# MOBILE CLOUD COMPUTING: THE STATE-OF-THE-ART, CHALLENGES, AND FUTURE RESEARCH

SAEID ABOLFAZLI*[A], ZOHREH SANAEI[A], MOHAMMAD HADI SANAEI[A], MOHAMMAD SHOJAFAR[B], ABDULLAH GANI[A]

CENTER FOR MOBILE CLOUD COMPUTING RESEARCH, UNIVERSITY OF MALAYA, MALAYSIA[A]

DEPARTMENT OF INFORMATION ENGINEERING ELECTRONICS AND TELECOMMUNICATIONS, UNIVERSITY SAPIENZA OF ROME, ITALY[B]

E-MAIL: ABOLFAZLI, SANAEI@IEEE.ORG, MH.SANAEI@STU-MAIL.UM.AC.IR, SHOJAFAR@DIET.UNIROMA1.IT, ABDULLAH@UM.EDU.MY

**Keywords:** Mobile Cloud Computing, cloud-based mobile augmentation, wireless communications, smartphone, resource-intensive mobile application, offloading.

**Summary:** Mobile Cloud Computing (MCC) is the state-of-the-art distributed mobile computing technology that is recently gaining ground to augment computational capabilities of resource-constraint mobile devices using cloud-based resources. MCC employs Cloud-based Mobile Augmentation (CMA) approaches to efficiently execute resource-intensive components of mobile applications outside the device in a resource-rich cloud-based computing resource(s). However, employing CMA approaches is not a straightforward panacea and is encumbered by varied challenges, particularly long WAN latency. Researchers endeavored to mitigate the impacts of long WAN (Wide Area Network) latency by proposing varied architectures for MCC. In this chapter, we present comprehensive overview of MCC, present its definition, motivation, and taxonomy of MCC building blocks followed by brief comparison of cloud computing and MCC. Various MCC architectures are illustrated and scrutinized. Also several open challenges that require future research are presented.

## 1. INTRODUCTION

The popularity and availability of mobile devices, especially smartphones are creating tense dependency that people do not leave home without them. However, smartphones' miniature nature, lightness, and mobility traits impose severe limitations on their processing capabilities, battery lifetime, storage capacity, and visualization power (e.g., screen size and rendering capability) impeding execution of resource-intensive computations and bulky data storage on smartphones (Abolfazli, Sanaei, Gani Abdullah, et al. 2013). Resource-intensive mobile applications are mobile applications whose execution requires large CPU (Central Processing Unit) transactions per second, big RAM (Random Access Memory) to load the code and data, extensive disk storage to store contents, and long lasting battery which are not available in today's mobile devices. Enterprise systems, 3-dimensional games, and speech recognition software are examples of such resource-intensive applications.

To alleviate mobile device shortcomings, researchers propose frameworks to perform resource-intensive computations outside the mobile devices inside cloud-based resources, that breeds Mobile Cloud Computing (MCC) (Sanaei, Abolfazli, Gani, et al. 2013). MCC infrastructures include multitude of heterogeneous mobile devices, cloud-based resources (individual/corporate computing devices that inherit cloud computing technologies and principles), and networking infrastructures that are managed via software systems named Cloud-based Mobile Augmentation (CMA) (Abolfazli, Sanaei, Ahmed, et al. 2013). The augmented mobile device can execute intensive computations that would not be done, otherwise. Thus, the mobile application programmers do not consider mobile devices deficiencies while programming applications and users will not entertain device limitations in employing intensive applications.

CMA approaches can overcome the resource deficiencies of mobile devices and realize execution of "three main categories of applications, namely (i) computing-intensive software such as speech recognition and natural language processing, (ii) data-intensive programs such as enterprise applications, and (iii) communication-intensive applications such as online video streaming applications" (Abolfazli, Sanaei, Ahmed, et al. 2013). To fulfill diverse computational and QoS (Quality of Service) requirements of numerous different mobile applications and end-users, several CMA solutions (reviewed in (Abolfazli, Sanaei, Ahmed, et al. 2013; Abolfazli, Sanaei & Gani 2012)) have been undertaken that suggest four major architectures for MCC. The major differences in suggested MCC architectures are emanated from various cloud-based resources with different features, namely multiplicity, elasticity (defined later in this chapter), mobility, and proximity level to the mobile users. Multiplicity refers to the abundance and volume of cloud-based resources and mobility is unrestricted movement of the computing device while its wireless communication is uninterruptedly maintained. Employed resources can be classified into four types of Distant Immobile Clouds (DIC), Proximate Immobile Clouds (PIC), Proximate Mobile Clouds (PMC), and Hybrid (H). Consequently, efforts can be classified under four architectures namely MDICC, MPICC, MPMCC, and HMCC that M represents mobile devices and ending C shows computing action.

Deploying various cloud-based resources in CMA solutions is not a straight forward panacea and existing resource diversity complicates system management and maintenance. Several open challenges such as high augmentation overhead, application dependency to the underlying platform (known as portability), lack of interoperation among various mobile devices and cloud-based resources, absence of standardization, and mobility management require future efforts before successful adoption of MCC solutions.

The remainder of this chapter is as follows. Section 2 presents an overview of MCC including its motivation, definition, and major building blocks. Section 3 briefly reviews four fundamental architectures for MCC. Section **Error! Reference source not found.** provides a brief discussion on open challenges and chapter is concluded in Section 5.

