

# Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges

Zohreh Sanaei, *Member, IEEE*, Saeid Abolfazli, *Member, IEEE*, Abdullah Gani, *Senior Member, IEEE* and Rajkumar Buyya, *Senior Member, IEEE*

**Abstract**—The unabated flurry of research activities to augment various mobile devices by leveraging heterogeneous cloud resources has created a new research domain called Mobile Cloud Computing (MCC). In the core of such a non-uniform environment, facilitating interoperability, portability, and integration among heterogeneous platforms is nontrivial. Building such facilitators in MCC requires investigations to understand heterogeneity and its challenges over the roots. Although there are many research studies in mobile computing and cloud computing, convergence of these two areas grants further academic efforts towards flourishing MCC. In this paper, we define MCC, explain its major challenges, discuss heterogeneity in convergent computing (i.e. mobile computing and cloud computing) and networking (wired and wireless networks), and divide it into two dimensions, namely vertical and horizontal. Heterogeneity roots are analyzed and taxonomized as hardware, platform, feature, API, and network. Multidimensional heterogeneity in MCC results in application and code fragmentation problems that impede development of cross-platform mobile applications which is mathematically described. The impacts of heterogeneity in MCC are investigated, related opportunities and challenges are identified, and predominant heterogeneity handling approaches like virtualization, middleware, and service oriented architecture (SOA) are discussed. We outline open issues that help in identifying new research directions in MCC.

**Index Terms**—Mobile Cloud Computing, Vertical and Horizontal heterogeneity, Mobile computation offloading, Interoperability, Portability, Seamless communication.

## I. INTRODUCTION

THE POPULARITY and availability of mobile devices such as smartphones, which are offering ubiquitous communication and information services, are creating such a dependency that one does not leave home without them. Smartphones possess a likelihood of one-upmanship from notebooks and desktops [1]. These smart handheld computers will generate 150.6-billion-dollar benefits by the end of 2014, while devices like PDAs will bring in 2.7 billion dollars comparatively [2]. Hence, with the increasing ubiquity of smartphones, the greater demand for heavier applications with high quality of experience is gaining ground. However,

smartphones' miniature nature, lightness, and mobility impose intrinsic limits on their processing abilities, battery lifetime, storage capacity, and visualization power (e.g. screen size and rendering capability). These restraints impede executing resource-intensive tasks and storing large amount of data on smartphones.

To overcome some of these limitations, Satyanarayana [3] proposes cyber foraging approach. In this approach, mobile augmentation is achieved via offloading applications (entirely or partially) to resource-rich, non-mobile computing devices in the vicinity called surrogates to provide heavy functionality and to conserve local resources, especially energy. Proposals similar to Spectra [4] and Scavenger [5] are aimed to augment mobile devices leveraging cyber foraging. However, safety and reliability are the major challenges for migrating codes to the surrogates in the absence of authorities supervising functionality, performance, and the reliability of surrogates [6].

Alternatively, cloud computing [7]–[9] as a distributed computing paradigm has recently obtained momentous ground in augmenting mobile devices. MAUI [10], CloneCloud [11], and Elastic Applications [12] are some efforts that exploit clouds infrastructures to mitigate resource poverty of mobile devices. ABI Research forecasts that by the end of 2015 more than 240 million mobile business users will utilize cloud services driving nearly \$5 billion in revenues [13]. Deploying cloud resources is a complex task that obliges programmers to acquire specific skills. To mitigate such design and programming overhead, cloud computing software platforms such as Eucalyptus [14] was developed to facilitate cloud deployment and management in IT business.

Recently, the exploitation of cloud resources for augmenting mobile devices has bred a new research area called Mobile Cloud Computing (MCC). Mobility established on convergent networks [15] is the key attribute that distinguishes MCC from cloud computing. MCC aims to augment computing capabilities of mobile devices, conserve local resources - especially battery, extend storage capacity, and enhance data safety to enrich the computing experience of mobile users. The main difference between surrogate-based and cloud-based augmenting approaches is that surrogates offer free services without commitment to complete assigned jobs (they can leave a task anytime at any stage of computing), whereas clouds provide paid services with assured availability, quality, and commitment according to the negotiated Service-Level Agreement (SLA) between cloud vendor and mobile client [16]. Nevertheless, MCC is a nascent technology restrained by heterogeneity that still requires a plethora of research and development for deployment in real augmentation scenarios.

Manuscript received May 5, 2012; revised November 14, 2012 and February 28, 2013. This work is funded by the Malaysian Ministry of Higher Education under the University of Malaya High Impact Research Grant - UM.C/HIR/MOHE/FCSIT/03.

Z. Sanaei, S. Abolfazli, and A. Gani are with the Department of Computer System & Technology, The University of Malaya, Kuala Lumpur, Malaysia (e-mail: {sanaei,abolfazli}@ieee.org; abdullah@um.edu.my).

R. Buyya is with the Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computing and Information, The University of Melbourne, Parkville, Melbourne, VIC 3010, Australia (e-mail: raj@csse.unimelb.edu.au).

Digital Object Identifier 10.1109/SURV.2013.050113.00090

MCC is a convergent technology comprised of three cornerstone heterogeneous technologies, namely mobile computing, cloud computing, and networking. Such heterogeneity aggregation complicates MCC and decelerates its success. In the mobile computing landscape, the vast variety of mobile devices with different operating systems (OSs), platforms, and wireless network standards is making mobile application development a major challenge for developers [17]. Also, from the mobile user perspective, application and data migration incur redundant monetary and temporal costs in the case of shifting from one platform to another.

In the cloud ecosystem, the rapid development of cloud computing in the absence of early standards beside fragmented business policies have created various types of heterogeneity (e.g. hardware and platform) that give rise to new problems such as vendor lock-in. Vendor lock-in problem is the state when code and data cannot (easily) be moved from one cloud to another because of dissimilarities among underlying architectures and programming languages. Though the problem might be attractive in business competition [18], it causes several challenges particularly data integrity, interoperability, and portability. Hence, cloud-users become vulnerable to problems like (i) reliability if the service provider stops serving requests, (ii) data extraction from the hosting server(s), and (iii) a price hike.

Furthermore, the convergence of heterogeneous wireless networks (e.g. WLAN and cellular) and wired networks makes MCC more challengeable than cloud computing. Intermittency, low security, and a high probability of signal interception in wireless networks are factors that degrade communication quality and increase application response time. These challenges are intensified when mobile users cross the Internet channel to utilize cloud services.

To actuate MCC vision, essential requirements like (i) managing distributed data between heterogeneous cloud servers [19], (ii) unifying network connectivity, (iii) utilizing IP-based wireless systems [20], and (iv) establishing common application programming interface (API) standards for mobile services (e.g. GSMA OneAPI [21]) as well as cloud services (e.g. OCCI [22] and OVF [23]) should be met. Furthermore, underpinning requisitions i.e. ubiquity, energy efficiency, and trust can enhance user experience quality toward unleashing the power of MCC [24].

To address the challenges within MCC, academic and industrial communities are bound together. Academic proposals particularly Cloudlet [25], Mirage [26], RASIC [27], and Cloud Mobile Hybrid (CMH) Applications [28] are sought to mitigate heterogeneity-caused problems by leveraging approaches particularly virtualization, middleware, semantic, Service Oriented Architecture (SOA), and Domain Specific Language (DSL). In industry, Marmalade [29] and PhoneGap [30] are some of the cross-platform toolkits that aim to reduce developers' workload by automatically transiting the mobile application code to the required platform (e.g. Android<sup>1</sup> and Blackberry<sup>2</sup>). However, building cloud-mobile applications in two widely divergent environments (i.e. mobile and cloud)

is maturing and necessitates further research. Surveys such as [31], [32] provide extensive overview of MCC including some challenges and advantages that sufficiently reflect the current trend in MCC. However, comprehensive study on heterogeneity in MCC as a root of great number of challenges is lacking.

This article provides a comprehensive study on MCC heterogeneity, presents MCC definition, and describes challenges that are either caused or intensified by heterogeneity. Taxonomy of heterogeneity roots in MCC is devised according to the fragmentation in mobile computing, cloud computing, and networking environments. Heterogeneity in MCC is analyzed and classified into vertical and horizontal aspects based on heterogeneity raised in mobile devices, clouds, and wireless networks. Furthermore, application and data fragmentation are described and the impacts of heterogeneity in MCC are presented. The survey ends by reviewing major heterogeneity handling techniques and describing few open issues as future research directions. In the burgeoning heterogeneous MCC domain, the comprehension and settlement of fundamental problems in the early development stages leads to the success of MCC. Considering the dominating popularity of smartphones compared to other mobile devices, this article studies smartphones as the main mobile computing devices. The terms mobile devices and smartphones are used interchangeably with the same notion.

The remainder of this paper is organized as follows. The next Section presents an overview of MCC and discusses its major challenges. Section III devises taxonomy of heterogeneity roots and describes heterogeneity dimensions (vertical and horizontal) in MCC. Impacts of heterogeneity in MCC is identified in Section IV. Technologies and techniques for alleviating the challenges of heterogeneity are discussed in Section V. Finally, Section VI highlights open issues and the paper concludes in Section VII.

## II. MOBILE CLOUD COMPUTING (MCC)

Resource poverty is a major shortcoming of mobile computing. Recently, the connotation of MCC has arisen in order to overcome the resource deficiency of mobile devices, especially smartphones. In order to bring a clear understanding of MCC and its challenges, we examine the respective definitions of mobile computing, ubiquitous computing, cloud, and cloud computing.

### A. Definitions

There are several definitions in mobile computing [33], [34], ubiquitous computing [35], [36], cloud [7], and cloud computing [8]. Here are some of the definitions deemed generic enough to understand the fundamentals.

**Mobile Computing.** The design of small, powerful devices enables mobility in wireless networks that supports a trend toward computing on the go, known as mobile computing.

Satyanarayana [37] describes the vision of mobile computing as “information at fingertips anywhere, anytime”, while Imielinski and Korth [33] discuss that “The mobile computing no longer requires users to maintain a fixed and universally

<sup>1</sup><http://www.android.com/>

<sup>2</sup><http://us.blackberry.com/>

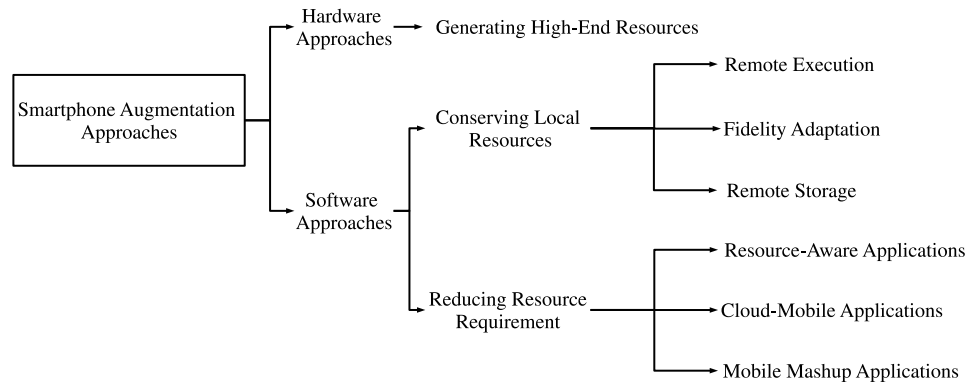


Fig. 1. Taxonomy of Smartphone Augmentation Approaches.

known position in the network and enables almost unrestricted mobility”.

**Ubiquitous Computing.** Mark Weiser [35] defines ubiquitous computing as “the method of enhancing computer use by making many computers available throughout the physical environment, but making them effectively invisible to the user”.

**Cloud.** Cloud is a rich computing resource. Buyya, Yeo, and Venugopal [7] propose one of the most popular definitions for cloud as “a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers”.

**Cloud Computing.** Cloud computing is a computing technology that leverages cloud’s resources for “enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction”, as NIST [8] describes.

Based on these definitions and our observation of the essence of what and how MCC landscape is deemed, the following definition is proposed.

**Mobile Cloud Computing** is 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 channel of Ethernet or Internet regardless of heterogeneous environments and platforms based on the pay-as-you-use principle.

## B. Vision

MCC is the state-of-the-art mobile computing technology that aims to augment a multitude of mobile devices, especially smartphones and alleviate their resource poverty. Mobile users can have access to their applications, data, and cloud services through the Internet by leveraging mobile web [31]. This futuristic accomplishment will be employed in several areas like healthcare (e.g. telemonitoring and telesurgery), education, IT Business, rural and urban development, and social

networking. In our previous work [38], smartphone augmentation approaches are reviewed and a taxonomy is defined which is depicted in Figure 1. Technological advancement in manufacturing high-end mobile resources is slower than the ever-growing expectations of mobile users and application requirements. Hence, soft resource augmentation is necessary for delivering user-centric computing capabilities [3] equal to user expectations. We advocated that cloud computing is the predominant technology recently deployed to augment smartphones by reducing application resource requirements. Several efforts such as [10], [12], [39]–[42] deploy cloud computing technology to enhance the capability of smartphone.

Moreover, cloud computing is beneficial in enhancing information safety and security. Storing data in smartphones’ local storage is a hazardous practice due to their susceptibility to theft, loss, and physical damage. Cloud data storage is envisioned to enhance data safety and security, provide pervasive accessibility, and facilitate data portability and synchronization among several devices (e.g. smartphones and PCs). DropBox<sup>3</sup>, SugarSync<sup>4</sup>, and Box<sup>5</sup> are examples of cloud storage services. People exploit such huge data warehouses to store and retrieve their data (bulk data) which are accessible from various devices. Users can even access their data through the Internet by utilizing public devices and providing unique credentials.

The advent of MCC has advanced into a technological revolution providing profitable opportunities for several domains such as healthcare, e-learning, and the tourism industry. It connotes the impression to reduce development cost and stimulate execution of resource-intensive mobile applications by leveraging distant rich resources to enhance the quality of user experience. Figure 2 illustrates a conceptual view of MCC and depicts its usability in several domains such as healthcare, social networking, urban development, and vehicular technology. It shows the possibility of utilizing geographically distributed private clouds (e.g. medical and biological research groups), public clouds (e.g. Google<sup>6</sup> and Facebook<sup>7</sup>), and hybrid clouds (that can be generated by

<sup>3</sup><https://www.dropbox.com/>

<sup>4</sup><http://www.sugarsync.com/>

<sup>5</sup><http://www.box.com/>

<sup>6</sup><https://www.google.com>

<sup>7</sup><http://www.facebook.com>

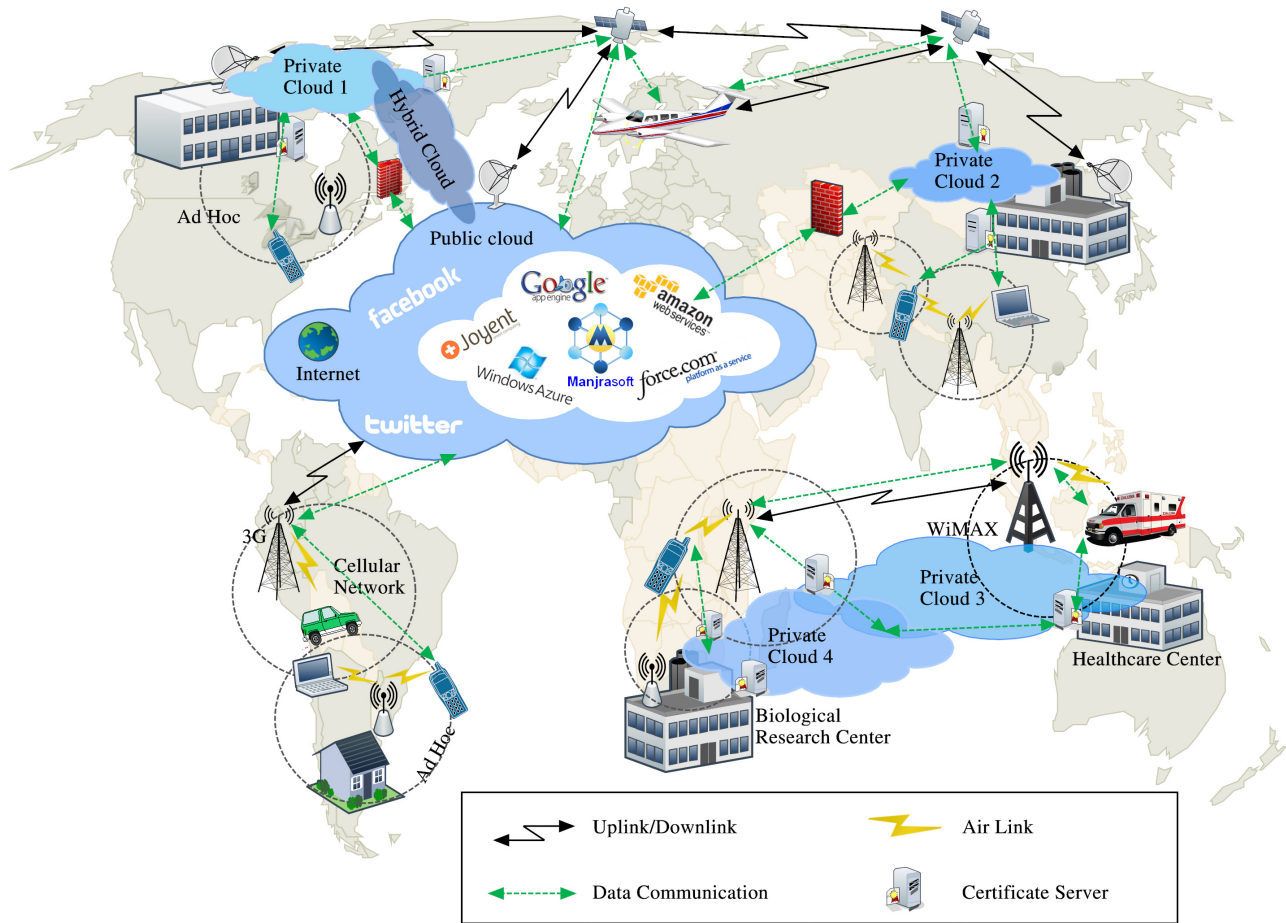


Fig. 2. A Conceptual View of Mobile Cloud Computing.

converging private and public clouds) in global roaming. It is notable that authorized mobile users can utilize more than one private cloud simultaneously. Furthermore, Figure 2 illustrates heterogeneity in MCC among varied mobile devices, communication networks, and clouds. Although heterogeneity has been existing in mobile and cloud computing domains, accumulated intensity and complexity make it a unique, challenging feature within MCC that necessitates a comprehensive study.

