dutta, Kevin Fu, Carlos Guestrin, Ben Taskar, Roozbeh Jafari, Douglas Jones, Vijay Kumar, Rahul Mangharam, George J. Pappas, Richard M. Murray, and Anthony Rowe. 2014. The swarm at the edge of the cloud. *IEEE Design Test* 31, 3 (June 2014), 8–20. DOI: https://doi.org/10.1109/MDAT.2014.2314600

[57] Seng W. Loke, Keegan Napier, Abdulaziz Alali, Niroshinie Fernando, and Wenny Rahayu. 2015. Mobile computations with surrounding devices. *ACM Trans. Embed. Comput. Syst.* 14, 2 (Feb. 2015), 1–25. DOI: https://doi.org/10.1145/2656214

[58] Eugene E. Marinelli. 2009. *Hyrax: Cloud Computing on Mobile Devices Using MapReduce*. Ph.D. Dissertation. Retrieved from http://www.contrib.andrew.cmu.edu/emarinel/masters_thesis/emarinel_ms_thesis.pdf.

[59] Microsoft. 2016. *Azure IoT Reference Architecture*. Technical Report. Retrieved from http://download.microsoft.com/download/A/4/D/A4DAD253-BC21-41D3-B9D9-87D2AE6F0719/Microsoft_Azure_IoT_Reference_Architecture.pdf.

[60] Emiliano Miluzzo, Ramón Cáceres, and Yih-Farn Chen. 2012. Vision. In *Proceedings of the 3rd ACM Workshop on Mobile Cloud Computing and Services (MCS'12)*. ACM Press, New York, New York, 9. DOI: https://doi.org/10.1145/2307849.2307854

[61] Amitabh Mishra and Gerald Masson. 2013. MoCCA: A mobile cellular cloud architecture. *J. Cyber Secur. Mobil.* 2, 2 (2013), 105–125. DOI: https://doi.org/10.13052/jcsm2245-1439.221

[62] Muhammad Baqer Mollah, Md. Abul Kalam Azad, and Athanasios Vasilakos. 2017. Security and privacy challenges in mobile cloud computing: Survey and way ahead. *J. Netw. Comput. Appl.* 84 (2017), 38–54. DOI: https://doi.org/10.1016/j.jnca.2017.02.001

[63] Ahsan Morshed, Prem Prakash Jayaraman, Timos Sellis, Dimitrios Georgakopoulos, Massimo Villari, and Rajiv Ranjan. 2018. Deep osmosis: Holistic distributed deep learning in osmotic computing. *IEEE Cloud Comput.* 4, 6 (2018), 22–32. DOI: https://doi.org/10.1109/MCC.2018.1081070

[64] Arijit Mukherjee, Himadri Sekhar Paul, Swarnava Dey, and Ansuman Banerjee. 2014. ANGELS for distributed analytics in IoT. In *Proceedings of the IEEE World Forum on Internet of Things (WF-IoT'14)*. IEEE, 565–570. DOI: https://doi.org/10.1109/WF-IoT.2014.6803230

[65] NIST SP 800-145. 2011. *A NIST Definition of Cloud Computing*. Technical Report. Retrieved from http://csrc.nist.gov/publications/drafts/800-145/Draft-SP-800-145_cloud-definition.pdf.

[66] Oded Nov, David Anderson, and Ofer Arazy. 2010. Volunteer computing: A model of the factors determining contribution to community-based scientific research. In *Proceedings of the 19th International Conference on World Wide Web (WWW'10)*. ACM Press, New York, New York, 741. DOI: https://doi.org/10.1145/1772690.1772766

[67] OpenFog. 2016. OpenFog Consortium. Retrieved from https://www.openfogconsortium.org.

[68] Xuan Qui Pham and Eui Nam Huh. 2016. Towards task scheduling in a cloud-fog computing system. In *Proceedings of the 18th Asia-Pacific Network Operations and Management Symposium, APNOMS 2016: Management of Softwarized Infrastructure*. DOI: https://doi.org/10.1109/APNOMS.2016.7737240

[69] DITAS Project. 2017. Data-intensive applications improvement by movind data in mixed cloud/fog environments. Retrieved from http://www.ditas-project.eu/.

[70] Han Qi and Abdullah Gani. 2012. Research on mobile cloud computing: Review, trend and perspectives. In *Proceedings of the 2nd International Conference on Digital Information and Communication Technology and Its Applications (DICTAP'12)*. 195–202. DOI: https://doi.org/10.1109/DICTAP.2012.6215350

[71] Chris Rec. 2018. Building Connected Vehicle Solutions on the AWS Cloud. Retrieved from https://aws.amazon.com/blogs/iot/building-connected-vehicle-solutions-on-the-aws-cloud/.