## 2. MOBILE CLOUD COMPUTING

In this section, we present motivation to augment mobile devices and present MCC definition. We also devise taxonomy of MCC building blocks and explain them accordingly. Moreover, major differences between cloud computing and MCC are explained.

### 2.1.Motivation

The MCC motivation lies in intrinsic deficiencies of mobile devices and realization of ever-increasing computing requirements of mobile users (Abolfazli, Sanaei, Gani Abdullah, et al. 2013; Abolfazli, Sanaei, Ahmed, et al. 2013). The miniature nature and mobility requirement of mobile devices impose significant constraints on their CPU, RAM, storage, and battery. Mobile device manufacturers are endeavoring to enhance computing capabilities of mobile devices by employing energy efficient multi-core processors, large fast RAM, massive low-overhead storages, and high charge density (long life) battery. However, technological limits, monetary and temporal deployment costs, weight and size, and user safety regulations decelerate mobile device empowerment.

Alternatively, researchers use the concept of on-demand, rich computing resources of cloud computing to alleviate mobile devices' limitations and fulfill insatiable users' demands, that breeds the state-of-the-art Mobile Cloud Computing (MCC) (Abolfazli, Sanaei, Ahmed, et al. 2013). MCC researcher envision enhancing computational capabilities of contemporary mobile devices to enable users to perform unrestricted computing, functionality, and mobility anywhere, anytime, from any device.

### 2.2.Definition

MCC "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" (Sanaei, Abolfazli, Gani, et al. 2013).

Computing resource richness in MCC is realized by exploiting computational power of one/several computing entities, including giant clouds, desktop computers in public places, and resource-rich mobile devices that inherit cloud computing technologies and principles, that are named cloud-based resources. Resource elasticity, as the most profound feature of cloud computing, enables automatic on-demand provisioning and deprovisioning of computing resources. Elasticity allows service consumers to use adequate amount of computing resources that match their requirements. The resources can be instantaneously acquired when necessary and can be released when not in use with minimum effort. Hence, mobile users pay (depend on the service delivery model) only for the resources consumed in time unit.

Moreover, accessing varied cloud-based resources in MCC does not necessitate communication through the Internet, whereas in stationary computing, cloud services are dominantly delivered via the risky channel of Internet. In MCC, services can more effectively be delivered using a local network via WLAN, regardless of networking technologies and standards.

It is noteworthy that MCC involves execution of only those applications that require extensive computational resources beyond native mobile devices. If a user starts an application on a mobile device and connects to the cloud to monitor resource utilization or VM status inside the cloud, it is not MCC. Similarly, when a mobile user consumes service of an application such as Facebook that is hosted in cloud server, there is no MCC.

### 2.3.Building Blocks

In this section, main building blocks of MCC are studied from two aspects of hardware and software. In every MCC system, hardware building blocks provide a rich mobile computation platform that can be employed by varied software programs. Our devised taxonomy is illustrated in Fig. 1 and explained as follows.

#### 2.3.1. Hardware

Hardware infrastructures, including heterogeneous resource-constraint mobile devices, cloud-based resources, and networking infrastructures are tangible building blocks of MCC. Heterogeneity in MCC is inherited from mobile and cloud computing technologies and is intensified in the existence of a multitude of dissimilar devices, infrastructures, technologies, and features (Sanaei, Abolfazli, Gani, et al. 2012).

- Mobile Devices: MCC is fraught by multitude of heterogeneous battery-operating wirelessly-connected mobile devices (e.g., Smartphones, tablets, and wearable computers) that feature varied limited computing capabilities.

- Cloud-based Resources: Computing entities that are built based on cloud computing technologies and principles (e.g., elasticity and pay-as-you-use) are called cloud-based resources. Four types of cloud-based resources are identified in (Abolfazli, Sanaei, Ahmed, et al. 2013) as Distant Immobile Clouds (DIC), Proximate Immobile Clouds (PIC), Proximate Mobile Clouds (PMC), and Hybrid (H) that are discussed in Section 3.

- Networking Infrastructures: Efficient, reliable, and high performance networking in MCC necessitates deployment of both wired and wireless networking technologies and infrastructures. Although mobile devices perform only wireless communications, immobile cloud-based resources require wired communication to transmit digital contents to different computing devices in a reliable and high speed medium.

### 2.3.2. Software

The software building block of MCC comprises augmentation protocols and solutions to efficiently leverage cloud-based resources to mitigate shortcomings of mobile devices.