### C. Heterogeneity

Heterogeneity in MCC is the existence of differentiated hardware, architectures, infrastructure, and technologies of mobile devices, clouds, and wireless networks. The cutting edge technologies are expected to initiate and facilitate collaboration among these heterogeneous computing devices toward unrestricted mobile computing.

**Heterogeneity in Mobile Devices:** Software, hardware, and technology variation among mobile devices cause heterogeneity in this domain. Moreover, increasing popularity of smartphones creates a dynamic and demanding market that disperse them to different dimensions, e.g. brand, hardware, OS, feature, and communication medium. Consequently, device-level collaboration becomes more challenging in MCC.

**Heterogeneity in Clouds:** Numerous cloud vendors provide different services with custom-built policies, infrastructures,

platforms, and APIs that make the cloud landscape heterogeneous. Such variations cause interoperability and portability [43] as major challenges in cloud computing. There is a notion [18] that business competition also diversifies cloud providers with their heterogeneous frameworks, exacerbating heterogeneity on the cloud side.

**Heterogeneity in Wireless Networks:** In MCC, the majority of communications take place in the wireless network environment which is a heterogeneous communication medium. Variations in wireless networks and their related technologies impact the delivery of cloud services and affect mobility, augmentation, and usability of smartphones [38].

### D. Challenges

In the MCC landscape, an amalgam of mobile computing, cloud computing, and communication networks (to augment smartphones) creates several complex challenges. This part describes challenges either stemmed or intensified by heterogeneity. Although some of these challenges such as seamless connectivity, vendor lock-in, and security and privacy are common with mobile computing and cloud computing, we aim to highlight how different are these traditional challenges and in what extent they increase the complexity in MCC.

- *Mobile Computation Offloading:* Leveraging heterogeneous cloud resources to augment computing limitations

of multitude of mobile devices towards realizing the vision of unrestricted functionality, storage, and mobility in current diverse communication environment is a non-trivial task. Realizing cloud-based augmentation vision is impeded by multi-dimensional overhead of identifying and efficiently partitioning resource intensive components, VM (Virtual Machine) creation and migration, and monitoring the overall outsourcing process [44]. Moreover, a plethora of hurdles in utilizing cloud resources such as latency, heterogeneity, security (both offloaded code and cloud permanent software), code portability, and cloud-mobile interoperability intensifies the situation. Although several efforts such as CloneCloud [11], MAUI [10], Cloudlet [25], SAMI [45], and MOMCC [46] are facilitating and promoting cloud-based computation offloading, still research and development in this domain remains a top priority.

- *Seamless Connectivity*: Wireless networks are characterized by low-bandwidth, intermittent, and less-reliable transmission grounds compared with the wired networks. Establishing and maintaining seamless sessions between nomadic MCC users (leveraging cloud computation or storage) and other entities (e.g. smartphones and clouds) in a wireless medium composed of heterogeneous network technologies are critical issues to fully unleash the power of MCC [24]. The intermittent connectivity causes several challenges such as dismissal of always-on connectivity, excessive consumption of limited mobile resources, and disproportionate delaying of application execution that sharply degrade quality of computing services.

Seamless connectivity in heterogeneous MCC environment demands reliable inter-system signal handoff schemes along next generation wireless networks [47]. Unstable wireless bandwidth, low wireless network security, and the high probability of signal interception are challenges likely intensified by crossing the Internet channel in order to utilize the cloud services for mobile augmentation.

- *Long WAN Latency*: Latency adversely impacts on the energy efficiency [48] and interactive response [49] of cloud-mobile applications by consuming excessive mobile resources and raising transmission delays. In cellular communication, distance from the base station (near or far) and variation in bandwidth and speed of various wireless technologies affect the energy efficiency and usability of MCC devices. For example, data transfer bit-rate consumes comparatively more energy in cellular networks than WLAN. The higher the transmission bit-rate, the more energy efficient the transmission [48]. Moreover, leveraging wireless Internet networks to offload mobile intensive applications to distant cloud resources creates a bottleneck. Consequently, the long WAN latency is increased while the quality of user experience is decreased. To reduce interaction latency, proposals such as Cloudlet, MOMCC, and SAMI are proposed to create a proximate cloud to access nearby remote resources, but further advancement to achieve crisper response is required.

- *Mobility Management*: Integration of dissimilar wireless networks is a challenging task due to heterogeneity in access technologies, architectures, protocols, user mobility pattern, and user service requirements [20]. To achieve the converged wired, mobile, and broadband communication not only accurate and efficient mobility management schemes are prerequisite, but also seamless integration and interoperation of mobility management approaches to address intrasystem and especially intersystem mobility is mandatory in MCC environment. Hence, an adaptive protocol suit similar to AdaptNet [50] is essential to alleviate a plethora of heterogeneity-made issues, particularly rate adaptation and congestion control, and provide interoperation among various networks. Integration and interoperation of converged heterogeneous networks in MCC demand lightweight, resource- and cost-effective, sustainable, and user-friendly approaches with optimized performance to address seamless mobility. Realizing such vital need with least signal traffic and latency can significantly enrich quality of cloud-mobile user experience.
- *Context-Processing*: Contemporary mobile devices are capable of gathering extensive context and social information, specifically available cloud resources, network bandwidth, weather conditions, and users' voice and gestures from their surrounding environment [51], [52]. Exponential growth in context and social information (considering environment dynamism and mobility of mobile clients) creates several challenges such as storing, managing, and processing information on resource-constraint smartphones. Although cloud infrastructures are connoted to be a suitable platform for context storage, management, and processing [53], the need for an energy-efficient, reliable, and robust cloud-mobile data migration and communication is unavoidable.
- *Energy Constraint*: Energy is the only unreplenishable resource in mobile devices that cannot be restored spontaneously and requires external resources to be renewed [54]. Current technologies can increase battery capacity by only 5% per annum [55]. Several energy harvesting efforts have been in progress since the 1990s to replenish energy from external resources like human movement [56] and wireless radiation [57], but these intermittent resources are not available on-demand [58]. Alternatively, application offloading [3] and fidelity adaptation [59] approaches are proposed to conserve local mobile resources, especially energy. However, application offloading is a risky, resource-intensive approach, that needs further research and development to be deployed in real scenarios [6]. Fidelity adaptation solutions compromise quality to conserve local resources which impoverish quality of user experience in MCC. Researchers [10]–[12] endeavoured to mitigate application offloading challenges by exploiting secure, reliable, elastic cloud resources instead of insecure, limited surrogates resources. However, cloud-based application offloading cannot always save energy with current developments and demands further efforts [6], [60], [61]. Therefore, the energy constraint of mobile nodes remains a challenge in MCC.
- *Vendor/data Lock-in*: Variety of cloud servers and mobile

devices beside a silo of varied OSs and applications on one hand and non-uniformity of APIs in the absence of underpinning standards on the other hand, intensify data lock-in problem in MCC that bounds customers to a specific mobile device and cloud provider. For example, if cloud-mobile users utilize Apple iCloud<sup>8</sup> as an automatic cloud-based storage service developed and dedicated to Apple products, no data fetch from non-Apple devices is addressable. Alleviating vendor lock-in problem in MCC is more challenging compared with cloud computing. Vendor lock-in solutions need to be applied in both mobile and cloud sides in an interoperable manner (different mobile nodes can communicate with different cloud vendors) which is excessively challenging.

- *Security and Privacy*: A drastic hike in cyber crimes and Internet threats [62], mandates restricted security provisions for publicly accessible cloud resources, especially storage. At a cursory glance, security is only an issue for requesters, but in a deep view, service providers need to set up tighter security provisions to protect their properties and clients' privacy. Ensuring user privacy and security of varied mobile applications running on different mobile devices that utilize heterogeneous cloud resources are the most challenging tasks in MCC. Security and privacy in MCC become more volatile compare to cloud computing due to the insecure nature of wireless communication medium. Hence, the absence of adaptive, rationalized, and rigorous security arrangements is a potential challenge with catastrophic consequences for service providers and requesters in MCC.
- *Elasticity*: Cloud providers confront situations in which there are more demands than available resources. Adverse impact of cloud-resource unavailability and service interruption for MCC clients is more severe than stationary clients connected to the wall power and fixed network. Frequent suspension of energy-constraint mobile clients due to resource scarcity, not only shrinks usefulness of cloud outsourcing for MCC end-users, but also divests privilege of intensive computation anytime, anywhere from mobile users. Therefore, several challenging tasks (e.g. resource provisioning without service interruption, quick disaster recovery, and high service availability) need to be realized since service unavailability and interruption prolong execution time, increase monitoring overhead, and deplete smartphones' local resources, especially battery. Emerging solutions such as Reservoir [63] can be employed to expand and contract cloud resources on demand with main focus on mobile clients.
- *Cloud Policies for Mobile Users*: Cloud service providers apply certain policies to restrain service quality to a desired level by imposing specific limitations via their intermediate applications like Google App Engine bulk loader<sup>9</sup>. Also, service provisioning, controlling, balancing, and billing are often matched with the requirements of desktop clients rather than mobile users. Considering

the great differences in wired and wireless communications, disregarding mobility and resource limitations of mobile devices in design and maintenance of cloud structures can significantly impact on feasibility of MCC solutions. Metrics such as bandwidth quota and number of API calls per day limit clients and impact on user experience [64]. Hence, it is essential to amend restriction rules and policies to meet MCC users' requirements and realize intense mobile computing on the go.

- *Service Execution and Delivery*: Cloud-mobile users, require an efficient monitoring means to measure and evaluate the quality of service they receive. SLA as a formal contract is employed and negotiated in advance to enforce certain level of quality against a fee. During negotiation phase, the terms of services are defined, while the real-time performance is screening during monitoring phase. However, considering MCC dynamism, several network challenges such as inconsistent bandwidth and packet delivery ratio, delay, jitter, and network blips hamper service delivery, raise ambiguity, and increase dispute between cloud vendors and MCC end-users. Therefore, current static SLA can be fleshed out with more powerful, dynamic representation and monitoring techniques established in heterogeneous wireless environment for cloud-mobile users.
- *MCC Billing*: Client mobility in the rapidly changing environment diverges the cloud billing system in MCC from cloud computing. Designing an appropriate billing system for MCC with dynamic heterogeneous environment requires considering additional parameters compared with cloud computing. Interception latency, jitter, session re-establishment delay, bandwidth capacity, and quality of security are examples of major parameters in designing a MCC billing system.

Designing an appropriate billing system for cloud-mobile users that continuously adapts to the heterogeneity and dynamism of MCC environment requires considering additional parameters compared with cloud computing. Interception latency, jitter, session re-establishment delay, bandwidth capacity, and quality of security are examples of major parameters in designing a MCC billing system.

Review of challenges in MCC highlights that heterogeneity has remarkable impacts on mobile computation offloading, seamless connectivity, long WAN latency, mobility management, and vendor/data lock-in which encumber resource-intensive computing on the go and necessitate in-depth analysis. Also, managing billing systems and SLAs in MCC become more complex considering heterogeneity among varied entities which demand upfront consideration on several parameters. Hence, mitigating the impact of heterogeneity can significantly enhance quality of MCC and extend usability of mobile devices to more resource-intensive computing areas. Due to its vital influence, we comprehensively analyze roots of heterogeneity in the following section.

### III. TAXONOMY OF HETEROGENEITY IN MCC

In this section, we comprehensively study heterogeneity in MCC by analyzing roots and dimensions of heterogeneity.

<sup>8</sup><http://www.apple.com/icloud/>

<sup>9</sup><https://developers.google.com/appengine/docs/python/tools/uploadingdata>

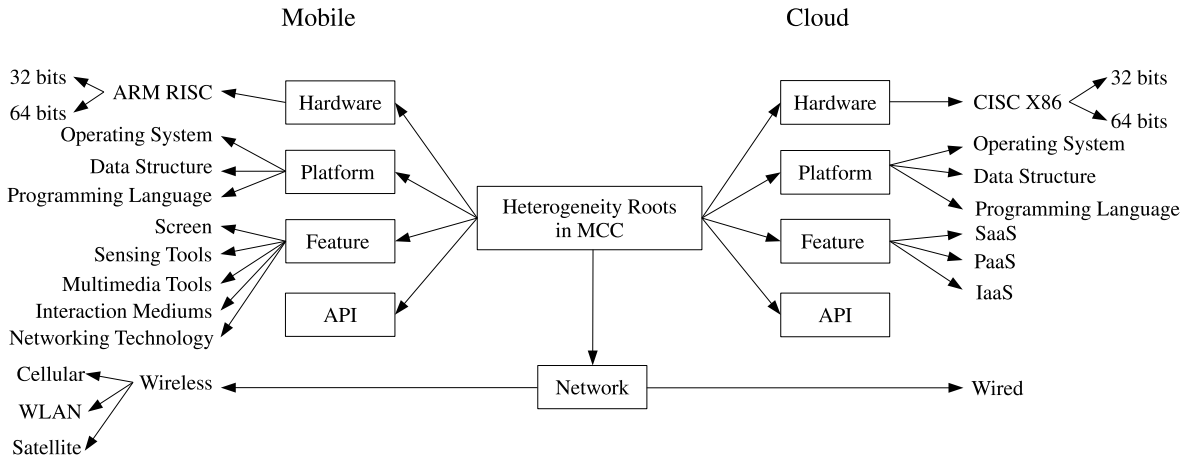


Fig. 3. Taxonomy of Heterogeneity Roots in MCC.

We identify heterogeneity roots as hardware, platform, feature, API, and network and devise a taxonomy depicted in Figure 3. We further scrutinize heterogeneity and realize that heterogeneity in MCC can be classified into two categories, namely *vertical* and *horizontal* that is shown in Figure 6.

#### A. Heterogeneity Roots in MCC

Being heterogeneous is neither a new paradigm nor limited to a specific domain. If we look around we may capture: (i) people speak different languages, but can talk to each other by using a translator (human or tools) or through a common language; (ii) all body parts are intrinsically heterogeneous, but cooperate very concisely and consciously to keep the body fully operational. Similarly, three various components (i.e. mobile devices, clouds, and wireless networks) in MCC should seamlessly cooperate and communicate with each other to benefit mankind, however, some causes make the situation difficult to flourish MCC vision that we examine them as follows.

1) *Hardware Heterogeneity*: Variety of hardware with different inward architecture between mobile devices, cloud servers, and network infrastructures (e.g. access points, radio transceivers, and routers) trigger hardware heterogeneity in MCC.