[72] Lukas Reinfurt, Uwe Breitenbücher, Michael Falkenthal, Frank Leymann, and Andreas Riegg. 2017. Internet of things patterns for devices. In *Proceedings of the 9th International Conferences on Pervasive Patterns and Applications (PATTERNS'17)*. Xpert Publishing Services, 117–126.

[73] Mathew Ryden, Kwangsung Oh, Abhishek Chandra, and Jon Weissman. 2014. Nebula: Distributed edge cloud for data-intensive computing. In *Proceedings of the International Conference on Collaboration Technologies and Systems (CTS'14)*. IEEE, 491–492. DOI: https://doi.org/10.1109/CTS.2014.6867613

[74] Zohreh Sanaei, Saeid Abolfazli, Abdullah Gani, and Rajkumar Buyya. 2014. Heterogeneity in mobile cloud computing: Taxonomy and open challenges. *IEEE Commun. Surveys Tutor.* 16, 1 (2014), 369–392. DOI: https://doi.org/10.1109/SURV.2013.050113.00090 arxiv:1305.0982

[75] Zohreh Sanaei, Saeid Abolfazli, Abdullah Gani, and Rashid Hafeez. 2012. Tripod of requirements in horizontal heterogeneous mobile cloud computing. In *Proceedings of CISCO'12*. CISCO, 217–222.
[76] Zohreh Sanaei, Saeid Abolfazli, Abdullah Gani, and Muhammad Shiraz. 2012. SAMI: Service-based arbitrated multi-tier infrastructure for mobile cloud computing. In *Proceedings of the 1st IEEE International Conference on Communications in China Workshops (ICCC'12)*. 14–19. DOI: https://doi.org/10.1109/ICCCW.2012.6316466
[77] Mahadev Satyanarayanan. 1993. Mobile computing. *Computer* 26, 9 (Sep. 1993), 81–82. DOI: https://doi.org/10.1109/2.231283
[78] Mahadev Satyanarayanan. 1996. Fundamental challenges in mobile computing. *Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing (PODC'96)*, 1–7. DOI: https://doi.org/10.1145/248052.248053
[79] Mahadev Satyanarayanan. 2017. The emergence of edge computing. *Computer* 50, 1 (2017), 30–39. DOI: https://doi.org/10.1109/MC.2017.9
[80] Mahadev Satyanarayanan, Paramvir Bahl, Ramon Caceres, and Nigel Davies. 2009. The case for VM-based cloudlets in mobile computing. *IEEE Pervas. Comput.* 8, 4 (2009), 14–23. DOI: https://doi.org/10.1109/MPRV.2009.82
[81] Mahadev Satyanarayanan, Zhuo Chen, Kiryong Ha, Wenlu Hu, Wolfgang Richter, and Padmanabhan Pillai. 2014. Cloudlets: At the leading edge of mobile-cloud convergence. In *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services*. ICST, 1–9. DOI: https://doi.org/10.4108/icst.mobicase.2014.257757
[82] Mahadev Satyanarayanan, Rolf Schuster, Maria Ebling, Gerhard Fettweis, Hannu Flinck, Kaustubh Joshi, and Krishan Sabnani. 2015. An open ecosystem for mobile-cloud convergence. *IEEE Commun. Mag.* 53, 3 (Mar. 2015), 63–70. DOI: https://doi.org/10.1109/MCOM.2015.7060484
[83] Mahadev Satyanarayanan, Pieter Simoens, Yu Xiao, Padmanabhan Pillai, Zhuo Chen, Kiryong Ha, Wenlu Hu, and Brandon Amos. 2015. Edge analytics in the internet of things. *IEEE Pervas. Comput.* 14, 2 (Apr. 2015), 24–31. DOI: https://doi.org/10.1109/MPRV.2015.32
[84] Amazon Web Services. 2017. Amazon Web Services Greengrass. Retrieved from https://aws.amazon.com/greengrass/.
[85] Amazon Web Services. 2017. Amazon Web Services Greengrass FAQs. Retrieved from https://aws.amazon.com/greengrass/faqs/.
[86] Weisong Shi, Jie Cao, Quan Zhang, Youhuizi Li, and Lanyu Xu. 2016. Edge computing: Vision and challenges. *IEEE Internet Things J.* 3, 5 (2016), 637–646. DOI: https://doi.org/10.1109/JIOT.2016.2579198
[87] Devu Manikantan Shila, Wenlong Shen, Yu Cheng, Xiaohua Tian, and Xuemin Sherman Shen. 2017. AMCloud: Toward a secure autonomic mobile ad hoc cloud computing system. *IEEE Wireless Commun.* 24, 2 (2017), 74–81. DOI: https://doi.org/10.1109/MWC.2016.1500119RP
[88] Muhammad Shiraz, Abdullah Gani, Rashid Hafeez Khokhar, and Rajkumar Buyya. 2013. A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing. *IEEE Commun. Surveys Tutor.* 15, 3 (2013), 1294–1313. DOI: https://doi.org/10.1109/SURV.2012.111412.00045
[89] Olena Skarlat, Stefan Schulte, and Michael Borkowski. 2016. Resource provisioning for IoT services in the fog resource provisioning for IoT services in the fog. In *Proceedings of the IEEE 9th International Conference on Service-Oriented Computing and Applications Resource*. DOI: https://doi.org/10.1109/SOCA.2016.10
[90] Ivan Stojmenovic. 2012. Keynote 1: Mobile cloud and green computing. *Procedia Comput. Sci.* 10 (Jan. 2012), 18–19. DOI: https://doi.org/10.1016/j.procs.2012.06.004
[91] The Economist. 2016. After Moore's law, The future of computing. Retrieved from https://www.economist.com/News/Leaders/21694528-Era-Predictable-Improvement-Computer-Hardware-Ending-What-Comes-Next-Future.
[92] Rico Valentino, Woo-sung Jung, and Young-bae Ko. 2018. Opportunistic computational offloading system for clusters of drones. In *Proceedings of the 20th International Conference on Advanced Communication Technology (ICACT'18)*. 303–306. DOI: https://doi.org/10.23919/ICACT.2018.8323734
[93] Luis M. Vaquero and Luis Rodero-Merino. 2014. Finding your way in the fog. *ACM SIGCOMM Comput. Commun. Rev.* 44, 5 (2014), 27–32. DOI: https://doi.org/10.1145/2677046.2677052
[94] Blesson Varghese and Rajkumar Buyya. 2018. Next generation cloud computing: New trends and research directions. *Future Gen. Comput. Syst.* 79 (2018), 849–861. DOI: https://doi.org/10.1016/j.future.2017.09.020 arxiv:1707.07452
[95] Massimo Villari, Maria Fazio, Schahram Dustdar, Omer Rana, and Rajiv Ranjan. 2016. Osmotic computing: A new paradigm for edge/cloud integration. *IEEE Cloud Comput.* 3, 6 (2016), 76–83. DOI: https://doi.org/10.1109/MCC.2016.124
[96] Marcelo Yannuzzi, Rodolfo Milito, Rene Serral-Gracia, Diego Montero, and Mario Nemirovsky. 2014. Key ingredients in an IoT recipe: Fog computing, cloud computing, and more fog computing. In *Proceedings of the IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD'14)*. IEEE, 325–329. DOI: https://doi.org/10.1109/CAMAD.2014.7033259
[97] Ibrar Yaqoob, Ejaz Ahmed, Gani Abdullah, Mokhtar Salimah, Imran Muhammad, and Guizani Sghaier. 2016. Mobile ad hoc cloud: A survey. *Wireless Commun. Mobile Comput.* 16, 16 (2016), 2572–2589. DOI: https://doi.org/10.1002/wcm.2709