**Cloud-based Mobile Augmentation (CMA):** CMA "is the-state-of-the-art mobile augmentation model that leverages cloud computing technologies and principles to increase, enhance, and optimize computing capabilities of mobile devices by executing

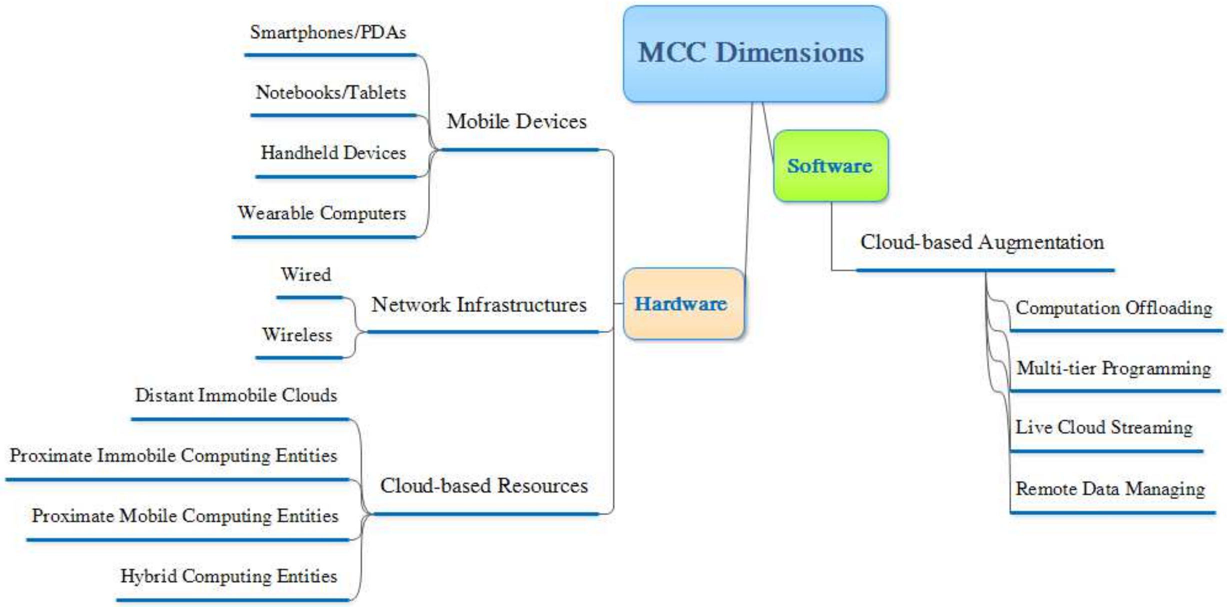


**Fig. 1. Taxonomy of Mobile Cloud Computing Building Blocks**

resource-intensive mobile application components in the resource-rich cloud-based resources" (Abolfazli, Sanaei, Ahmed, et al. 2013). Major CMA approaches consist of computation offloading, live cloud-streaming, multi-tier programming, and remote data managing.

**a) Computation offloading** is the process of identifying, partitioning, and migrating resource-intensive components of mobile applications to cloud-based resources. Identifying intensive components and partitioning can take place in three different approaches of static, dynamic, and hybrid. Static partitioning is a one-time process of identifying and partitioning the intensive components of mobile application at design and development time. The benefit of static partitioning is that it does not impose runtime overhead on a mobile device and once the application is partitioned, the same partitions can be used for infinite executions. However, static partitioning is not adaptable to the environment changes and its dynamism. On the contrary, in dynamic partitioning the identification and partitioning take place at runtime to better meet dynamism of the MCC environment. The challenge in dynamic partitioning is the excessive overhead of identifying intensive tasks, monitoring the environment, partitioning the application, and offloading the components. The third approach is to use a hybrid model where some part of the application is partitioned at design time and some at runtime to mitigate the partitioning overhead and adapt to the environmental changes. Despite significant efforts in offloading (efforts are reviewed in our previous work (Abolfazli, Sanaei, Ahmed, et al. 2013; Abolfazli, Sanaei, Gani Abdullah, et al. 2013)), offloading performance is degraded due to the overhead of partitioning and content offloading (Shiraz et al. 2012).

**b) Multi-tier Programming** is proposed and employed (Sanaei, Abolfazli, Shiraz, et al. 2012) to alleviate the overhead of code partitioning and offloading by building loosely coupled applications that perform resource-intensive computations (often web services) in the remote resources and minimize mobile-side computations. Resource-intensive computations are always available in the remote servers to be called for execution. Thus, the overhead of identifying, partitioning, and migrating tasks from mobile device to remote resources is omitted. Upon successful execution of the intensive tasks, the results are synchronized with the native components in the mobile device. In this model, only data is transmitted to the remote resources and codes are not migrated from the mobile device. Thus, the transmission overhead is significantly reduced. At application runtime, when the execution reaches the intensive components, it pauses local execution and transmits application stack memory and raw data to the remote resources for execution. Upon completion of the execution, results are integrated and execution is resumed. However, application functionality in these solutions depends on the remote functions and services whose failure impacts on the application execution. For instance, the speech recognition component in navigation applications is a resource-intensive task whose execution with acceptable accuracy is impossible inside the mobile device.

**c) Live cloud streaming** is another approach that aims to augment mobile devices by performing the entire computations outside the mobile device. Results are delivered to users as pre-compiled screenshots streaming to the mobile device lively. This approach requires low latency, high throughput, reliable wireless network that is challenging to establish using current technologies. For instance, onLive[1] delivers fascinating games to the mobile users via live cloud streaming.

[1] http://www.onlive.com/

**d) Remote Data Managing solutions** such as Dropbox virtually expand mobile storage by storing users' digital contents in the cloud-based resources. Parallel to growth in computing, digital data are sharply increasing that demand huge space on mobile devices which further encumbers mobile device adoption and usability. Although cloud storages enhance storage deficiency of mobile devices and improve data safety, trust and data security and privacy prevent users to intuitively leverage remote storages.