In the cloud environment, cloud providers maintain different infrastructures and architectural design to enhance quality of their service. Servers use X86 CISC (Complex Instruction Set Computer) architecture with two variations of 32-bit and 64-bit. Moreover, cloud infrastructures gradually grow more heterogeneous due to upgrade and replacement. The emerging growth of cloud computing will increase the number of geographically distributed cloud nodes that intensifies hardware heterogeneity among cloud providers.

The inward architecture and resource specifications of mobile devices such as processor speed, internal memory, radio specification, and battery capacity vary widely among different brands and models. The majority of smartphones are built based on 32-bit ARM RISC (Reduced Instruction Set Computer) processor architecture, but there is a large variation in terms of speed, number of cores, and amount of processor

cache which suits them in specific domains. For instance, among recent Cortex series processors, Cortex-A is ideal for computing intensive multi user applications while Cortex-R is more suitable for real-time data processing scenarios [65]. In the near future, the 64-bit ARM processors are expected to increase the heterogeneity among smartphones [66].

Hardware and architectural heterogeneity among mobile devices and cloud servers hamper direct deployment of cloud resources and services in mobile devices and leads to several problems as below.

- *Imbalanced quality and performance*: Variation in computing resources and their implementations diversify performance and quality of cloud services. Although such variation triggers business competition towards improved services and avoids service monopoly, it negatively affects the business cooperation among cloud providers and complicates user's decision making in choosing the most appropriate vendor among available options. To facilitate the user's decision making, a comprehensive study and comparison among reputed cloud vendors is presented in [67]. Users can compare and contrast the service quality of each vendor from various aspects.
- *Data management and integrity*: The increasing number of very large scale geographically distributed data warehouses and the non-similarity of data structures complicate data management. Integrating huge distributed data and providing virtually unified storage for mobile users is becoming more complicated with the ever increasing heterogeneity in MCC [19].
- *Interoperation*: Data interoperation is the ability of connecting heterogeneous systems (based on wired or wireless), understanding geographical information resources, and exchanging data between/across two or more heterogeneous systems [68]. However, in MCC infrastructure diversity among various clouds on one hand and dissimilarities between cloud and mobile infrastructures with existence of wired against wireless network hardware systems on the other hand, have created data integration and interoperation problems in the absence of interface's standards and uniform platforms. For example, when

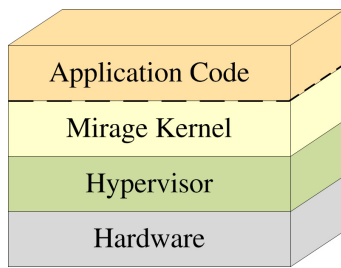


Fig. 4. Mirage Approach with Statistically-linked Kernel and App. [71]

Alice moves from current city to another while utilizing a nearby cloud 'A' via her Android mobile device, remote data (in part or whole) might be migrated to a nearer cloud 'B' for the sake of performance. In this situation, if the cloud 'B' fails to connect to the cloud 'A', if cloud 'B' cannot understand cloud 'A' database information after establishing a connection, or fail to exchange data with cloud 'A', the Alice computing experience will be degraded because of data interoperability problem.

- *Portability*: Codes are not easily movable and executable to/on heterogeneous hosts and the privilege of “write once run anywhere” is divested from developers. For instance, the application written for quad-core processor is not executable on dual-core processor due to architectural and hardware dissimilarities. Similarly, the applications developed for the ARM architecture cannot be executed on X86 without code modification and re-configuration.
- *Accurate energy estimation*: One of the most important aims of MCC is to conserve mobile battery power. Thus, resource-intensive tasks are offloaded from mobile device to the cloud to deliver long-lasting online mobile services. However, before offloading, a decision making system needs to determine whether offloading computation can save energy or not [61]. If the remote execution can save energy, then the offloading is performed, otherwise the application is either terminated or executed locally. However, estimating energy efficiency of offloading is a non-trivial task due to heterogeneity of wireless technologies and infrastructures. Metrics such as power and bit-rate of wireless modems, activating time of interface, activation and deactivation delay of interface, and traffic pattern complicate accurate energy estimation. Also, varied hardware technologies and implementations are different in terms of power consumption [69]. Therefore, precise estimation of required energy for application execution is difficult in different platforms.
- *Cloud-mobile application development*: Developing cross-platform components (i.e. cloud, mobile, and hybrid) for cloud-mobile application is a complicated task. Mobile components should be able to move among various smartphones while cloud components must be portable to all cloud infrastructures. The hybrid components should easily travel among various smartphone and cloud platforms.

To address these problems, Stone et al. [70] propose OpenCL as a parallel programming standard for heterogeneous computing devices. OpenCL enables developers to create

desktop applications executable on various types of computing elements like multicore CPU, GPU, and other accelerators. The same approach can be maintained in the cloud which shrinks hardware heterogeneity. Applications developed for certain architectures using OpenCL can be ported to other architectures with guaranteed correct functionalities. Using multiple programming strategies, the application can query hardware specifications and capabilities of the hosting machine and choose an appropriate kernel to increase correctness and compatibility. However, developing and managing several kernels incur extra cost and encumbrance beside occupying lots of space.

To dilute the impact of hardware variety in the cloud and smartphones, Madhavapeddy et al. [71] propose a cloud OS called Mirage, based on virtualization technology. Mirage runs on top of a hypervisor to produce cross-platform applications that are portable to a plethora of mobile devices and cloud servers. In the Mirage, applications are developed on normal OS like Linux and then compiled into a kernel that is able to run directly on mobile devices and virtual clouds. Figure 4 depicts the layered architecture of Mirage that links the microkernel to an application on top of the hypervisor. Mirage microkernel leverages Xen hypervisor to lessen the impact of architecture heterogeneity of mobile and PCs on mobile applications. However, creating, maintaining, and destroying VM over a smartphone consume local resources and shorten battery life.

Huerta-Canepa and Lee [72] create a virtual cloud computing platform in the absence of a real cloud to augment mobile devices using an ad-hoc network of mobile phones. In this approach impact of hardware heterogeneity is mitigated by leveraging smartphones as remote servers instead of using desktop machines or cloud servers. Similarly, Marinelli [73] propose a MCC platform composed of Android smartphones called Hyrax to lessen the architectural impact of non-ARM (X86) devices. However, using smartphones as cloud providers is hindered by two main challenges. Firstly, giant cloud processing resources are exchanged for limited resources of smartphones. Secondly, the lack of appropriate security and billing mechanisms for individual smartphone owners, discourage the sharing of scarce resources. However, mutual benefits of individual smartphone clients will likely encourage resource sharing in this context.

2) *Platform Heterogeneity*: Platform heterogeneity is the availability of various OSs, programming languages, and data structures in MCC. Currently, a plethora of heterogeneous mobile OSs such as Google's Android and Apple's iOS, each with multiple versions, developed to provide rich, compelling services to end-users. Each platform supports different combinations of programming language and data structures. For instance, Android offers Java language, native code with JNI, and C/C++, while iOS supports Objective-C [17].

In the context of cloud computing, the most widespread cloud providers such as Amazon Web Service (AWS)<sup>10</sup>, Google App Engine<sup>11</sup>, and Microsoft Azure<sup>12</sup> offer different

<sup>10</sup><http://aws.amazon.com>

<sup>11</sup><https://appengine.google.com/>

<sup>12</sup><http://www.windowsazure.com>



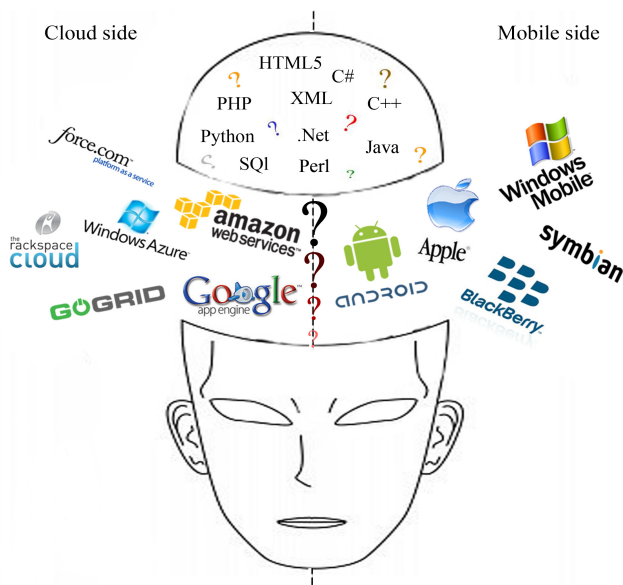


Fig. 5. Platform Heterogeneity in MCC and Challenges for Application Developers.

OSs, programming languages, and data structures. Windows Azure supports a multiple choice of languages like .NET, PHP, Ruby, Python, and Java [74], whereas Google App Engine supports Java and its products plus Ruby and Python. AWS supports most developing languages and offers SDK for Android and iOS smartphones [75]. Azure is built on the SQL, while the App Engine datastore includes GQL, and Amazon's Dynamo system [76] has a different structural design and partitioning scheme. Cloud providers enforce restrictions in databases to provide more flexible services.

Such non-uniformity makes an irksome impediment for cloud-mobile users and application developers. Portability and data integrity problems are also exacerbated in MCC and porting data to different clouds imposes extra monetary and temporal costs. The costs may include downloading data from the current cloud, performing required modification and conversion, and uploading data to the new cloud. Therefore, mobile users, especially corporate mobile users with bulk sensitive data face problems in transferring a huge amount of data between heterogeneous clouds because it is a costly, time consuming, and risky process [77]. Millions of records stored in a cloud database cannot be utilized in another cloud without compromising privacy and integrity when there is a difference between file systems and encryption techniques. Figure 5 depicts heterogeneous platforms and programming technologies in the cloud and mobile domains, and challenges for application programmer in selection of a suitable development environment or a programming language.

Furthermore, application development is a complex task due to platform heterogeneity. Developers need to acquire extensive knowledge of various mobile and cloud programming languages to be able to migrate the application from one platform to another. To mitigate the impact of platform heterogeneity on application portability, several cloud-ready application transition solutions such as Marmalade and PhoneGap are proposed to automatically generate compatible codes for

various platforms. Though these approaches reduce the impact of heterogeneity on the application development process, the effort is not congruous with the definition of portability offered by the InterNational Committee for Information Technology Standards (INCITS) [78] as portability is the capability of transferring one application from one device to a wide range of devices with little or no modification and conversion.

For MCC applications, we need a unified application environment similar to Aneka [79], which allows one to develop cloud applications and deploy them on multiple cloud infrastructures such as AWS, Azure, and GoGrid<sup>13</sup> in a seamless manner. Although Aneka was developed to serve stationary clients, minor extensions and modifications (depending on the needs of client applications) can be performed to leverage Aneka in mobile environments.

3) *Feature Heterogeneity*: Feature heterogeneity in MCC is a result of feature variations in the mobile and cloud domains. Feature heterogeneity in smartphones is due to variation in native features like multimedia, sensing, and interaction tools, visualization area, and networking technologies. For instance, HTC Sensation<sup>14</sup> possesses an 8MP camera while BlackBerry Curve 8520<sup>15</sup> provides 2MP camera. Hence, the development process and performance of applications on various devices are different. For example, the bar code reader application whose functionality depends on the quality of captured image may not offer similar functionality in HTC Sensation and BlackBerry Curve 8520.

Differentiation in sensing apparatuses and interaction features intensify application portability and usability. Recently, sensing tools such as tilting sensors have received noticeable attention from academia and the industry to enhance the quality of interaction in smartphones. In various proposals [80], [81] authors exploit sensing tools such as accelerometer to augment the interaction capabilities of smartphones towards delivering rich user experience. However, feature variation and lack of feature upgrading facility in smartphones obscure usability of feature-dependent applications on various devices.

Moreover, variation in the data visualization area remains a challenge for developers to design and deliver a common user interface for all smartphones. Omitting redundant and less important data according to the preference and context of individual users is an approach in some proposals like event-based semantic image adaptation scheme [82]. Authors identify important objects in an image and collect user preference using a simple feedback mechanism. An adaptation algorithm integrates important objects with user preferences to produce a suitable version for target mobile devices. This approach not only reduces the volume of presentation data by omitting non-important objects, but also offers a superb feeling of customization to mobile users.

Similarly, Chen *et al.* [83] propose a page split method to adapt web content for mobile devices. This approach is implemented inside the mobile browser to adapt the web page content according to the screen size and semantics of the content. However, components like page analysis, content

<sup>13</sup><http://www.gogrid.com/>

<sup>14</sup><http://www.htc.com/us/products/sensation4g-tmobile/>

<sup>15</sup><http://us.blackberry.com/smartphones/blackberrycurve8500/>

detection, page splitting, and index page generation consume a high volume of local resources. Programming level solutions such as screen support APIs in Android 3.2 provides more control to developers to adapt screen content for different Android devices with varied screen sizes [84].

Feature heterogeneity in the cloud domain arises from variations in the services (e.g. infrastructure, platform, software and security as a service) offered by different vendors. For example, Google App Engine (PaaS vendor) and Microsoft Windows Azure (PaaS) provide dissimilar security features; though they offer paid backup storage service, critical data privacy is only offered by Azure [67]. Therefore, users, especially corporate users, face difficulties in moving from one vendor to another.

4) *API Heterogeneity*: Application Programming Interface (API) is an interface supplied by OS vendors or service providers that allows an application written in a high-level language to access specific data or functions from the API distributor. Programmers, including mobile application developers, are usually in a hurry, while mobile users are increasingly demanding a rich computing experience. Therefore, APIs play an important role in delivering a rich experience to mobile users. Mobile platforms such as Android, BlackBerry, and iOS offer a gigantic number of APIs to assist programmers with developing rich mobile applications without direct access to the kernel. However, application portability has become an irksome practice for developers due to inward dissimilarities of APIs. Tarkoma and Lagerspetz [17] study the role of APIs in mobile devices and argue that “The marketplace has a clear need for a common API that unifies network connectivity, energy awareness, and the user experience”.

Similarly, on the cloud side, the majority of cloud providers develop and deploy their own proprietary APIs to describe syntax of specific operations to be utilized by their clients. A drastic growth in the number of cloud providers has created a huge silo of different APIs that intensifies the difficulty of developing applications due to interpreting semantics of data and operations. This outlook, results in API variation intensifying interoperability and portability issues. To mitigate the impact of API heterogeneity on the cloud, several regulatory and research unions endeavour to provide common cloud APIs through, including the European Telecommunications Standards Institute Technical Community (ETSI TC Cloud)<sup>16</sup>, DMTF<sup>17</sup>, and Cloud Audit<sup>18</sup>.

5) *Network Heterogeneity*: The composition of various wireless technologies such as Wi-Fi, 3G, and WiMAX makes MCC more complicated compared to cloud computing. Unlike desktop computers, smartphones utilize wireless communication which is comparatively more intermittent and unreliable while offers lower bandwidth. Client mobility among varied network environments intensifies communication deficiencies and stems complex issues like signal handover. Inappropriate decision making during the handover process like (i) less appropriate selection of network technology among available candidates and (ii) transferring the communication link at the wrong time, increases WAN latency and jitter which directly

degrade the quality of mobile cloud services, especially for delay sensitive content and services [85]. Consequently, quality of cloud-mobile user experience is degraded.

To tackle these challenges, the concept of seamless connectivity among heterogeneous wireless technologies plays a vital role that necessitates reliable intra-system and inter-system handoff schemes [47]. Intra-system handover is a less challenging task due to inward homogeneity of engaging technologies, while addressing inter-system handover is more complicated due to signal transmission difficulties between heterogeneous environments. To realize seamless connectivity across heterogeneous wireless networks, the burgeoning concept of next generation wireless networks [20] with the notion of all IP-based infrastructures is emerging. In the absence of seamless connectivity, the quality of user experience is decreased because of decrements in communication quality and increments in code execution and application response time.

In addition, convergence of wireless and wired networks creates data bottleneck problem. The problem happens when large data streams from a wired network flood the limited bandwidth of wireless networks. This congestion not only increases packet drop ratio and prolongs data transfer between cloud and mobile, but also rises control and maintenance operations which demands enhancement in current network architecture and design for mobile operators. However, network operators are employing powerful helping nodes like wall-connected Wi-Fi hotspots to relay data packets and reduce the intensive load from congested cells. The helping nodes' bandwidth is not infinite because very large bandwidth cannot effectively enhance data trafficking and is limited to a certain upper bound [86]. Hence, a large number of such helping nodes are needed to alleviate ever increasing wireless traffic. Although, MCC imposes overhead on operators' network, strategies like Cisco's innovative Next-Generation Hotspot (NGH) [87] and proposals such as SAMI and MOMCC can reduce data traffic and hike the MNOs' revenue by increasing market share through subscriber retention.