[98] Ibrar Yaqoob, Ejaz Ahmed, Abdullah Gani, Salimah Mokhtar, and Muhammad Imran. 2017. Heterogeneity-aware task allocation in mobile ad hoc cloud. *IEEE Access* 5 (2017), 1779–1795. DOI: https://doi.org/10.1109/ACCESS.2017.2669080

[99] Bilel Zaghdoudi, Hella Kaffel Ben Ayed, and Islem Gnichi. 2017. A protocol for setting up ad hoc mobile clouds over spontaneous MANETs: A proof of concept. In *Proceedings of the Conference on Cloudification of the Internet of Things (CIoT'17)*, 1–6. DOI: https://doi.org/10.1109/CIOT.2016.7872919

[100] Bilel Zaghdoudi, Hella Kaffel Ben Ayed, and Imen Riabi. 2015. Ad hoc cloud as a service: A protocol for setting up an ad hoc cloud over MANETs. *Procedia Comput. Sci.* 56, 1 (2015), 573–579. DOI: https://doi.org/10.1016/j.procs.2015.07.256

[101] Yang Zhang, Dusit Niyato, and Ping Wang. 2015. Offloading in mobile cloudlet systems with intermittent connectivity. *IEEE Trans. Mobile Comput.* 14, 12 (Dec. 2015), 2516–2529. DOI: https://doi.org/10.1109/TMC.2015.2405539

[102] Longzhao Zhong, Beizhan Wang, and Haifang Wei. 2012. Cloud computing applied in the mobile Internet. In *Proceedings of the 7th International Conference on Computer Science and Education (ICCSE'12)*. IEEE, 218–221. DOI: https://doi.org/10.1109/ICCSE.2012.6295061

Received November 2017; revised July 2018; accepted July 2018