## 2.4. Mobile Cloud Computing vs Cloud Computing

Although MCC inherits cloud computing traits, significant fundamental differences exist between these two technologies that are summarized in Table 1 and explained as follows.

Cloud computing aims at providing rich elastic computing resources for desktop clients, whereas MCC envisions serving mobile users and realizing unrestricted functionality on the go. Service providers also differ in cloud computing and MCC. Resources in cloud computing comprised of one or more unified computing entities working in a parallel and distributed manner that are known as cloud datacenters under corporate ownership located in the vendor's premise. Nevertheless, resources in MCC can be any computing device inheriting cloud technologies and principles capable of mitigating resource deficiencies of mobile devices, that are referred to as cloud-based resources (Abolfazli, Sanaei, Ahmed, et al. 2013).

Cloud computing leverages only wired communications, whereas MCC utilizes both wired and wireless devices. Although wireless is the dominant communication mode in MCC, immobile cloud-based resources leverage wired networks to enhance computational experience of the end-users. Wired network is beneficial in areas such as live Virtual Machine (VM) migration (Clark et al. 2005) which is an emerging phenomenon that aims to mitigate the impacts of user mobility in augmentation process. Live VM migration in MCC allows immobile computing entities to transfer the running computational tasks over the wired network to a computing device which is proximate to the new location of the nomadic user.

Another major difference between cloud computing and MCC is their objectives. The former aims to reduce the ownership and maintenance costs of running private data centers by introducing the concepts of resource elasticity and pay-as-you-use. Cloud computing promises on-demand elastic resources by which desktop users can automatically provision computing resources and pay accordingly. Researchers in cloud computing endeavor to improve resource utilization rate, minimize the energy cost of intensive computing, and shrink negative impacts

**Table 1. Comparison of Major Cloud Computing and Mobile Cloud Computing Characteristics**

| Characteristics | Cloud Computing | Mobile Cloud Computing |
|---|---|---|
| Service Consumers | Desktop Users | Mobile Users |
| Service Providers | Giant Datacenters | Cloud-based Resources |
| Network Carrier | Wired | Wired/Wireless |

| Objectives | Elasticity, Pay-As-You-Use | Mobile Augmentation |
|---|---|---|
| Energy Solutions | Conserve energy & emit less CO2 on server side | Conserve Client's Battery |
| Mobility | Neither client nor server | Both client and server |
| End-users' major considerations | Monetary ownership and maintenance costs of proprietary resources | Temporal, energy, and communication overhead |

on the environment. However, MCC envisions augmenting computational capabilities of mobile devices by enabling long time execution of resource-intensive mobile computing tasks. Mobility is not provisioned for cloud data centers or desktop service consumers, whereas it is necessary for service consumers and feasible/beneficial for service providers. Users in cloud computing are concerned about high resource availability and saving monetary costs of executing intensive computations on demand, whereas mobile service consumers utilize cloud-based services to enhance application execution time, reduce energy consumption of the mobile device, and lessen the wireless communication cost (in the absence of monthly flat communication plans).

## 3. MOBILE CLOUD COMPUTING ARCHITECTURES

Numerous MCC proposals have been investigated over the last few years developing four major complementing architectures for MCC, which are briefly discussed in this section. Table 2 presents major characteristics of four MCC architectures. It is noteworthy that these architectures are applicable to all cloud-based augmentation models described in Fig. 1.

- **MOBILE DISTANT-IMMOBILE-CLOUD COMPUTING (MDICC)**: A general abstract architecture for a typical MCC system comprises of a mobile device that consumes computing resources of a computing entity using a typical offloading framework like MAUI (Cuervo et al. 2010) via a bidirectional wireless link. The first proposed architecture for MCC is depicted in Fig. 2 (a) where the mobile user is consuming computational resources of public clouds using the channel of Internet. Computational tasks in this model are executed inside the

**Table 2. Major Characteristics of Varied Mobile Cloud Computing Architectures**

| Characteristics | MDICC | MPICC | MPMCC | HMCC |
|---|---|---|---|---|
| Architecture | Client-server | | Client-server/peer-to-peer | |
| Heterogeneity | High | High | Low | Medium |
| WAN Latency | High | Medium | Low | Medium |
| Resource Elasticity | High | Medium | Low | High |
| Resource Multiplicity | Low | Medium | High | High |
| Resource Availability | High | Low | Medium | High |
| Mobility Implication | Medium (client-side mobility) | | High since both can move | |
| Utilization Cost | Low | Medium | High | Medium |
| Security & Privacy | High | Medium | Low | High |
| Trust | High | Medium | Low | High |

DIC resources (i.e., public cloud service providers such as Amazon EC2) and the results are sent back to the mobile client. The main advantages of DIC are high computational capabilities, resource elasticity, and relatively high security. High computing capabilities and elasticity of DIC minimize remote computational time and conserve mobile battery. The utilization cost of DIC datacenters is the least possible cost considering the ultimate goal of cloud computing to reduce the computing costs.