NGH is an advance approach to provide mobile network optimization by offering Wi-Fi as a side mechanism for secure mobile access of data traffic, while aims to enrich the user experience. SAMI is an architecture providing service-based arbitrated multi-tier infrastructure for MCC. The major strengths of this architecture are the existence of SOA-based mobile applications and multi-tier infrastructure layer to leverage infrastructures from three main sources of clouds, MNOs, and MNOs' authorized dealers. On top of the infrastructure layer, an arbitrator layer is designed to classify Services and assign them the suitable resources to them based on several factors such as resource requirement, latency, and security. The solution not only provides an opportunity to employ mobile network operators in sundry supervisory activities, but also hikes the MNOs' revenue, facilitates development and deployment of service-based platform-neutral mobile applications for cloud-mobile application developers, and enhances quality of user experience by minimizing latency. MNOs not only act as a well-established carriers in heterogeneous wireless networks, but also can perform as an adaptor in two different edges; wired (connectible to other clouds like Google) and

<sup>16</sup>[http://www.etsi.org/WebSite/Technologies/GRID\\_CLOUD.aspx](http://www.etsi.org/WebSite/Technologies/GRID_CLOUD.aspx)

<sup>17</sup><http://www.dmtf.org>

<sup>18</sup><http://cloudaudit.org/CloudAudit/Home.html>

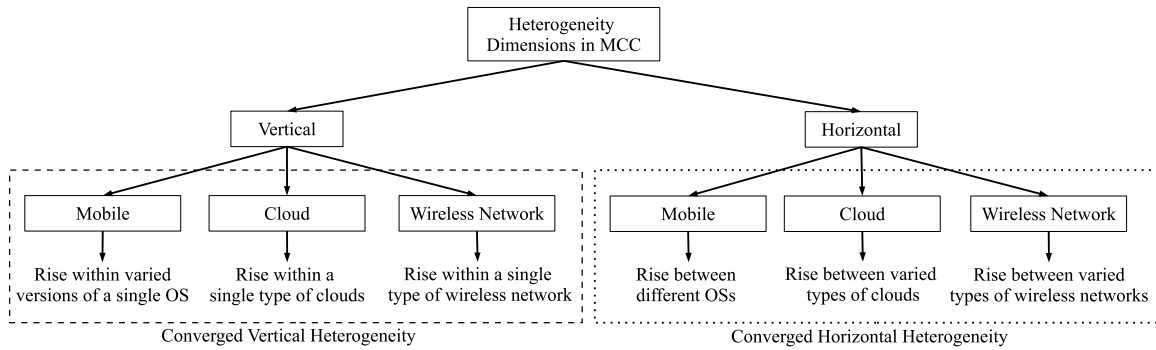


Fig. 6. Heterogeneity Dimensions in MCC.

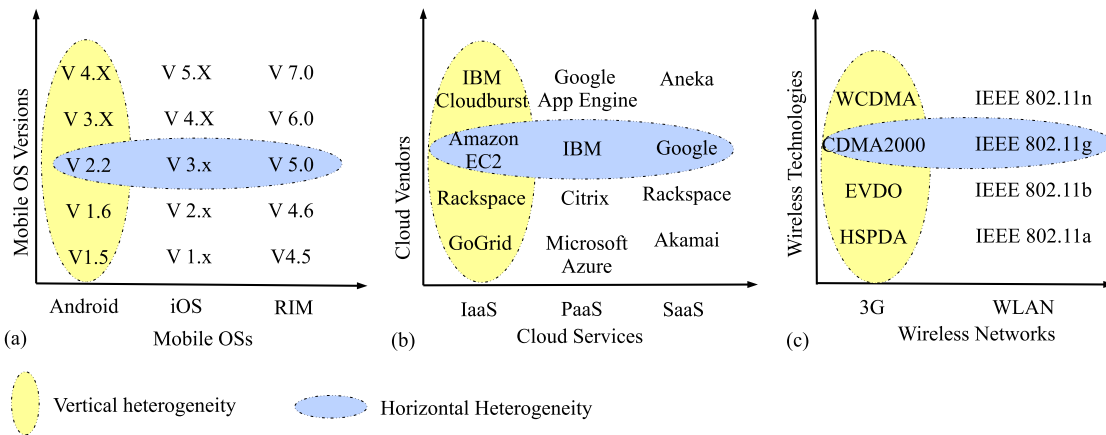


Fig. 7. Vertical and Horizontal Heterogeneity in Three Aspects within MCC: (a) mobile OSs and their versions, (b) cloud services and vendors, and (c) wireless networks and related technologies.

wireless (connectable to mobile devices). Therefore, MNOs can facilitate data interoperability in MCC.

In MOMCC, MNOs are employed as a trusted authority to authorize mobile application programmer, and govern a cloud of decentralized mobile users to enhance computing capabilities of mobile users. Here, MNOs play different roles such as UDDI (Universal Description, Discovery and Integration) and security server authenticating and authorizing varied engaged entities such as mobile service host and mobile end-user. Programmers can develop mobile computational services and store them in MNO servers to be called by MCC users for execution. Although, such operational extension can drastically increase MNOs’ data revenues, but capital investment in infrastructure and architecture is unavoidable.

**B. Heterogeneity dimensions in MCC**

Analysing roots of heterogeneity in MCC results significant differentiation in silo of mobile devices, cloud, and wireless networks. Thus, we describe and taxonomize heterogeneity dimensions in MCC into two categories of *vertical* and *horizontal*. The proposed taxonomy depicted in Figure 6 shows how three underlying MCC components are influenced by two dimensions of heterogeneity. Figure 7 shows three examples of vertical and horizontal heterogeneity in MCC.

1) *Vertical Heterogeneity*: When differentiation is within a single type of mobile OS, cloud service, or wireless network it is named vertical heterogeneity.

- *Mobile Devices*: Among mobile devices, vertical heterogeneity appears within a similar family of products. Different flavors of the OSs offer some unique features and services that are not compatible with other versions. The vertical oval shape in Figure 7(a) shows vertical heterogeneity within different flavors of Android OS. Android 4.0.3 offers social, calendar, and visual voice mail APIs which are totally new compared to Android 3.x [88]. Similarly, in various BlackBerry mobile products, different features and hardware specifications are deployed that prevent the application portability among devices from the same manufacturer. Moreover, deprecating programming elements by OS vendors such as Google<sup>19</sup> and Apple<sup>20</sup>, and discouraging mobile programmers from using such elements further highlights vertical heterogeneity. Although utilizing deprecated methods is currently possible, developers require employing extra mechanisms to ensure compatibility among various OS versions, which cannot satisfy portability principles described by INCITS [78]. Therefore, the application developed for an OS version and deployed in one specific product may not be portable to the same family of products with different versions of the same OS with no or less configuration and modification.

<sup>19</sup><http://developer.android.com/reference/java/lang/Deprecated.html>

<sup>20</sup><http://developer.apple.com/library/ios/#documentation/uikit/reference/UIKit/Class/DeprecationAppendix/AppendixADeprecatedAPI.html>

- *Clouds*: In the cloud, vertical heterogeneity occurs within a single type of clouds that provides similar services, e.g. IaaS (Infrastructure as a Service) or PaaS (Platform as a Service). The vertical oval shape in Figure 7(b) shows vertical heterogeneity within various IaaS service vendors. Though Amazon EC2 and Rackspace are IaaS clouds, they are built on different pillars: internal infrastructures, technologies, and business policies. Therefore, demand for switching between these two cloud services incurs redundant cost, even though both vendors provide IaaS. It also creates data and application portability issues and hinders easy code and data migration within a single type of clouds. Cloud users are forced to adhere to specific cloud service provider(s) [89]. However, standardization efforts like the Open Virtualization Format (OVF) [23] are emerging to alleviate problems and facilitate the deployment of virtual appliances in various clouds.
- *Wireless Networks*: Among different wireless technologies, horizontal handoff is a well-known phenomenon caused due to vertical heterogeneity in mobile wireless environments including MCC. In MCC, horizontal handoff happens in the situations when a cloud-mobile user is moving across heterogeneous access points within a single type of wireless network to access cloud services. For instance, this happens when cloud-mobile user is moving between IEEE 802.11a and IEEE 802.11g WLAN, or between CDMA 2000 and WCDMA 3G technologies. In majority of offloading algorithms in MCC, the network characteristics highly influence on offloading decision [6]. Hence, any change in networking technologies directly impacts on efficiency and effectiveness of the offloading decision and overall process. The vertical oval shape in Figure 7(c) shows vertical heterogeneity within various cellular technologies. Data transmission in cognitive wireless access networks [90] which is configured with a set of different Radio Access Technologies (RATs) and Frequencies (F) is an illustrious example of vertical heterogeneity. When an application or data is tended to change the environment, the transceiver may need to change RAT and F, just change RAT and maintain F, or reverse. Ultimately, the user session is maintained continuously and consistently during mobility by leveraging horizontal handoff process [47].

2) *Horizontal Heterogeneity*: When differentiation is across different types of mobile OSs, cloud services, or wireless networks it is named horizontal heterogeneity.

- *Mobile devices*: Among mobile devices, horizontal heterogeneity appears between different platforms: two or more OSs (e.g. Android and RIM) or brands (e.g. Samsung and Nokia). The horizontal oval shape in Figure 7(a) shows horizontal heterogeneity between different OSs. For instance, the application developed for BlackBerry Torch<sup>21</sup> (RIM V6.0) is not executable in Android V3.x products. Horizontal heterogeneity is usually more challenging compared with the vertical heterogeneity. Portability is exacerbated when development of applications such as

CMH applications is concerned. To develop CMH applications, developers should design the application for the cloud as well as the mobile side. This development process must be repeated for various platforms, which is an exasperating effort for the developer [28]. Developers should consider various mobile platforms, cloud vendors and supporting programming languages to decide how to develop the application which is an irksome impediment.

- *Clouds*: In the cloud, horizontal heterogeneity occurs between different types of clouds that provide heterogeneous services, like IaaS and PaaS. The horizontal oval shape in Figure 7(b) shows horizontal heterogeneity between various types of cloud services. In a scenario that some PaaS vendors offer free limited storage, if a new application utilizes such storage that is incidentally coupled with specific data structure like Google App Engine<sup>22</sup> (the only Google Query Language (GQL)-based PaaS cloud), such dependency locks the application in the cloud. Hence, porting rapidly growing data to an IaaS cloud (for less hosting cost) which is non-GQL-based IaaS (e.g. SQL-based cloud) is hardly possible and inflicts upfront investment.

This type of heterogeneity, similar to the mobile side, is more difficult to address as compared with vertical heterogeneity because of switching difficulties between various service providers with different patterns, architectures, APIs, and business policies.

- *Wireless Networks*: Horizontal heterogeneity in wireless networks occurs when a mobile client is travelling across various networks like cellular and WLAN. Changing network node and supporting mobility, termed “vertical handoff”, highlighting a dilemma in horizontal heterogeneous wireless network, which is a more challenging task with presence of different infrastructure. In this situation, signal handoff processes are more difficult than vertical heterogeneity due to the switching process between different types of network [47]. Unlike the decision making algorithm in vertical handoff that relies on several parameters (like energy efficiency, Received Signal Strength (RSS), accessible bandwidth, security, financial cost, and user preference), decision in horizontal handoff is made based on the RSS only [85]. Such increased complexity obliges vertical handoff optimization toward seamless connectivity in MCC which will enhance on-demand services and increase the quality of user experience. The horizontal oval shape in Figure 7(c) shows horizontal heterogeneity between various types of networks.

#### IV. IMPACTS OF HETEROGENEITY IN MCC

Variations in smartphones, clouds, and networking technologies breed several opportunities and challenges that are discussed in this section.

##### A. Opportunities

The opportunities arising from heterogeneity in MCC are explained as follows:

<sup>21</sup><http://us.blackberry.com/smartphones/blackberrytorch/>

<sup>22</sup><https://developers.google.com/appengine/>

1) *Business opportunity*: One of the positive aspects of heterogeneity in IT, is the business opportunity that creates new jobs (e.g. system analyst, application designer, programmers, and testers) by manufacturing and maintaining multiple versions of a single software for different clients. Market competition is increased and client-centric services are stimulated. If challenges are addressed, IT corporates loss business and many employees become jobless.

2) *Performance Gain*: User perceived performance from online mobile applications is highly influenced by computing performance of server and wireless networks in MCC [91]. While it is financially impossible to avoid infrastructure heterogeneity within a single cloud (due to maintenance, update, and technological advancements), cloud service providers use such heterogeneity as an opportunity to enhance performance and cost of their services. Rosenberg and Chiang [92] study two clusters of computers with identical mean speed. In one cluster, all CPU speeds are similar while in the other one, heterogeneous CPUs are utilized. The authors analyze execution performance of both clusters and mathematically demonstrate that the cluster with heterogeneous CPU speed outperforms another cluster with homogeneous CPUs. Moreover, because varied applications have different architectural preferences, combination of heterogeneous processors with dissimilar architectural features (e.g., pipeline depth, in-order versus out-of-order execution, and superscalar width) can efficiently meet application preferences toward better performance [93] (further explanation is out of the scope of this paper). For instance, Amazon leverages dissimilar processing entities such as Intel Xeon E5507<sup>23</sup> and AMD Opteron 2218HE<sup>24</sup> for creating an instance of VM to fulfill various computing requirements of different mobile applications with least possible cost. Exploiting heterogeneous computing resources in creating VM instances enhances execution performance of online applications in MCC [92]–[94].

Moreover, researchers and industrialists can leverage benefits of heterogeneous communication and networking technologies such as 2G and 3G towards efficient communication. In enterprise organization, running communication-intensive cloud-mobile applications on 3G-ready devices and rest of the applications on 2G devices can reduce capital and operational costs of buying 3G-enabled mobile device and long-lasting batteries [95]. In academia, researchers [96] study execution time and cost of code offloading in MCC. They exploit heterogeneous communication technologies to access remote computation resources located near/far to/from the mobile users and could remarkably enhance application performance and responsiveness by leveraging both 3G and WiFi. The authors propose a multi-tiered cloud infrastructure consist of distant clouds and nearby cloudlets aiming to alleviate effectiveness and efficiency of computation offloading process. They leverage Wi-Fi technology to save energy while communicating with nearby cloudlets and use 3G to access distant giant clouds. The authors report 32% lower delay and 40% cost-reduction in offloading computation-intensive tasks to two-tiered cloud with heterogeneous infrastructure compare to

single-tiered cloud. Thus, heterogeneous co-existence of various communication technologies can originate a performance-energy trade off which can benefit mobile clients, cloud vendors, and network operators. Future mobile communication technologies will likely enable mobile nodes to negotiate with network infrastructures for optimized communication technology. Therefore, heterogeneous selection of infrastructures in mobile-cloud environment provides a higher performance and service range to MCC community.

3) *Enhanced Application Response Time*: Reducing cloud-mobile application response time is one of the most important requirements of MCC which is likely achievable by enhanced performance of leveraging heterogeneous resources while outsourcing computation. Heterogeneous computation hardware in the cloud datacenters, remarkably decreases computation delay in server side resulting less cloud-mobile execution latency and better overall response time. Researchers in [93] explore that leveraging combination of three heterogeneous processor architectures can reduce response time violation by 12X leading to crisp application response which is critical in MCC. Therefore, leveraging heterogeneous computing resources can increase application responsiveness in MCC.

4) *Cost Efficiency*: One of the most critical metrics in successful MCC adoption is the cost of utilizing cloud services by mobile users which includes the amount of native resources, energy, and time used as well as monetary cost of inter-system communication and computation. Efficient match of communication requirements and available wireless technologies, and migrating resource-intensive tasks to high-performance heterogeneous clouds, reduce overall application execution cost. Researchers in [94] observe heterogeneity within computing entities in Amazon EC2 for two period in 2011-2012 and explore that Amazon EC2 leverages various processors in creating a single VM to reduce costs and achieve efficiency. For example, in large VM instances, heterogeneous CPU models, namely Intel Xeon E5507, E5430, E5645, and AMD Opteron 2218HE and 270 are being used. Experimental analysis of CPU, memory, and disk performance of various VM instances resulted different performance outcomes in varied computation loads. The authors conclude that efficient selection of service instance from pool of heterogeneous instances, can deliver up to 30% cost saving to Amazon end-users. Therefore, by assuming rich computing capabilities in cloud and seamless high bandwidth connectivity between mobile and cloud, mobile application responsiveness, local resource consumption, and utilization cost could be enhanced towards more efficiency.