However, existing architecture, hardware, and platform heterogeneities between DIC resources (x86 architecture) and mobile devices (ARM-based) complicate code and data portability and interoperability among mobile and cloud computers (Sanaei, Abolfazli, Gani, et al. 2013). Heterogeneity also imposes excess overhead by employing handling techniques such as virtualization, semantic technology, and middleware systems. Moreover, DIC are coarse location granular resources (meaning they are few in numbers located far away from majority of mobile service consumers) and are intrinsically immobile computing resources. Thus, exploiting DIC for augmenting mobile devices originates long WAN latency due to manifold intermediate hops and high data communication overhead in the intermittent wireless networks. Long WAN latency degrades the application execution performance and wastes limited mobile battery. Moreover, service consumers' mobility on one hand and lack of clouds' mobility on the other hand intensify WAN latency, and degrade effectiveness and efficiency of MCC solutions.

In fact WAN latency will likely remain in wireless communications for a long time despite of significant improvements in data transmission speed and network bandwidth (Satyanarayanan et al. 2009). The main delaying factor in WAN latency is the processing delay at each intermediate hop to perform tasks such as decompression, decryption, security checking, virus scanning, and routing for each packet (Abolfazli, Sanaei, Alizadeh, et al. 2013). Thus, the larger is the number of hops, the longer is the WAN latency.

- **MOBILE PROXIMATE-IMMOBILE-CLOUD COMPUTING (MPICC):** To mitigate the impacts of long WAN latency, researchers (Satyanarayanan et al. 2009) endeavor to access computing resources with least number of intermediate hops and propose alternative architecture for MCC depicted in Fig. 2 (b). Hence, mobile devices utilize computing resources of desktop computers in nearby public places such as coffee shops and shopping malls. Instead of travelling through numerous hops to performing intensive computations in DIC, tasks are executed inside the one-hop distance public computers in vicinity (called PIC). PICs are medium location-granular (compared to the coarse grain resources, PICs are more in number and are located nearer to mobile service consumers) and feature moderate computational power that provide less scalability and elasticity. Moreover, employing computing resources of PICs holds several implications, particularly service provider and consumers' security and privacy, isolating a computer's host OS from a guest mobile OS, incentivizing computer owners and encourage them to share resources with nearby mobile devices, lack of on-demand availability (shops are open in certain hours and days), and lack of PIC mobility that demand future research.

- **MOBILE PROXIMATE-MOBILE-CLOUD COMPUTING (MPMCC):** The third MCC architecture depicted in Fig. 2 (c) is recently proposed (Marinelli 2009; Abolfazli, Sanaei, Shiraz, et al. 2012) to employ a cloud of nearby resource-rich mobile devices (i.e., PMCs) that are willing to share resources with proximate resource-constraint mobile devices. Rapidly emerging popularity and ever-increasing multiplicity of contemporary mobile devices, especially smartphones and tablets realize the vision of building PMC. Two different computing models are feasible which are peer-to-peer and client-server. In peer-to-peer, service consumers and providers can directly communicate with each other to negotiate and initiate the augmentation. However, service consumer needs to perform energy-consuming node discovery task to find appropriate mobile service provider in vicinity. Moreover, peer-to-peer systems are likely vulnerable to fraudulent service providers that can attack the service consumer device and violate its privacy. The alternative MPMCC communication model is arbitrated client-server in which mobile client communicates with a trusted arbitrator and request for the most reliable and appropriate proximate node. The arbitrator can keep track of different service providers and perform security monitoring on mutual communications. The crucial advantages of exploiting such resources are negligible service provider and consumers' heterogeneity, resource pervasiveness, and short WAN latency. In either client-server or peer-to-peer models, the number of intermediate hops are small due to service provider and consumer's vicinity. The resource pervasiveness of PMCs enables execution of resource-hungry tasks anytime anywhere (either in an ad-hoc ecosystem or infrastructure environment where Mobile Network Operators (MNO) like Verizon can manage the process). Therefore, diminutive WAN latency, and high multiplicity and ubiquity of mobile service providers establish a solid ground to leverage such ubiquitous resources in high latency-sensitive, low security-sensitive mobile computational tasks.

Although resource multiplicity in this architecture is high, scalability and elasticity is limited due to constraint computing power of individual mobile devices. Mobility management is another challenging feature of this architecture. Unrestricted mobility of mobile service providers and mobile service consumers in this architecture significantly complicates seamless connectivity and mobility, and noticeably degrade efficiency of augmentation solutions. When a user (either service consumer or provider) starts moving across heterogeneous wireless networks with dissimilar bandwidths, jitters, and latencies, the varying network throughputs, increasing mobile-cloud distance, and frequent network disconnections increase WAN latency

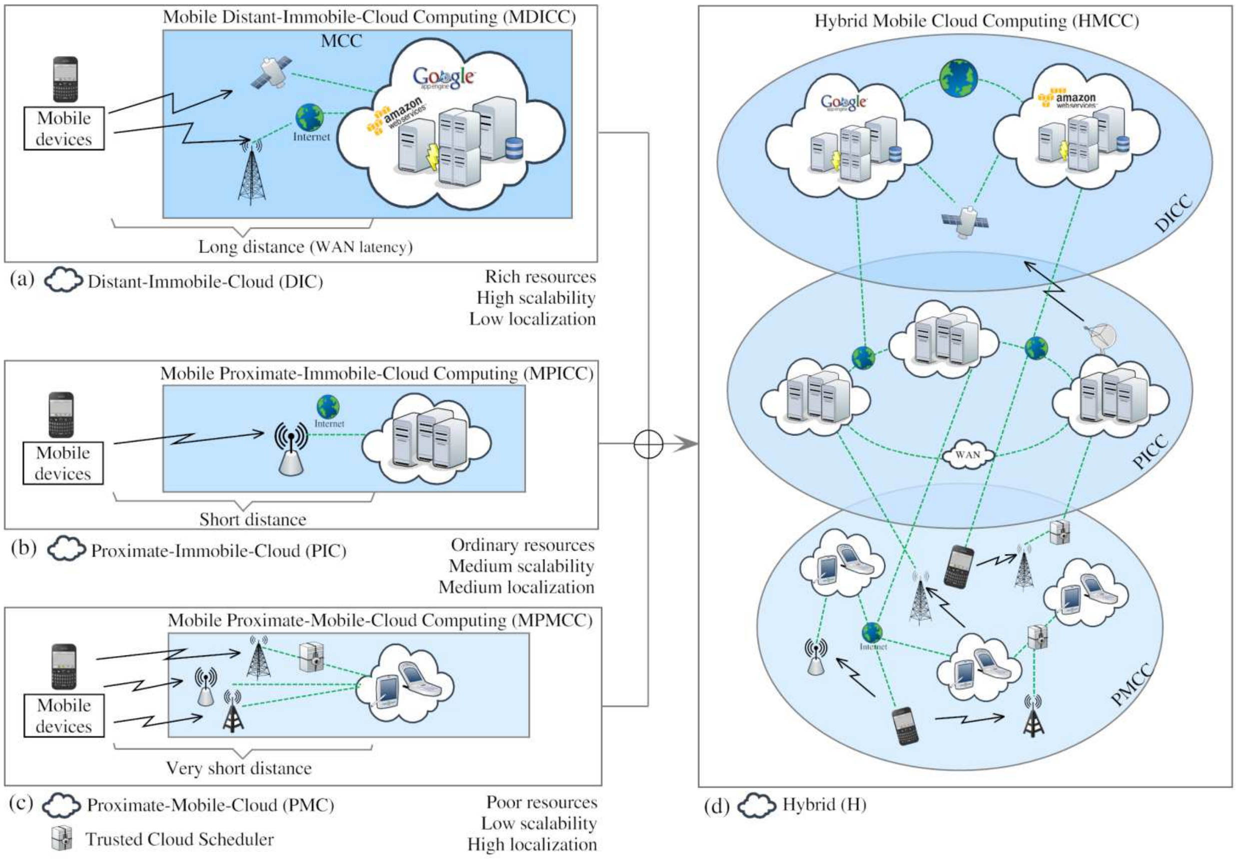


**Fig. 2. Illustrative View of Varied Mobile Cloud Computing Architectures**

and directly impact on application response time and energy efficiency. However, the utilization cost of PMCs is likely higher than other immobile resources due to ubiquity and negligible latency, though they feature finite computing resources. Other shortcomings of this model are security, privacy, and data safety in mobile devices. Mobile devices are not safe to store user data because they are susceptible to physical damage, robbery, hardware malfunction, and risk of loss.

- **HYBRID MOBILE CLOUD COMPUTING (HMCC):** Each of the mentioned three architectures features cons and pros that encumber optimal exploitation for efficient mobile computation augmentation. Researchers in (Sanaei, Abolfazli, Khodadadi, et al. 2013) demonstrate feasibility of consolidate various resources to build a HMCC model (see Fig. 2 (d)) that mitigates deficiencies of pervasive systems and promotes strength towards optimal CMA. SAMI (Sanaei, Abolfazli, Shiraz, et al. 2012) is a multi-tier infrastructure that convergences public clouds, MNOs, and MNOs' trusted dealers to optimize the augmentation of resource-constraint mobile devices. In the core of the multi-tier infrastructure, a resource scheduler program needs to evaluate each computational task and allocate appropriate resource(s) from the silo of heterogeneous cloud-based resources that optimally meet computational needs and fulfill user's QoS requirements (like cost, security, and latency). However, considering challenges, especially seamless mobility, significant MCC heterogeneity, wireless network intermittency, and current wireless networking, developing an optimal generic scheduler is a non-trivial task. Moreover, increasing number of mobile service consumers and hybrid cloud-based resources increase system complexity and complicate management and maintenance. Lightweight resource discovery and scheduling algorithms are essential for this MCC architecture.

## 4. OPEN CHALLENGES

Although advancements in MCC research are significantly impressive, several open challenges demand future efforts that are discussed as follows.

### 4.1.Lightweight Techniques

Resource-poverty in mobile computing is the major thrust that necessitates development of lightweight techniques for mobile consumers (e.g., offloading techniques), cloud service providers (e.g., light resource scheduling methods), and network providers (lightweight signal handoff). Native CPU, RAM, storage, and battery are major resources to be conserved