## B. Challenges

Several important challenges resulting from, or intensified by heterogeneity are explained below.

1) *Application Fragmentation*: MCC applications have already been fragmented because of heterogeneity roots explained in section III. For instance, application developed for iPhone cannot run on Android mobile devices. The remedy to this situation is two-folds. Firstly, handling existing heterogeneity; because generating multiple versions of contents for various mobile and cloud platforms is not feasible due

<sup>23</sup><http://ark.intel.com/products/37100?wapkw=e5507>

<sup>24</sup>[http://products.amd.com/\(S\(iecvm0y4bbazo4ujjtjcl55\)\)/pages/OpteronCPUDetail.aspx?id=350&AspxAutoDetectCookieSupport=1](http://products.amd.com/(S(iecvm0y4bbazo4ujjtjcl55))/pages/OpteronCPUDetail.aspx?id=350&AspxAutoDetectCookieSupport=1)

to the monetary and temporal costs. Secondly, decelerating fragmentation growth via standardization and common API. Mobile manufacturers endeavour to enhance user experience by utilizing high-end resources and technologies such as multi-core Samsung Exynos 5 processor<sup>25</sup> and the face unlock feature of Android 4.0<sup>26</sup>. Although these accomplishments enrich computing quality, they intensify heterogeneity and impose extra cost on end-users as well as developers. Porting a platform-dependent application to a new platform levies monetary and temporal cost on the user. Similarly, developing multiple versions of a single application is frustrating and a time consuming practice for developers, that also goes against the “write once, run anywhere” principle. Hence, application portability and interoperability are exacerbated. However, application fragmentation differs in web, online, and native applications.

- *Web applications:* In web applications, data and logic layers are device-agnostic because data storage and code execution take place on the web, whereas presentation layer is naturally device-dependent. Hence, fragmentation in web applications stems from variations in the visualization area and screen rendering capabilities of different mobile devices (excluding feature-dependent components). Several content adaptation approaches [82], [83], [97] are proposed to identify, extract, and adapt essential contents from web pages to the variety of mobile devices. However, successful content adaptation in smartphones needs existing issues (such as functional, structural, and semantical analysis of content, analyzing the screen’s building blocks, and applying user preferences) to be addressed.
- *Online applications:* Similar to web applications, online applications such as cloud-mobile applications [40] and rich mobile applications are distributed according to the client-server concept. Computing and storage intensive components of cloud-mobile applications are likely executed inside cloud resources while native components are executed locally [40]. Hence, fragmentation in on-line applications is more severe than web applications. Separating the screen rendering from the presentation layer is a significant effort [41] in which screen rendering tasks are performed in the cloud where data and logic layers are located. This approach not only conserves local smartphone resources, but also adapts screen contents according to the target device which is deemed to alleviate heterogeneity problems in application fragmentation. In a cross platform cloud-mobile application, cloud components are required to be portable to various cloud vendors, and mobile components should be executable on a multitude of mobile devices.
- *Native applications:* Unlike web and online applications, native applications are monolithic, meaning they are entirely installed and run locally. Hence, developing cross-platform native applications is comparatively the most expensive and time consuming process. Developing

cross-platform application not only requires developers to acquire programming skills in multiple languages, but also inflicts extra temporal and financial costs due to redundant work of porting the application to various platforms. To have a clear understanding of what portability challenge is, we use a mathematical model as follows.

If the development process of a mobile application  $A$  is composed of  $n$  phases (e.g. creation, deployment, and testing) and each phase imposes a differing development cost  $C_i$ , then:

$$C_{DA} = \sum_{i=1}^n C_i \quad (1)$$

where  $C_{DA}$  is the total cost for developing  $A$  for the primary mobile platform and  $C_i$  is the cost of developing phase  $i$ .

To generate a cross-platform application executable on  $m$  new platforms, developers need to transit the application  $A$  to  $m$  new platforms (e.g. Android, iPhone, and BlackBerry) via code regeneration from the scratch (in the worst case) or partial code conversion/modification. Therefore,

$$C_{TA} = \sum_{j=1}^m C_{Tj} \quad (2)$$

where  $C_{TA}$  is the cost of transiting  $A$  to  $m$  destination platforms and  $C_{Tj}$  is the cost of transiting the application to mobile platform  $j$ .

From (1) and (2), we drive:

$$C_{CA} = \sum_{i=1}^n C_i + \sum_{j=1}^m C_{Tj} = C_{DA} + C_{TA} \quad (3)$$

where  $C_{CA}$  is the total cost of developing a cross-platform application for  $m+1$  platforms.

In practice, if the portability problem is addressed, the cost of developing a cross-platform application will be as low as (3) which is the real taste of “write once, run anywhere”. However, according to the definition of portability by INCITS [78], for cross-platform applications,  $\alpha$  is feasible as a little modification or conversion cost. Therefore, we will have:

$$C_{CA} = \sum_{i=1}^n C_i + \alpha = C_{DA} + \alpha \quad (4)$$

According to (4), it is observed that cross-platform solutions omit transition cost for  $m$  platforms if the application is deployable and executable on  $m+1$  platforms. This cost includes coding charges, electricity, maintenance, time, and sundry charges. Furthermore, if developing cloud components is required, the development process becomes excessively complicated and costly. To reduce the burden of building cloud-mobile applications, Manjunatha et al. [28] utilize Domain Specific Language (DSL)

<sup>25</sup><http://www.samsung.com/global/business/semiconductor/minisite/Exynos/index.html>

<sup>26</sup><http://developer.android.com/sdk/android-4.0-highlights.html>

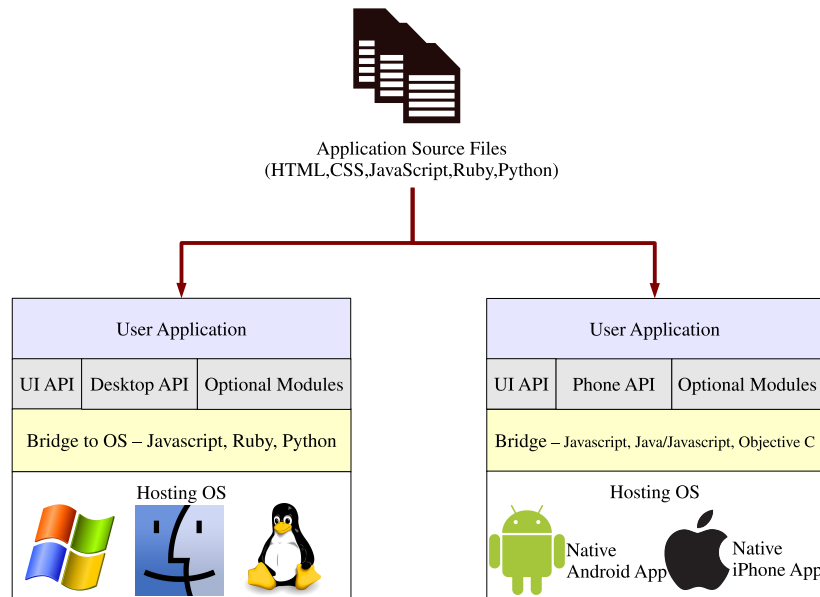


Fig. 8. Titanium Cross-Platform Project [98].

to auto-generate communication interfaces for mobile and cloud target platforms.

Moreover, to alleviate the fragmentation problem, industrial toolkits such as Appcelerator Titanium [98], PhoneGap, and Marmalade aim to auto-generate cross-platform mobile applications. Figure 8 depicts the layered architecture of Titanium. However, these toolkits convert the source code to the destination platforms by creating an abstraction on top of the source code. Also, these tools cannot be a generic solution for developing cross-platform mobile applications, especially for background processing and complex business logics. Because JavaScript API in PhoneGap which exposes native functionality to the browser's JavaScript code is a resource-intensive and slow process compared to the native code [99]. Similarly usability of Titanium Appcelerator is hindered by newly released mobile platforms' APIs. Implementing new APIs in Titanium middleware is a time consuming process while end-users are not patient enough to wait till the API is implemented in Titanium. In general, usability of these transition solutions is hindered by some issues explained below.

- *No dynamic alteration*: Even though cross-platform toolkits are likely suitable for business applications, they are unable to fully alleviate the feature heterogeneity in MCC. Cross-platform tools do not generate customized codes for destination platforms, but convert the source code to a new format. For instance, applications developed on traditional mobile devices cannot utilize special features of contemporary devices like GPS. Similarly, applications which depend on accelerometer functionality are unusable in a device without a tilting sensor.
- *Further enhancements required*: Application transition toolkits require post-transition enhancements (e.g. modification, testing, and maintenance) to fit the application to the target platform [100]. Performing cyclic transition

and post-transition tasks are complex processes due to the diversity of underlying environments which might take weeks for an application to perform [101], [102]. Also, there are many environment variables to be configured based on the new execution environment.

2) *Data Fragmentation*: Successful movement and machine-understandable data across various devices are one of the most irksome impediments of mobile end-users, especially for platform-dependent data such as messages and call history. Storing contact list, messages, and personal data on a mobile device bounds user to the data structure and policies of the mobile platform vendor and locks data in the device. Therefore, transferring data to a device with heterogeneous data structure and policies originates extra temporal and monetary cost for converting data to a format supported by the new platform, due to portability problem in mobile computing. Furthermore, utilizing third party applications for such conversion compromises user privacy and security. Yet, a likelihood of data corruption and mismatch at conversion phase is threatening end-users. High heterogeneity of hardware, platform, and brand among mobile devices complicate data exchange among multitude of mobile devices in the absence of common standards and portability solutions.

To alleviate this problem, utilizing cloud resources and storing personal data on the cloud, similar to the iCloud, becomes prevalent. End-users store data on the cloud to benefit from the anywhere, anytime, any device principal of MCC and avoid data lock-in in their mobile device. However, the storage structure of clouds is fragmented due to the varied architectural designs, supported programming languages, and policies enforced by different cloud vendors [28]. Storing data in the cloud may alleviate data exchange problem between heterogeneous devices in vertically heterogeneous dimension of MCC, but it is more challenging with horizontal heterogeneity when data are migrating across two different mobile OSs

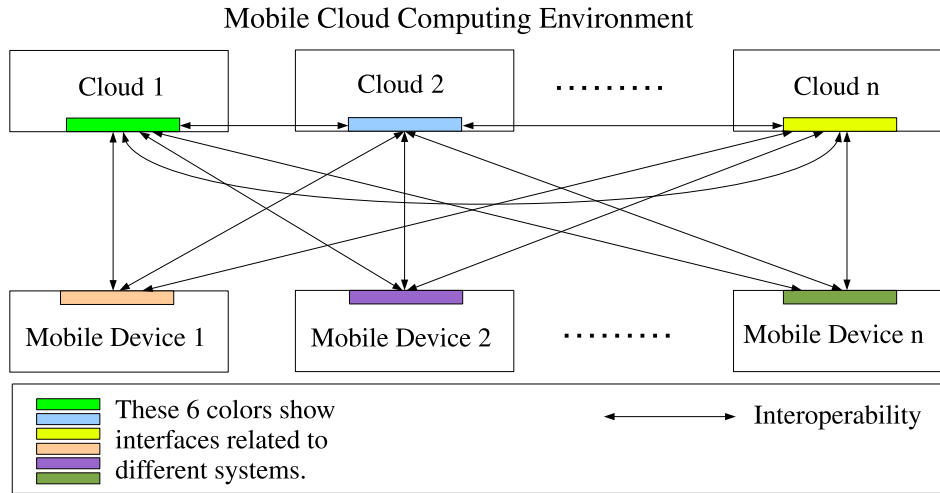


Fig. 9. Interoperability in MCC: Collaboration of Inter-Cloud and Mobile-Cloud Systems with Varied Interfaces Provides Interoperability.

supported by two varied cloud storages based on fragmented data structures.

Furthermore, considering huge data growth in the near future [103], storing whole data in a single warehouse is often impossible. Heterogeneous storage infrastructures not only reduce data integrity and consistency, but also make data management an open cloud issue in MCC [19]. Applying a single access control mechanism for relevant data in various storage environments is another challenging task in a heterogeneous domain such as MCC. Hence, a logical data warehouse is deemed to alleviate the problem by virtually merging the data from multiple warehouses [104].

Despite heterogeneity-originated problems, variations in data structures may offer data security and isolation benefits to the MCC users because of various customized data structures and enforced management policies. Different structural characteristics of cloud storage like architectural schema, cryptography mechanisms, and data compression methods can enhance data security in the cloud.

3) *Interoperability*: API Heterogeneity of mobile and cloud systems beside inward variation in cloud system structure originate interoperability as a major challenge in MCC [43]. The challenge intensifies vendor lock-in problem and obscures data migration and code transition across a multitude of existing processing units. In MCC, providing collaboration among various mobile and cloud processing units with different interfaces is a non-trivial matter. Figure 9 depicts mobile-cloud interoperability across a silo of computing devices and illustrates the inter-cloud collaboration between different cloud providers as essential requirements in MCC.

4) *Portability*: A lack of standards, technologies, and solutions to handle heterogeneity in MCC implicitly creates the portability challenge. In the cloud, providers offer various computing services with different structures and programming languages. Similarly, smartphone vendors develop various approaches and technologies to enhance the quality of their products. Thus, porting an application to varied devices in the fragmented MCC domain becomes more challenging.

Ideally, data and application should be able to cross a multitude of clouds and smartphones with no or little configuration

and conversion [78]. However, it is almost impractical to port native codes to the cloud and transfer resource-intensive codes from clouds to the weak smartphones. Therefore, we limit portability in MCC to the ability of (i) migrating cloud components from one cloud to other clouds, (ii) migrating mobile components from one smartphone to other smartphones, and (iii) migrating data across heterogeneous clouds and smartphones. Figure 10 depicts portability in MCC.

5) *Developing cost*: Developing one application for several platforms is a costly process that demands knowledge of several programming languages and enforces redundant design and programming tasks. In the presence of heterogeneity, the market share for a single version of an application is diminished to a fraction of total customers that decrease revenue and demotivates individual programmers.

6) *Time lag*: In many scenarios, although the financial cost of developing a multi-platform application is affordable and welcomed due to its business opportunity, prolonged application development process procrastinates service offering to a wider community and leads to financial loss.

7) *Application maintenance*: When there are multiple versions of a single application, each for a different platform, if one application is enhanced or changed, the modifications should be reflected to all versions. Usually a single development model cannot be employed and extended to different platforms, hence, different modification approaches should be undertaken for each platform which is a hectic job for developers.

8) *Communication*: The heterogeneity of wireless technologies in MCC created communication problems like signal handover. Bandwidth variation between wireless and wired technologies is the source of data bottlenecks when huge amounts of data stream from wired (cloud servers) into a wireless medium (mobile devices). Hence, continuous, consistent connectivity and scalable, accessible networking services are necessary to enhance the quality of mobility and communication between a wide range of mobile devices and clouds.

9) *Security and privacy*: Although MCC can enhance the quality of mobile computing and increase the usability of smartphones, lack of strict, robust supervision over the wire-



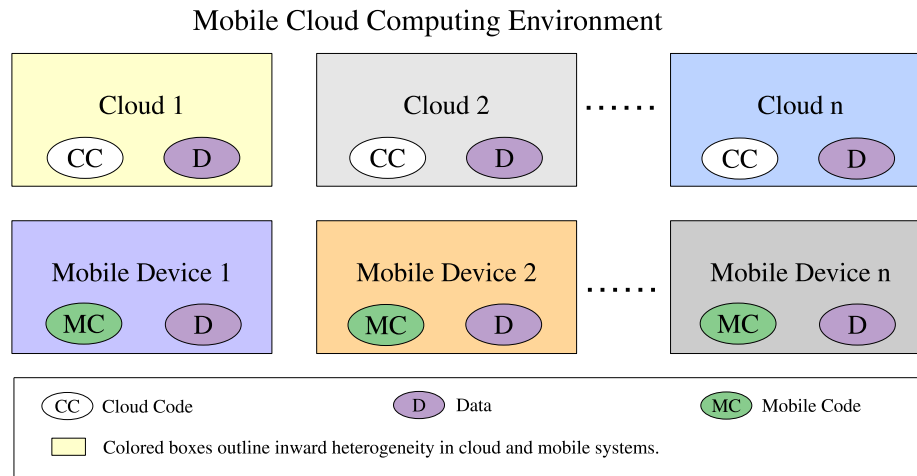


Fig. 10. Portability in MCC: Data should be portable to all cloud and mobile devices. Cloud codes should move between clouds while mobile codes should move between a multitude of mobile devices regardless of the inward heterogeneity of hosting machines.

less communications and cloud resources utilization [62] significantly abate trust among cloud users with cloud attachment. Storing confidential information (e.g. banking information, medical records, and social security numbers) in cloud infrastructures and remote access to them via the Internet and wireless mediums threaten mobile users in the presence of numerous hackers. Therefore, to increase trust among cloud-mobile users and a secure collaboration process between different cloud service providers and consumers, strong authentication, authorization, and communication protection are required. For example, data migration from one cloud to another (serially) or operation across multiple clouds (simultaneously) should be secured by cloud providers. During communication processes, personal information and personally identifiable information require protection by cloud providers, mobile network operators, and trusted third parties. Identity provisioning and access management through different environments are a sample of the security keys which manifest the necessity of secure intercommunication in MCC.