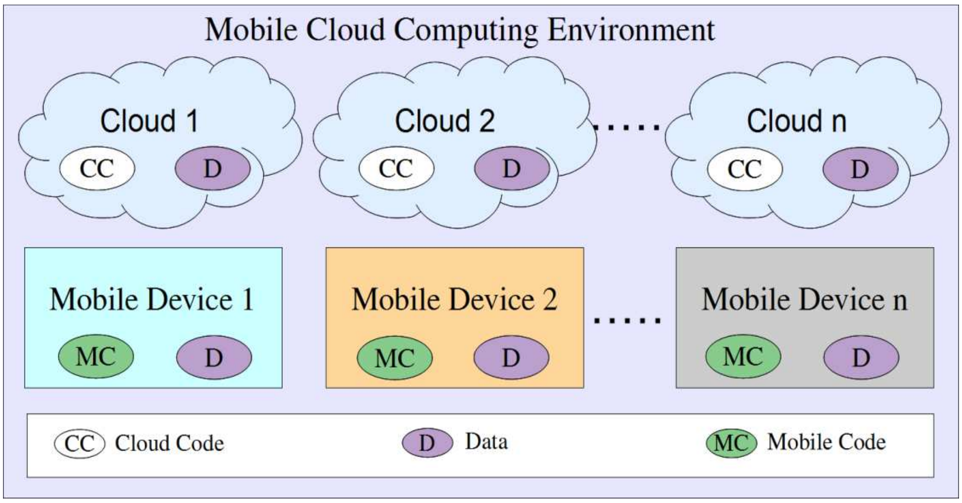


**Fig. 3. Portability in Mobile Cloud Computing**

when performing intensive computation. To realize lightweight approaches in MCC, shrinking application dependency to underlying platforms (towards portability) is significantly beneficial. Moreover, exploiting lightweight communication technology (i.e., wireless local area network compared to cellular), omitting excess/redundant native computation outsourcing protocols, using light data compression algorithms, and leveraging nearby high performance cloud-based resources are feasible and beneficial solutions.

### 4.2.Portability

Multitudes of heterogeneous mobile and cloud operating systems, programming languages, APIs (Application Programming Interface), and data structures fragment the MCC domain and encumber porting contents among various computing entities. Portability in MCC illustrated in Fig. 3 refers "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" (Sanaei, Abolfazli, Gani, et al. 2013) with little/no modification or configuration which are non-trivial tasks in the absence of standards, technologies, and solutions to handle heterogeneity in MCC. To realize portability in MCC, automatic code convertor solutions such as PhoneGap[2] that regenerates codes for different platforms, early standardization, and lightweight heterogeneity handling techniques such as service oriented architecture that enables developing loosely coupled mobile-cloud applications, are promising.

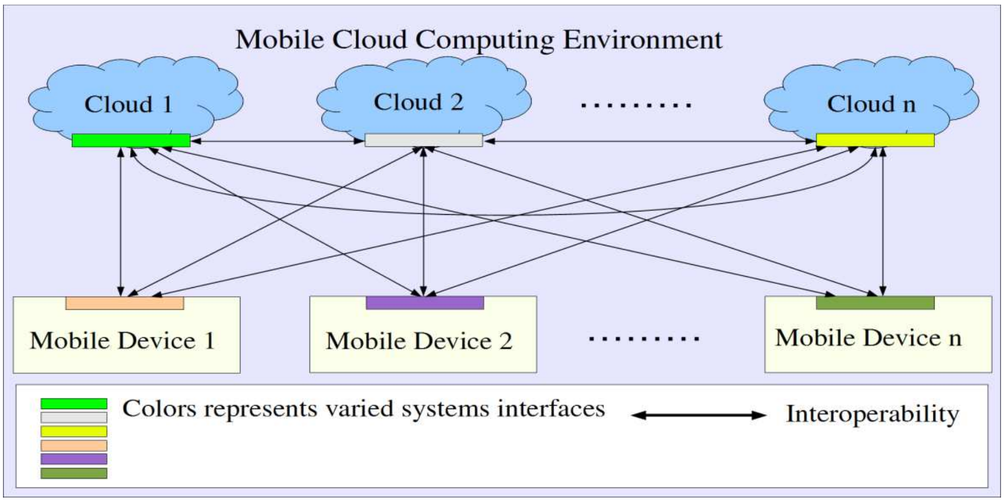


**Fig. 4. Interoperability in Mobile Cloud Computing**

### 4.3.Interoperability

Interoperability in MCC shown in Fig. 4 refers to collaboration of inter-clouds, mobile-cloud, and inter-mobiles with heterogeneous APIs and data structures which is a challenging task.

[2] http://phonegap.com/

Lack of interoperability in MCC breeds vendor lock-in problem which locks user data and applications inside certain cloud. When users change service providers (mainly due to quality and cost issues), migrating content originates high monetary and temporal costs of porting codes and data from one format to another and transmitting them from old provider to the new cloud provider. Still risks of code and data corruption during conversion and transmission threaten cloud consumers in MCC. Addressing interoperability in MCC demands standard protocols and common APIs such as Open Cloud Computing Interface (OCCI)[3].

### 4.4.Seamless Connectivity

Mobility in MCC is an inseparable property of service consumers and mobile service providers that obliges seamless connectivity. Establishing and maintaining continuous consistent wireless sessions between moving service consumers and other computing entities (e.g., mobile devices and clouds) in the presence of heterogeneous networking technologies requires — similar to the wireless hand-off— future research and developments. Lack of seamless connectivity increases application execution time and mobile energy consumption due to frequent session disconnections and interruptions that substantially degrade user experience. Seamless connectivity across heterogeneous wireless ecosystems requires solutions such as next generation all IP-based infrastructures wireless networks.

### 4.5.Live VM Migration

Executing resource-intensive mobile application via VM migration-based application offloading frameworks, involves the encapsulation of an application and migrating it to the remote VM, which are non-trivial tasks due to additional overheads of deploying and managing a VM on mobile devices. Live VM migration between distributed cloud-based resources (especially for DICs) is a vital need in executing intensive applications, considering wireless bandwidth, network intermittency, and mobility constraints. When a roaming user increases distance from the offloaded contents (code or data), the increased distance prolongs access latency and degrades user-observed application performance. Thus, mobilizing the running VM and migrating it to resources nearer to the user without perceivable service interruption becomes essential to avoid user experience degradation. Therefore, optimal solutions such as reactive and proactive migration of VM instances (proactive model requires predicting new user destination) to a place closer to the mobile user without service interruption, and solutions alike VMware vMotion[4] are vital to smooth live migration of VM and avoid user experience degradation. Consider a scenario that reactively initiates the migration of a VM to a server in Detroit as soon as you depart a New York airport and ends migration before you reach Detroit.

---

[3] http://occi-wg.org
[4] http://www.vmware.com/files/pdf/VMware-VMotion-DS-EN.pdf

## 5. CONCLUSIONS

Mobile Cloud Computing (MCC) is the state-of-the-art distributed mobile computing technology aiming to alleviate resource deficiency of multitude of heterogeneous resource-constraint mobile devices by performing resource-intensive computations inside cloud-based resources. Mobile augmentation solutions are highly influenced by the employed cloud-based resources. Granularity (resource multiplicity and proximity), computational capability, mobility, and heterogeneity between mobile device and resources are major influential resource properties that impact on augmentation performance and needs consideration in design and development of imminent augmentation solutions. Despite impressive MCC findings, several challenges such as lightweight low-latency architecture, live VM migration, efficient resource scheduling (automatically allocate resources to intensive mobile tasks), and seamless mobility require further efforts to enable MCC in real scenarios.

## REFERENCES

Abolfazli, S., Sanaei, Z., Alizadeh, M., et al., 2014. An Experimental Analysis on Cloud-based Mobile Augmentation in Mobile Cloud Computing, IEEE Transactions on Consumer Electronics, Feb 2014, pp. 1-9.

Abolfazli, S., Sanaei, Z., Ahmed, E., et al., 2014. Cloud-based Augmentation for Mobile devices: Motivation, Taxonomies, and Open Challenges. *IEEE Communications Surveys & Tutorials*, 16(1),pp.337–368.

Abolfazli, S., Sanaei, Z., Shiraz, M., et al., 2012. MOMCC: Market-oriented architecture for Mobile Cloud Computing based on Service Oriented Architecture. *Proceedings IEEE Mobile Cloud Computing Workshop*, Beijing,pp. 8– 13.

Abolfazli, S., Sanaei, Z., Gani Abdullah, et al., 2013. Rich Mobile Application: Genesis, Taxonomy, and Open Issues. *Journal of Network and Computer Applications*, Vol 40, pp. 345-362.

Abolfazli, S., Sanaei, Z. & Gani, A., 2012. Mobile Cloud Computing: A Review on Smartphone Augmentation Approaches. *Proceedings IEEE Mobile Cloud Computing Workshop,* Singapore,pp.199-204.

Clark, C. et al., 2005. Live Migration of Virtual Machines. *Proceedeings Symposium Networked Systems Design and Implementation,*Boston,pp. 273–286.

Cuervo, E. et al., 2010. MAUI: Making Smartphones Last Longer With Code Offload. *Proceedings ACM Conference Mobile Systems, Application, Services*, San Francisco,pp. 49–62.

Marinelli, E.E., 2009. *Hyrax: Cloud Computing on Mobile Devices Using MapReduce*. Computer Science Department, Carnegie Mellon University.

Rahimi, M. et al., 2012. MAPCloud: mobile applications on an elastic and scalable 2-tier cloud architecture. *Proceedings IEEE/ACM Conference Utility and Cloud Computing,* Chicago pp.83-90

Sanaei, Z., Abolfazli, S., Gani, A., et al., 2013. Heterogeneity in Mobile Cloud Computing: Taxonomy and Open Challenges. *IEEE Communications Surveys & Tutorials*,16(1),pp.369–392.

Sanaei, Z., Abolfazli, S., Khodadadi, T., et al., 2013. Hybrid Pervasive Mobile Cloud Computing: Toward Enhancing Invisibility. *INFORMATION-An international interdisciplinary journal*, 16(11),pp.1–12.

Sanaei, Z., Abolfazli, S., Shiraz, M., et al., 2012. SAMI: Service-Based Arbitrated Multi-Tier Infrastructure Model for Mobile Cloud Computing. *Proceedings IEEE Mobile Cloud Computing*,Beijing,pp. 14–19.

Sanaei, Z., Abolfazli, S., Gani, A., et al., 2012. Tripod of Requirements in Horizontal Heterogeneous Mobile Cloud Computing. *Proceedings Computing, Information Systems and Communications,Singap*ore,pp.217-222.

Satyanarayanan, M. et al., 2009. The Case for VM-Based Cloudlets in Mobile Computing. *IEEE Pervasive Computing*,8(4),pp.14–23.

Shiraz, M. et al., 2012. A study on Virtual Machine Deployment for Application Outsourcing in Mobile Cloud Computing, *Journal of Supercomputing*,63(3),pp.946–964.