## V. HETEROGENEITY HANDLING APPROACHES

To avail maximum benefits from heterogeneity and overcome its challenges, researchers have leveraged several technologies and approaches like virtualization, middleware, and SOA which are referred to as heterogeneity handling approaches. This Section presents a survey of generic solutions to handle heterogeneity in mobiles, clouds, and networks.

### A. Virtualization Technology

Virtualization [105] is one of the cornerstone technologies of MCC promising to reduce the negative impacts of hardware, feature, and platform heterogeneity. Using the virtualization approach, a VM manager (hypervisor) is deployed on top of a cloud, mobile, or both to host desired platform(s) in order to create a homogeneous execution environment between various smartphones and clouds. Mirage is an example effort that exploits hypervisor in both the cloud and smartphone to build cross-platform applications regardless of variations in underlying devices. However, in prior work [106], we showed that

VM deployment and management impose excessive overhead on resource-constrained mobile devices.

Moreover, recent application offloading approaches like [10], [11] leverage virtualization technology to offload mobile application (entirely or partially) to remote cloud resources for execution. Similarly, efforts like [41] exploit virtualization technology to separate screen rendering tasks from the presentation layer and migrate them to the cloud. On top of a VM inside a cloud, a remote server can render screen-related processing tasks and send the result to the device. The authors aim to address the feature heterogeneity (screen size) of smartphones using a virtual screen rendering approach. However, virtualization gives rise to several security threats such as VM hopping and VM escape [107]. VM hopping is a virtualization threat through which an attacker can exploit a VM and attack other VM(s) on the same host. VM escapes can violate the security of VMs when an attacker accesses control over the hypervisor. Therefore, several open challenges such as VM deployment and management [106] and security establishment [108], [109] should be alleviated before virtualization technology can be fully established as grounding technology in MCC.

### B. Middleware/Adapter

Adapter is an intermediate tool or approach to smooth out the impact of heterogeneity in a specific domain like software engineering and distributed computing environments. A 'plug adapter' is a well-known example that helps an international traveler to overcome the incompatibility problem caused by dissimilarity of power outlets in different countries. In software engineering, adapter is a pattern designed to address the common problems caused by heterogeneity in many situations [110]. Distributed computing middleware such as Object Request Broker (ORB) acts as an arbiter to enable communication between heterogeneous object systems regardless of their inward differences.

Emergent middleware as Blair and Grace [111] describe, is the contemporary approach to tackle problems caused by extreme heterogeneity. The idea is advocated by enormous

accomplishments in academia and industry. Emerging efforts such as [112]–[117] in academy and commercial products such as Oracle Fusion Middleware 11g<sup>27</sup> and Open Middleware Adapter (OMA)<sup>28</sup> advocate suitability of adapters and more specifically middleware technology to tackle the heterogeneity-made problems. However, contemporary middleware require great deal of research and development to become an appropriate heterogeneity handling technique in MCC.

### C. Standardization

Standardization is a widespread process to create outward homogenization of heterogeneous entities to enhance quality of service while stimulate mutual business opportunities among participants. Standardization is often enforced when a new technology, product, or approach like cloud computing or MCC emerges. Maturing cloud computing standards, not only prevent vendor lock-in problem (by establishing interoperability across cloud providers) and ensure quality, reliability, and safety of services, but also facilitate trade in IT businesses and computing shift to the cloud. Novice vendors can promote their services by obtaining validity and accreditation from reputed third-party standard organizations such as NIST, DMTF, Open Grid Forum<sup>29</sup>, ETSI, and Cloud Security Alliance<sup>30</sup> toward increasing the level of trust among potential customers. OVF is an open DMTF standard aims to enable VMs migration among different cloud platforms. Open cloud computing interface (OCCI)<sup>31</sup> is an API to realize integration, portability, and interoperability among various clouds. Cloud Data Management Interface (CDMI) creates a functional interface with main focus on data management (create, retrieve, update, and delete) in the cloud to discover clouds' inward specifications and manage stored data accordingly.

In the context of MCC, futuristic standards can alleviate impact of heterogeneity by ensuring desirable service characteristics like compatibility, interoperability, portability, efficiency, reliability, and safety among heterogeneous computing elements and technologies.

### D. Service Oriented Architecture (SOA)

Service Oriented Architecture (SOA) is a well-known design philosophy independent from specific technology, vendors, and business policies that incorporates different services towards generating complex applications and services. Web services are well-known SOA implementation models that could successfully integrate heterogeneous services from various service providers like Facebook, Google, and Yahoo<sup>32</sup> and enable interoperability across them. For instance, Facebook delivers multimedia YouTube<sup>33</sup> content regardless of inward differentiations.

Service Oriented Computing (SOC) [118] is a service-driven approach to generate service-based applications with least dependency to the specific platform. In service-based applications, functions are defined, implemented, and combined as services to enhance application granularity and modularity, flexibility, scalability, and reusability. This approach encourages the development of elastic applications meaning that users are able to extend and shrink functionality on-demand which is parallel to the vision of cloud computing. Aneka [119] is an example of service-oriented solution to automatically manage distributed resources (clouds and grids). Aneka is designed to be robust against variations in application models, security solutions, and communication protocols such that client choices can be applied at any time without affecting the existing system.

In summary, loosely coupled SOA-driven services have a proactive potential to integrate heterogeneous resources in MCC. Loutas et al. [27] observe interoperability issues between cloud systems and perceive the ability of SOA plus semantics for a new cloud landscape. They present an architecture for heterogeneous clouds, called RASIC, to enable semantic interoperability among clouds. The authors propose a user-centric paradigm to facilitate developing and deploying of SOA-based services in a large-scale, resource-intensive environment hosted by different cloud providers. It is concluded that utilizing SOA-based design philosophy towards a common API standard for cloud can eliminate vendor lock-in problem, ease content migration across heterogeneous clouds, and reduce the cost of porting data and application from one cloud to another.

### E. Semantic Technology

Semantic Technology is a distinctive research field for bridging and interconnecting heterogeneous data, applications, and processes by encoding meanings and providing abstraction layers [120]. It generates new data from massive pools of heterogeneous information by training codes from data, file content, and application codes. Semantic systems based on logic and knowledge can surpass human deductive abilities. Also with a strong ability to create a semantic link between mined data, the machine can present logical results from big data warehouses. Hence, semantic technology is potentially appropriate to be deployed in MCC and alleviate the impact of variations in converged domains of smartphones and clouds.

In the syntax-less and semantic-full environment of MCC, semantic data can be combined to mobile and cloud contexts to overcome heterogeneity challenges. Semantic Technology in MCC has the potential to provide interoperability, data portability, scalability, flexibility, and context-awareness across heterogeneous environments and systems. Sheth and Ranabahu [89], [121] discuss portability and interoperability problems in the context of cloud computing and propose a solution by adding semantics to the cloud in order to overcome three major problems of functionality definition, data modeling, and service description enhancement. The authors also extend their work and propose a semantic-centric programming paradigm by identifying four types of semantics for an application, namely system, non-functional, data, and logic & process

<sup>27</sup><http://www.oracle.com/us/products/middleware/index.html>

<sup>28</sup><http://ultra-ats.com/products/open-middleware-adapter-oma-software-development-kit-sdk/>

<sup>29</sup><http://www.ogf.org>

<sup>30</sup><https://cloudsecurityalliance.org/>

<sup>31</sup><http://occi-wg.org/>

<sup>32</sup><http://yahoo.com>

<sup>33</sup><http://youtube.com>

[77]. This effort is envisioned to facilitate interoperability and portability issues that are manifested as a vendor lock-in problem in the cloud. However, resource-intensive processing and reasoning transactions beside difficulties to acquire and integrate huge, reliable, and accurate semantic data [122], [123] impede deployment of semantic technology in MCC.

## VI. OPEN ISSUES

This section presents some of the research directions in MCC, especially those which are more complex due to the heterogeneity. Addressing these open issues is vital to alleviating the restrictions caused by heterogeneity.

**Architectural issues:** A reference architecture for heterogeneous MCC environment is a crucial requirement for unleashing the power of mobile computing towards unrestricted ubiquitous computing. Employing a unified or a joint architecture composed of varied architectures requires further studies. A generic architecture might be a little optimistic when market competition enforces business policies for mobile manufacturers and cloud providers. However, it is achievable by leveraging technology-neutral design approaches such as SOA. Several research communities like NIST [124], HP [125], and IBM [126] endeavour to address the open challenges of cloud computing by proposing conceptual reference architectures. In the absence of such reference architecture for MCC, ubiquity of mobile computing is diminished. Several open challenges can be alleviated in the presence of the reference architecture to release the power of mobile devices.

**Context-awareness issues:** Context-aware and socially-aware computing [52] are inseparable traits of contemporary handheld computers. To achieve the vision of mobile computing among heterogeneous converged networks and computing devices, designing resource-efficient environment-aware applications is an essential need. Rapidly changing execution environment necessitates continuous context gathering and analysis which heavily consume native resources of mobile devices and increase execution latency. Context gathering is a well investigated area [127], but management, processing, and interfering huge volumes of context in a rapidly changing mobile ecosystem remain an open issue.

Convergence of cloud computing, mobile computing, and varied network technologies notifies integration of context-aware and socially-aware functionalities for designing an integrated architecture to support seamless mobility and smart computing in MCC. Autonomic computing and networking with context-aware and socially-aware systems can underpin ubiquitous applications, but standard interfaces have a pivotal role for their success.

**Live VM migration issues:** Executing resource-intensive mobile application via VM migration-based application offloading involves encapsulation of application in VM instance and migrating it to the cloud, which is a challenging task due to additional overhead of deploying and managing VM on mobile devices [128]. Additionally, it becomes more challenging when a mobile user changes its location to a place far from the offloaded contents (code or data). Prolonged distance raises latency and degrades mobile user-observed application

performance that necessitates live migration of VM [129] across geographically distributed cloud datacenters which is a non-trivial task considering user mobility, wireless communications, and heterogeneity among various cloud vendors and network providers. Hence, effective solutions like reactive and proactive VM migration (proactive approach needs predicting user destination) to a place near the mobile user without service interruption becomes essential to avoid user experience degradation. Efforts similar to VMware vMotion [130] and [131] in cloud computing are necessary to optimize VM migration overhead in MCC. Reducing computation complexity and overhead, energy, data volume, and communication cost, are critical in low-latency, low-cost migration of VM related with mobile devices.

**Mobile communication congestion issues:** Mobile data traffic is tremendously hiking by ever increasing mobile user demands for exploiting cloud resources which impact on MNOs. Data storage/retrieval, application offloading, and live video streaming are example of cloud-mobile operations that drastically increase traffic, leads to excessive congestion and packet loss. Furthermore, employing MCC in several domains such as VANET, wireless sensor networks, and M2M communications can further increase data volume across the network. Hence, managing such huge data becomes challenging, especially when offloading mobile data are distributed among helping nodes to commute to/from the cloud.

Although solutions such as utilizing and sharing heterogeneous wireless spectrum [132] and leveraging regional hotspots as helping nodes (to relay traffic at peak hours) can contribute to smooth traffic, several decisions need to be made like how efficient is relaying offloading data packets? How secure are the helping nodes and in what extend security of shifted MCC data would be protected? How latency caused by data migration to other node impacts on interactive applications' responsiveness? Such kinds of questions, not only necessitate intelligent systems to manage offloaded MCC data, but also might alter systems' overall architecture and network structure. Because, sharp MCC data hike, necessitates cost-effective, efficient deployment of infrastructures (e.g. hotspots) and innovative strategies with least overhead and latency. Therefore, a cognitive system within MCC that likely pre-identifies congestion issues and considers factors like MCC application types (e.g. data-intensive, computation-intensive, and communication-intensive) to determine the best action(s) to relay traffic, is imperative as a future research direction.

**Trust, security, and privacy issues:** Trust is an essential factor for the success of the burgeoning MCC paradigm [24]. Constructing a trustable, secure environment is an open issue which is exacerbated when the Internet is utilized as the bridge between front-end and back-end devices (over wireless and wired networks). Provisioning security and providing data integrity and reliability beside delivering essential services (e.g. always on connectivity and cloud services) over the heterogeneous distributed systems, wireless networks, and the Internet require novel lightweight methods. Trust establishment based on the service provider's reputation (i.e. cloud, mobile, and Internet provider) and aggregation of trust from each service node would be a valuable approach that requires future research.

Privacy is exclusively a big issue in the vast convergence of several network technologies, which is exacerbated when cloud users trust the cloud providers and store sensitive information on public data warehouses. Hence, absorbing user trust is an important criterion leading to yet another challenge, that of how cloud service providers can ensure confidentiality of user information.

## VII. CONCLUSIONS

Recently, cloud computing has created a new research impetus in smartphone augmentation leading to the emergence of mobile cloud computing paradigm. The ultimate goal of MCC is to provide rich mobile computing through seamless communication between front-users (cloud-mobile users) and end-users (cloud providers) regardless of heterogeneous, wireless environments and underlying platforms in global roaming. MCC landscape is providing higher magnitude values than mobile computing since it can release the power of mobile devices by stimulating virtual heavy functionality and bulk data storage in mobile devices. Similarly, it will face greater challenges because of converged heterogeneous domains of mobile and cloud computing. Since the emergence of cloud computing, several cloud providers have come up with different strategies, infrastructures, and platforms. Such differentiation impedes data extraction from various clouds and executing applications on inhomogeneous clusters or servers. These problems are exacerbated by wireless networks and fragmented mobile devices. Several R&D groups are concentrating on common APIs and a unified architecture for MCC to smooth out heterogeneities and enable interoperability within and across the mobile and cloud environments.

This paper presented an overview of MCC, identified and analyzed roots and dimensions of heterogeneity in MCC. Also, some of the major MCC challenges are described based on literature. It was argued that MCC is a more heterogeneous domain compared to cloud computing due to divergent computing (mobile computing and cloud computing) paradigms and networking technologies. The taxonomy of heterogeneity roots in MCC was also devised. We analyzed and classified the pivotal roots of heterogeneity and related approaches that handle certain classes of heterogeneity. According to the types of heterogeneity in each landscape, we categorized heterogeneity of cloud computing, mobile computing, and wireless networks into two classes, namely vertical and horizontal. It is noteworthy that data and application fragmentation stemming from fundamental diversities adds monetary and temporal cost to the application generation process. Moreover, dissimilar platform performances (i.e. CPU performance in mobile devices as well as cloud servers) beside differing power consumption and bit-rate of heterogeneous wireless technologies can affect the overall performance of remote processing approaches and mechanisms. The survey advocated that although there are several academic and industrial solutions, there is no suitable ground yet that can cover the highlighted challenges for end-users, application developers, mobile cloud service providers, and third parties. The absence of early standards and lack of advanced technologies decelerate MCC growth and ubiquitous computing. Providing loosely coupled cloud services and semantically interoperable platforms across the MCC domain,

besides enhancing virtualization techniques, has a clear need for research advancements.

Finally, research directions pertaining to the burgeoning field of convergent computing and networking opened new research and innovation opportunities. Developing a reference architecture and establishing trust are the most prominent issues that demand unified network connectivity and standards.

## REFERENCES

- [1] C. Albanesius. (2011, Feb.) "smartphone shipments surpass PC shipments for first time. what's next?". PC Magazine. [Online]. Available: <http://www.pcmag.com/article2/0,2817,2379665,00.asp>
- [2] (2009) "Global markets for smartphones and PDAs". [Online]. Available: <http://www.bccresearch.com/pressroom/report/code/IFT068A>
- [3] M. Satyanarayanan, "Pervasive computing: vision and challenges," *IEEE Pers. Commun.*, vol. 8, no. 4, pp. 10–17, 2001.
- [4] J. Flinn, S. Park, and M. Satyanarayanan, "Balancing performance, energy, and quality in pervasive computing," in *Proc. IEEE 22nd International Conference on Distributed Computing Systems (ICDCS'02)*, Vienna, Austria, Jul. 2002, pp. 217–226.
- [5] M. D. Kristensen, "Scavenger: Transparent development of efficient cyber foraging applications," in *Proc. IEEE Eighth Annual International Conference on Pervasive Computing and Communications (PerCom)*, Mannheim, Germany, Apr. 2010, pp. 217–226.
- [6] M. Sharifi, S. Kafaie, and O. Kashefi, "A survey and taxonomy of cyber foraging of mobile devices," *IEEE Commun. Surveys & Tutorials*, vol. PP, no. 99, pp. 1–12, 2011.
- [7] R. Buyya, C. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation Computer Systems*, vol. 25, no. 6, pp. 599–616, 2009.
- [8] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011. [Online]. Available: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- [9] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
- [10] E. Cuervo, A. Balasubramanian, D. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: making smartphones last longer with code offload," in *Proc. ACM 8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys'10)*, San Francisco, CA, USA, Jun. 2010, pp. 49–62.
- [11] B. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "CloneCloud: Elastic execution between mobile device and cloud," in *Proc. ACM The European Professional Society on Computer Systems (EuroSys'11)*, Salzburg, Austria, Apr. 2011, pp. 301–314.
- [12] X. W. Zhang, A. Kunjithapatham, S. Jeong, and S. Gibbs, "Towards an elastic application model for augmenting the computing capabilities of mobile devices with cloud computing," *Mobile Networks & Applications*, vol. 16, no. 3, pp. 270–284, 2011.
- [13] "Mobile cloud applications," ABI Research. [Online]. Available: <http://www.abiresearch.com/research/1003385>
- [14] D. Milojicic and R. Wolski, "Eucalyptus: Delivering a private cloud," *IEEE Computer*, vol. 44, no. 4, pp. 102–104, 2011.
- [15] D. Macedo, A. dos Santos, and G. Pujolle, "From TCP/IP to convergent networks: challenges and taxonomy," *IEEE Commun. Surveys & Tutorials*, vol. 10, no. 4, pp. 40–55, 2008.
- [16] S. Abolfazli, Z. Sanaei, A. Gani, and R. Buyya, "Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges," *IEEE Commun. Surveys & Tutorials*[Submitted], 2013.
- [17] S. Tarkoma and E. Lagerspetz, "Archiving over the mobile computing chasm: Platforms and runtimes," *IEEE Computer*, vol. 44, no. 4, pp. 22–28, 2011.
- [18] D. DURKEE, "Why cloud computing will never be free," *Communication of the ACM*, vol. 53, no. 5, pp. 62–69, 2010.
- [19] S. Sakr, A. Liu, D. Batista, and M. Alomari, "A survey of large scale data management approaches in cloud environments," *IEEE Commun. Surveys & Tutorials*, vol. 13, no. 3, pp. 311–336, 2011.
- [20] I. F. Akyildiz, X. Jiang, and S. Mohanty, "A survey of mobility management in next-generation all-ip-based wireless systems," *IEEE Wireless Commun.*, vol. 11, no. 4, pp. 16–28, 2004.
- [21] "The standard for network APIs," GSM Association's OneAPI. [Online]. Available: <http://oneapi.gsma.com/>

- [22] "An open community leading cloud standards," Open Cloud Computing Interface. [Online]. Available: <http://occi-wg.org/>
- [23] (2009) "Open virtualization format specification (OVF)". [Online]. Available: [http://www.dmtf.org/sites/default/files/standards/documents/DSP0243\\_1.0.0.pdf](http://www.dmtf.org/sites/default/files/standards/documents/DSP0243_1.0.0.pdf)
- [24] Z. Sanaei, S. Abolfazli, A. Gani, and R. H. Khokhar, "Tripod of requirements in horizontal heterogeneous mobile cloud computing," in *Proc. 1st International Conference on Computing, Information Systems, and Communications (CISCO'12)*, Singapore, May 2012.
- [25] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for VM-Based cloudlets in mobile computing," *IEEE Pervasive Computing*, vol. 8, no. 4, pp. 14–23, 2009.
- [26] A. Madhavapeddy. (2010) "Mirage, a cloud operating system". [Online]. Available: <http://www.openmirage.org>
- [27] N. Loutas, V. Peristeras, T. Bouras, E. Kamateri, D. Zeginis, and K. Tarabanis, "Towards a reference architecture for semantically interoperable clouds," in *Proc. IEEE 2nd International Conference on Cloud Computing Technology and Science (CloudCom'10)*, Dayton, OH, USA, Dec. 2010, pp. 143–150.
- [28] A. Manjunatha, A. Ranabahu, A. Sheth, and K. Thirunarayan, "Power of clouds in your pocket: An efficient approach for cloud mobile hybrid application development," in *Proc. IEEE 2nd International Conference on Cloud Computing Technology and Science (CloudCom'10)*, Dayton, OH, USA, Dec. 2010, pp. 496–503.
- [29] "Marmalade." [Online]. Available: <http://madewithmarmalade.com/>
- [30] "Phonogap." [Online]. Available: <http://Phonogap.com>
- [31] H. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless Communications and Mobile Computing*, 2011.
- [32] N. Fernando, S. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2012.
- [33] T. Imielinski and H. Korth, *Introduction to Mobile Computing*, ser. The Springer International Series in Engineering and Computer Science, 1996, vol. 353, pp. 1–43.
- [34] A. Talukdar, H. Ahmed, and R. Yavagal, *Mobile Computing, 2E*. Tata McGraw-Hill Education, 2010.
- [35] M. Weiser, "The computer for the 21st century," *IEEE Pervasive Computing*, vol. 99, no. 1, pp. 19–25, 2002 Reprinted from 1991.
- [36] D. Saha and A. Mukherjee, "Pervasive computing: a paradigm for the 21st century," *IEEE Computer*, vol. 36, no. 3, pp. 25–31, 2003.
- [37] M. Satyanarayanan, "Mobile computing: the next decade," *ACM SIG-MOBILE Mobile Computing and Communications Review*, vol. 15, no. 2, pp. 2–10, 2011.
- [38] S. Abolfazli, Z. Sanaei, and A. Gani, "Mobile cloud computing: A review on smartphone augmentation approaches," in *Proc. 1st International Conference on Computing, Information Systems, and Communications (CISCO12)*, Singapore, May 2012.
- [39] B.-G. Chun and P. Maniatis, "Augmented smartphone applications through clone cloud execution," in *Proc. 12th Conference on Hot Topics in Operating Systems (HotOS'09)*, 2009, p. 8.
- [40] V. March, Y. Gu, E. Leonardi, G. Goh, M. Kirchberg, and B. S. Lee, "μcloud: Towards a new paradigm of rich mobile applications," *Procedia Computer Science*, vol. 5, pp. 618–624, 2011.
- [41] Y. Lu, S. P. Li, and H. F. Shen, "Virtualized screen: A third element for cloud-mobile convergence," *IEEE Multimedia*, vol. 18, no. 2, pp. 4–11, 2011.
- [42] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, "Cuckoo: a computation offloading framework for smartphones," in *Proc. 2nd International Conference on Mobile Computing, Applications, and Services, (MobiCASE '10)*, Santa Clara, CA, USA, Oct. 2010.
- [43] M. Hogan, F. Liu, A. Sokol, and J. Tong, "NIST cloud computing standards roadmap-version 1.0," Tech. Rep., July 2011.
- [44] M. Shiraz, A. Gani, R. Hafeez, and R. Buyya, "A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing," *IEEE Commun. Surveys & Tutorials*, vol. Accepted, November 2012.
- [45] Z. Sanaei, S. Abolfazli, A. Gani, and M. Shiraz, "SAMI: Service-based arbitrated multi-tier infrastructure for mobile cloud computing," in *Proc. IEEE 1st International Conference on Communications in China Workshops (ICCC)*, Beijing, China, Aug. 2012, pp. 14–19.
- [46] S. Abolfazli, Z. Sanaei, M. Shiraz, and A. Gani, "MOMCC: Market-Oriented Mobile Cloud Computing based on Service Oriented Architecture," in *Proc. IEEE 1st International Conference on Communications in China Workshops (ICCC)*, Beijing, China, Aug. 2012, pp. 8–13.
- [47] N. Nasser, A. Hasswa, and H. Hassanein, "Handoffs in fourth generation heterogeneous networks," *IEEE Commun. Mag.*, vol. 44, no. 10, pp. 96–103, 2006.
- [48] A. Miettinen and J. Nurminen, "Energy efficiency of mobile clients in cloud computing," in *Proc. 2nd USENIX Conference on Hot Topics in Cloud Computing*, Boston, MA, Jun. 2010, pp. 4–11.
- [49] H. Lagar-Cavilla, N. Tolia, E. de Lara, M. Satyanarayanan, and D. O'Hallaron, "Interactive resource-intensive applications made easy," in *Middleware*, ser. Lecture Notes in Computer Science, vol. 4834. Springer Berlin / Heidelberg, 2007, pp. 143–163.
- [50] I. Akyildiz, Y. Altunbasak, F. Fekri, and R. Sivakumar, "Adaptnet: an adaptive protocol suite for the next-generation wireless internet," *IEEE Commun. Mag.*, vol. 42, no. 3, pp. 128–136, 2004.
- [51] N. D. Lane, E. Miluzzo, L. Hong, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Commun. Mag.*, vol. 48, no. 9, pp. 140–150, 2010.
- [52] P. Lukowicz, S. Pentland, and A. Ferscha, "From context awareness to socially aware computing," *IEEE Pervasive Computing*, vol. 11, no. 1, pp. 32–41, Jan 2012.
- [53] J. L. Jiang, Y. W. Wu, X. M. Huang, G. W. Yang, and W. M. Zheng, "Online video playing on smartphones: A context-aware approach based on cloud computing," *J. Internet Technology*, vol. 11, no. 6, pp. 821–827, 2010.
- [54] M. Satyanarayanan, "Avoiding dead batteries," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 2–3, 2005.
- [55] S. Robinson. (2009) "Cellphone energy gap: Desperately seeking solutions". [Online]. Available: <http://www.strategyanalytics.com/default.aspx?mod=reportabstractviewer&a0=4645>
- [56] J. Flinn and M. Satyanarayanan, "Energy-aware adaptation for mobile applications," in *Proc. ACM 17th Symposium on Operating Systems Principles (SOSP'99)*, SC, USA, Dec. 1999, pp. 48–63.
- [57] S. R. Avro. (2009) "Wireless electricity is real and can change the world". [Online]. Available: [www.consumerenergyreport.com/2009/01/15/wireless-electricity-is-real-and-can-change-the-world/](http://www.consumerenergyreport.com/2009/01/15/wireless-electricity-is-real-and-can-change-the-world/)
- [58] W. F. Pickard and D. Abbott, "Addressing the intermittency challenge: Massive energy storage in a sustainable future [scanning the issue]," *Proc. IEEE*, vol. 100, no. 2, pp. 317–321, 2012.
- [59] B. Rajesh Krishna, "Powerful change part 2: reducing the power demands of mobile devices," *IEEE Pervasive Computing*, vol. 3, no. 2, pp. 71–73, 2004.
- [60] K. Kumar, J. Liu, Y. Lu, and B. Bhargava, "A survey of computation offloading for mobile systems," *Mobile Networks and Applications*, pp. 1–12, 2012.
- [61] K. Kumar and Y. H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *IEEE Computer*, vol. 43, no. 4, pp. 51–56, 2010.
- [62] C. Cachin and M. Schunter, "A cloud you can trust," *IEEE Spectrum*, vol. 48, no. 12, pp. 28–51, 2011.
- [63] B. Rochwerger, D. Breitgand, A. Epstein, D. Hadas, I. Loy, K. Nagin, J. Tordsson, C. Ragusa, M. Villari, S. Clayman, E. Levy, A. Maraschini, P. Massonet, H. Munoz, and G. Toffetti, "Reservoir-when one cloud is not enough," *IEEE Computer*, vol. 44, no. 3, pp. 44–51, 2011.
- [64] "Google app engine: Quotas." [Online]. Available: <https://developers.google.com/appengine/docs/quotas>
- [65] (2009) "The ARM cortex-A9 processors". White Paper. [Online]. Available: <http://arm.com/files/pdf/ARMcortexA-9Processors.pdf>
- [66] (2011, Oct.) "ARM discloses technical details of the next version of the ARM architecture". SANTA CLARA, CA, USA. [Online]. Available: <http://goo.gl/98kBc>
- [67] "Find the best: Compare cloud computing providers." [Online]. Available: <http://cloud-computing.findthebest.com>
- [68] G. Blair, M. Paolucci, P. Grace, and N. Georgantas, "Interoperability in complex distributed systems," in *Formal Methods for Eternal Networked Software Systems*, ser. Lecture Notes in Computer Science, M. Bernardo and V. Issarny, Eds. Springer Berlin Heidelberg, 2011, vol. 6659, pp. 1–26.
- [69] G. P. Perrucci, F. H. P. Fitzek, and J. Widmer, "Survey on energy consumption entities on the smartphone platform," in *Proc. IEEE 73rd Vehicular Technology Conference (VTC Spring)*, Budapest, Hungary, May 2011, pp. 1–6.
- [70] J. Stone, D. Gohara, and G. Shi, "OpenCL: A parallel programming standard for heterogeneous computing systems," *Computing in science & engineering*, vol. 12, no. 3, p. 66, 2010.
- [71] A. Madhavapeddy, R. Mortier, J. Crowcroft, and S. Hand, "Multiscale not multicore: Efficient heterogeneous cloud computing," in *Proc. ACM-BCS International Conference on Visions of Computer Science (ACM-BCS '10)*, Edinburgh, UK, Apr. 2010, pp. 1–12.

- [72] G. Huerta-Canepa and D. Lee, "A virtual cloud computing provider for mobile devices," in *Proc. ACM 1st Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond (MCS'10)*, San Francisco, California, Jun. 2010, pp. 1–5.
- [73] E. E. Marinelli, "Hyrax: Cloud computing on mobile devices using MapReduce," Master's thesis, School of Computer Science, Carnegie Mellon University, 2009.
- [74] "Windows Azure Developer Center," 2012. [Online]. Available: <http://www.windowsazure.com/en-us/develop/overview>
- [75] "Amazon Web Services: Mobile developer center." [Online]. Available: <http://aws.amazon.com/mobile>
- [76] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo: amazon's highly available key-value store," in *Proc. ACM 21st Symposium on Operating Systems Principles (SOSP'07)*, Stevenson, Washington, USA, Oct. 2007, pp. 205–220.
- [77] A. Ranabahu and A. Sheth, "Semantics centric solutions for application and data portability in cloud computing," in *Proc. IEEE 2nd International Conference on Cloud Computing Technology and Science (CloudCom'10)*, Dayton, OH, USA, Nov. 2010, pp. 234 – 241.
- [78] INCITS, "Portability definition," InterNational Committee for Information Technology Standards. [Online]. Available: [www.incits.org/ansdit/p2.htm](http://www.incits.org/ansdit/p2.htm)
- [79] C. Vecchiola, X. Chu, and R. Buyya, "Aneka: A software platform for .NET-based cloud computing," in *High Speed and Large Scale Scientific Computing*, vol. W. Gentsch, L. Grandinetti, G. Joubert (Eds.), ISBN: 978-1-60750-073-5, pp. 267–295, 2009.
- [80] S. Valberg and P. T. Christensen, "Improving usability of mobile devices by means of accelerometers," Master Thesis, 2009.
- [81] E. Jones, J. Alexander, A. Andreou, P. Irani, and S. Subramanian, "Gestext: accelerometer-based gestural text-entry systems," in *Proc. ACM 28th International Conference of Human Factors in Computing Systems (CHI'10)*, Atlanta, GA, USA, Apr. 2010, pp. 2173–2182.
- [82] W. Yin, J. Luo, and C. Chen, "Event-based semantic image adaptation for user-centric mobile display devices," *IEEE Trans. Multimedia*, vol. 13, no. 3, pp. 432–442, 2011.
- [83] Y. Chen, X. Xie, W. Ma, and H. Zhang, "Adapting web pages for small-screen devices," *IEEE Internet Computing*, vol. 9, no. 1, pp. 50–56, 2005.
- [84] "Android 3.2." [Online]. Available: <http://developer.android.com/sdk/android-3.2.html#api>
- [85] X. Yan, Y. Ahmet Sekercioglu, and S. Narayanan, "A survey of vertical handover decision algorithms in fourth generation heterogeneous wireless networks," *Computer Networks*, vol. 54, no. 11, pp. 1848–1863, 2010.
- [86] P. Li and Y. Fang, "On the throughput capacity of heterogeneous wireless networks," *IEEE Trans. Mobile Computing*, vol. 11, pp. 2073–2086, 2012.
- [87] (2012) "The future of hotspots: Making Wi-Fi as secure and easy to use as cellular." [Online]. Available: <http://www.cisco.com/en/US/solutions/collateral/ns341/ns524/ns673/white%5Fpaper%5Fc11-649337.html>
- [88] "Android API levels." [Online]. Available: <http://developer.android.com/guide/appendix/api-levels.html>
- [89] A. Sheth and A. Ranabahu, "Semantic modeling for cloud computing, part 1," *IEEE Internet Computing*, vol. 14, no. 3, pp. 81–83, 2010.
- [90] P. Demestichas, V. Stavroulaki, D. Boscovic, A. Lee, and J. Strassner, "m@ANGEL: autonomic management platform for seamless cognitive connectivity to the mobile internet," *IEEE Commun. Mag.*, vol. 44, no. 6, pp. 118–127, 2006.
- [91] J. Huang, Q. Xu, B. Tiwana, Z. M. Mao, M. Zhang, and P. Bahl, "Anatomizing application performance differences on smartphones," in *Proc. ACM 8th international conference on Mobile systems, applications, and services (MobiSys'10)*, San Francisco, CA, USA, Jun. 2010, pp. 165–178.
- [92] A. Rosenberg and R. Chiang, "Toward understanding heterogeneity in computing," in *Proc. IEEE International Symposium on Parallel & Distributed Processing (IPDPS'10)*, Fort Collins, CO, USA, Apr. 2010, pp. 1–10.
- [93] B. L. M. Guevara, B. Lubin, "Navigating heterogeneous processors with market mechanisms," in *Proc. IEEE 19th International Symposium on High-Performance Computer Architecture (HPCA)*, Shenzhen, China,, Feb. 2013.
- [94] Z. Ou, H. Zhuang, J. K. Nurminen, A. Ylä-Jääski, and P. Hui, "Exploiting Hardware Heterogeneity within the same Instance Type of Amazon EC2," in *Proc. 4th USENIX Conference on Hot Topics in Cloud Computing (HotCloud'12)*, Boston, MA, Jun. 2012, p. 4.
- [95] Motorola. (2008) "2G and 3G cellular networks: Their impact on todays enterprise mobility solutions and future mobility strategies". White Paper. [Online]. Available: <http://www.motorola.com/web/Business/Products/Mobile%20Computers/Handheld%20Computers/%5FDocuments/staticfile/3G%5FWhitepaper%5F0608.pdf>
- [96] M. R. Rahimi, N. Venkatasubramanian, S. Mehrotra, and A. V. Vasilakos, "MAPCloud: Mobile applications on an elastic and scalable 2-tier cloud architecture," in *Proc. IEEE/ACM 5th International Conference on Utility and Cloud Computing (UCC'12)*, Chicago, Illinois, USA, Nov. 2012, pp. 83–90.
- [97] X. Yunpeng, T. Yang, and L. Qian, "Mashup-Based Web Page Adaptation for Small Screen Mobile Device," in *Proc. IEEE 5th International Conference on Wireless Communications, Networking and Mobile Computing (WiCom'09)*, Beijing, China, 2009, pp. 1–4.
- [98] "Appcelerator." [Online]. Available: <http://www.appcelerator.com>
- [99] A. Lacenski. (2012) "phonegap plugins". Adobe-Nitobi. [Online]. Available: <http://wiki.phonegap.com/w/page/36752779/PhoneGap%20Plugins#OverallArchitecture>
- [100] S. Allen, V. Graupera, and L. Lundrigan, *Pro Smartphone cross-platform development: iPhone, BlackBerry, Windows Mobile and Android development and distribution*. Springer, 2010.
- [101] D. Dern. (2010, Jun.) "cross-platform smartphone apps still difficult". [Online]. Available: <http://spectrum.ieee.org/geek-life/tools-toys/crossplatform-smartphone-apps-still-difficult>
- [102] P. Wayner. (2009) "The cross-platform option: Web apps for smartphones". [Online]. Available: <http://www.infoworld.com/d/developer-world/cross-platform-option-web-apps-smartphones-485>
- [103] J. F. Gantz, C. Chute, A. Manfrediz, S. Minton, D. Reinsel, W. Schlichting, and A. Toncheva. (2008, Mar.) "The diverse and exploding digital universe: An updated forecast of worldwide information growth through 2011". White Paper. International Data Corporation. [Online]. Available: <http://www.emc.com/collateral/analyst-reports/diverse-exploding-digital-universe.pdf>
- [104] Gartner. (2011, Oct.) "Gartner identifies the top 10 strategic technologies for 2012". [Online]. Available: <http://www.gartner.com/newsroom/id/1826214>
- [105] B. Golden, *Virtualization For Dummies*. Wiley, 2008.
- [106] M. Shiraz and A. Gani, "Mobile cloud computing: Critical analysis of application deployment in virtual machines," in *Proc. Int'l Conf. Information and Computer Networks (ICIN'12)*, vol. 27, 2012.
- [107] K. Owens. (2009) "Securing virtual compute infrastructure in the cloud". Whitepaper. SavvisCorp. [Online]. Available: <http://www.savvis.com/en-us/info%5Fcenter/documents/hos-whitepaper-securingvirtualcomputeinfrastructureinthecloud.pdf>
- [108] H. Takabi, J. Joshi, and G. Ahn, "Security and privacy challenges in cloud computing environments," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 24–31, 2010.
- [109] Y. Chen, V. Paxson, and R. H. Katz, "Whats new about cloud computing security?" EECS Department, University of California, Berkeley, Tech. Rep. UCB/EECS-2010-5, Jan 2010. [Online]. Available: <http://www.eecs.berkeley.edu/Pubs/TechRpts/2010/EECS-2010-5.html>
- [110] P. Wolfgang, *Design patterns for object-oriented software development*. Addison-Wesley, 1994.
- [111] G. Blair and P. Grace, "Emergent middleware: Tackling the interoperability problem," *IEEE Internet Computing*, vol. 16, no. 1, pp. 78 –82, 2012.
- [112] N. Chandrakant, A. Bijil, P. Shenoy, K. Venugopal, and L. Patnaik, "Middleware service oriented rescue and crime information system (RCIS) using heterogeneous fixed nodes in WSNs," in *Proc. International Conference on Advanced Computing, Networking and Security (ADCON'11)*, Surathkal, India, Dec. 2012, pp. 389–398.
- [113] E. Andriescu, R. Speicys Cardoso, and V. Issarny, "AmbiStream: a middleware for multimedia streaming on heterogeneous mobile devices," in *Proc. ACM/IFIP/USENIX 12th International Conference on Middleware (Middleware'11)*, Lisboa, Portugal, Dec. 2011, pp. 249–268.
- [114] P. Gupta, N. Zeldovich, and S. Madden, "A trigger-based middleware cache for ORMs," in *Proc. ACM/IFIP/USENIX 12th International Conference on Middleware (Middleware'11)*, Lisboa, Portugal, Dec. 2011, pp. 329–349.
- [115] S. Walraven, E. Truyen, and W. Joosen, "A middleware layer for flexible and cost-efficient multi-tenant applications," in *Proc. ACM/IFIP/USENIX 12th International Conference on Middleware (Middleware'11)*, Lisboa, Portugal, Dec. 2011, pp. 370–389.
- [116] Y. Bromberg, P. Grace, L. Réveillère, and G. Blair, "Bridging the interoperability gap: Overcoming combined application and middleware heterogeneity," in *Proc. ACM/IFIP/USENIX 12th International*

- Conference on Middleware (Middleware'11)*, Lisboa, Portugal, Dec. 2011, pp. 390–409.
- [117] J. Rellermeyer and R. K pfer, "Co-managing software and hardware modules through the juggle middleware," in *Proc. ACM/IFIP/USENIX 12th International Conference on Middleware (Middleware'11)*, Lisboa, Portugal, Dec. 2011, pp. 431–450.
- [118] M. Huhns and M. Singh, "Service-oriented computing: Key concepts and principles," *IEEE Internet Computing*, vol. 9, no. 1, pp. 75–81, 2005.
- [119] R. N. Calheiros, C. Vecchiola, D. Karunamoorthy, and R. Buyya, "The Aneka Platform and QoS-Driven Resource Provisioning for Elastic Applications on Hybrid Clouds," *Future Generation Computer Systems*, vol. 28, no. 6, pp. 861–870, June 2012.
- [120] A. Sheth and C. Ramakrishnan, "Semantic (web) technology in action: Ontology driven information systems for search, integration and analysis," *IEEE Data Engineering Bulletin*, vol. 26, pp. 40–48, 2003.
- [121] A. Sheth and A. Ranabahu, "Semantic modeling for cloud computing, part 2," *IEEE Internet Computing*, vol. 14, no. 4, pp. 81–84, 2010.
- [122] M. Sabou, "Smart objects: Challenges for semantic web research," *Semantic Web*, vol. 1, no. 1, pp. 127–130, 2010.
- [123] O. Corcho and R. Garc a-Castro, "Five challenges for the semantic sensor web," *Semantic Web*, vol. 1, no. 1, pp. 121–125, 2010.
- [124] L.-J. Zhang and Q. Zhou, "CCOA: Cloud computing open architecture," in *Proc. IEEE 7th International Conference on Web Services (ICWS'09)*, Los Angeles, CA, USA, Jul. 2009, pp. 607–616.
- [125] R. Bohn, J. Messina, F. Liu, J. Tong, and J. Mao, "NIST cloud computing: Reference architecture," in *Proc. IEEE 7th World Congress on Services (SERVICES'11)*, Washington DC, USA, Jul. 2011, pp. 594–596.
- [126] M. Behrendt, B. Glasner, P. Kopp, R. Dieckmann, G. Breiter, S. Papp , H. Kreger, and A. Arsanjani, "Cloud computing reference architecture v2.0," IBM, Tech. Rep., Feb. 2011. [Online]. Available: [https://s3-sa-east-1.amazonaws.com/bucketmanoelveras/manoel\\_veras\\_PROFESSIONAL/CLOUDCOMPUTING/Artigos/Artigos\\_IBM/arq\\_ibm.pdf](https://s3-sa-east-1.amazonaws.com/bucketmanoelveras/manoel_veras_PROFESSIONAL/CLOUDCOMPUTING/Artigos/Artigos_IBM/arq_ibm.pdf)
- [127] J. Hong, E. Suh, and S. Kim, "Context-aware systems: A literature review and classification," *Expert Systems with Applications*, vol. 36, no. 4, pp. 8509–8522, 2009.
- [128] M. Shiraz, S. Abolfazli, Z. Sanaei, and A. Gani, "A study on virtual machine deployment for application outsourcing in mobile cloud computing," *The J. Supercomputing*, vol. 63, pp. 946–964, Mar. 2013.
- [129] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Wared, "Live migration of virtual machines," in *Proc. 2nd Symposium on Networked Systems Design & Implementation (NSDI'05)*, vol. 2, Boston, MA, May 2005, pp. 273–286.
- [130] "Virtual machine migration comparison: VMware vSphere VS. Microsoft Hyper-V," VMware, Tech. Rep., Oct. 2011. [Online]. Available: <http://www.vmware.com/files/pdf/vmw-vmotion-verus-live-migration.pdf>
- [131] H. Liu, H. Jin, C.-Z. Xu, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in *Proc. ACM 20th International Symposium on High Performance Distributed Computing (HPDC'11)*, San Jose, California, Jun. 2011, pp. 171–182.
- [132] A. Min, X. Zhang, J. Choi, and K. Shin, "Exploiting spectrum heterogeneity in dynamic spectrum market," *IEEE Trans. Mobile Computing*, vol. 11, pp. 2020–2032, 2012.



**Zohreh Sanaei** is currently a Ph.D. candidate and research assistant in High Impact Research Project (Mobile Cloud Computing: Device and Connectivity) fully funded by Malaysian Ministry of Higher Education in the Department of Computer Systems and Technology at the University of Malaya, Malaysia. She received her M.Sc. in Information Systems in 2008 from India and BE (Software Engineering) in 2001 from Iran. She worked in 3MCD and EIC, Iran as a network engineer and participated in several wireless communication projects from 2001 till 2006. She has been working for more than 6 years as a part-time lecturer for the ministry of social affairs, Iran as a technical and vocational trainer. Her main research interests include mobile cloud computing, distributed computing, and ubiquitous computing. She is a member of IEEE society and can be corresponded via [zsanaeim@gmail.com](mailto:zsanaeim@gmail.com) or [sanaei@ieee.org](mailto:sanaei@ieee.org). For further information, please visit her cyberhome at [www.mobilecloudfamily.com](http://www.mobilecloudfamily.com)



**Saeid Abolfazli** is currently a Ph.D. candidate, research assistant in High Impact Research Project (Mobile Cloud Computing: Device and Connectivity) fully funded by Malaysian Ministry of Higher Education, and part time lecturer in the Department of Computer Systems and Technology at the University of Malaya, Malaysia. He received his M.Sc in Information Systems in 2008 from India and BE (Software Engineering) in 2001 from Iran. He has been serving as CEO of Espanta Information Complex during 1999–2006 in Iran. He also was part time

lecturer to the ministry of education and Khorasan Technical and Vocational Organization between 2000 and 2006. He is a member of IEEE society and IEEE CS Cloud Computing STC. He has been serving as a reviewer for several international conference and ISI journals of computer science. His main research interests include Mobile Cloud Computing, lightweight protocols, and service oriented computing (SOC). Please write to him at [abolfazli.s@gmail.com](mailto:abolfazli.s@gmail.com) or [abolfazli@ieee.org](mailto:abolfazli@ieee.org). For further information, please visit his cyberhome at [www.mobilecloudfamily.com](http://www.mobilecloudfamily.com)



**Dr. Abdullah Gani** is Associate Professor of Computer System and Technology at the University of Malaya, Malaysia. His academic qualifications were obtained from UK's universities - bachelor and master degrees from the University of Hull, and Ph.D from the University of Sheffield. He has vast teaching experience due to having worked in various educational institutions locally and abroad - schools, teaching college, ministry of education, and universities. His interest in research started in 1983 when he was chosen to attend Scientific Research course in RECSAM by the Ministry of Education, Malaysia. More than 100 academic papers have been published in conferences and respectable journals. He actively supervises many students at all level of study - Bachelor, Master and PhD. His interest of research includes self-organized systems, reinforcement learning, and wireless-related networks. He is now working on mobile cloud computing with High Impact Research Grant for the period of 2011–2016.



**Dr. Rajkumar Buyya** is Professor of Computer Science and Software Engineering, Future Fellow of the Australian Research Council, and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University, commercializing its innovations in Cloud Computing. He has authored over 425 publications and four text books including “Mastering Cloud Computing” published by McGraw Hill and Else-

vier/Morgan Kaufmann, 2013 for Indian and international markets respectively. He also edited several books including “Cloud Computing: Principles and Paradigms” (Wiley Press, USA, Feb 2011). He is one of the highly cited authors in computer science and software engineering worldwide (h-index=70, g-index=144, 23000+ citations). Microsoft Academic Search Index ranked Dr. Buyya as the world’s top author in distributed and parallel computing between 2007 and 2012. Recently, ISI has identified him as a “Highly Cited

Researcher” based on citations to his journal papers.

Software technologies for Grid and Cloud computing developed under Dr. Buyya’s leadership have gained rapid acceptance and are in use at several academic institutions and commercial enterprises in 40 countries around the world. Dr. Buyya has led the establishment and development of key community activities, including serving as foundation Chair of the IEEE Technical Committee on Scalable Computing and five IEEE/ACM conferences. These contributions and international research leadership of Dr. Buyya are recognized through the award of “2009 IEEE Medal for Excellence in Scalable Computing” from the IEEE Computer Society, USA. Manjrasoft’s Aneka Cloud technology developed under his leadership has received “2010 Asia Pacific Frost & Sullivan New Product Innovation Award” and “2011 Telstra Innovation Challenge, People’s Choice Award”. He is currently serving as the foundation Editor-in-Chief (EiC) of IEEE Transactions on Cloud Computing. For further information on Dr. Buyya, please visit his cyberhome: [www.buyya.com](http://www.buyya.com)