

Lappeenranta University of Technology

Faculty of Technology Management

Degree program of Information Technology

**Mohamed Nuh**

**DEVELOPMENT OF AGNOSTIC MOBILE APPLICATIONS WITH CROSS-  
PLATFORM CLOUD COMPUTING PLATFORMS**

Examiners: Professor Kari Smolander  
Erno Vanhala

## **Abstract**

Lappeenranta University of Technology  
Faculty of Technology Management  
Degree Program in Computer Science

Mohamed Nuh

### **Development of agnostic mobile applications with cross-platform cloud computing platforms**

Master's Thesis

2014

121 Pages, 42 Figures and 1 Table

Examiners: Professor Kari Smolander

Erno Vanhala

Keywords: fragmentation, development platforms, market consumption, smartphones

Smart phones became part and parcel of our life, where mobility provides a freedom of not being bounded by time and space. In addition, number of smartphones produced each year is skyrocketing. However, this also created discrepancies or fragmentation among devices and OSes, which in turn made an exceeding hard for developers to deliver hundreds of similar featured applications with various versions for the market consumption.

This thesis is an attempt to investigate whether cloud based mobile development platforms can mitigate and eventually eliminate fragmentation challenges. During this research, we have selected and analyzed the most popular cloud based development platforms and tested integrated cloud features.

This research showed that cloud based mobile development platforms may able to reduce mobile fragmentation and enable to utilize single codebase to deliver a mobile application for different platforms.

## **ACKNOWLEDGEMENTS**

Foremost, I want to express my sincere gratitude to my supervisor Professor Kari Smolander for the continuous and priceless support of Master study and thesis, for his patience, encouragement and I could not have done this thesis without his guidance.

Finally, I want to thank my family, my wife Fadumo and my three children for their support, understanding, patience, assistance and faith in me.

Thank you

Lappeenranta, November 2014

Mohamed Nuh

## TABLE OF CONTENT

<b>1. INTRODUCTION.....</b>	<b>11</b>
1.1 Background .....	11
1.2 Objectives and restrictions .....	12
1.3 Structure of the thesis .....	12
1.4 Cloud Models .....	13
1.5 Vertical fragmentations .....	14
1.6 Factors that influence development costs.....	15
1.7 Fragmentation of Mobile Network.....	15
1.8 Defragmentation attempt through MWA .....	16
<b>2. CLOUD COMPUTING AS A SOLUTION.....</b>	<b>18</b>
2.1 Service Models .....	18
2.2 SaaS Model.....	19
2.3 PaaS Model.....	20
2.4 IaaS Model .....	22
2.5 Types of Clouds.....	23
2.6 Private Cloud.....	24
2.6.1 Advantages and disadvantages of private cloud .....	25
2.6.2 Key elements of private cloud .....	25
2.7 public cloud .....	26
2.7.1 Advantages and concerns of public cloud.....	26

2.8	Hybrid cloud.....	27
2.8.1	Benefit of hybrid cloud .....	28
2.9	Community Cloud .....	29
2.10	Cloud Layers.....	30
2.11	Cloud Architecture.....	32
2.12	Cloud computing and virtualization .....	37
2.13	Types of virtualization .....	38
2.13.1	Hardware virtualization.....	38
2.13.2	Partial virtualization.....	39
2.13.3	Process virtualization .....	39
2.13.4	System virtualization.....	40
2.13.5	Paravirtualization .....	40
2.13.6	Pre-virtualization.....	41
<b>3.</b>	<b>MOBILE CLOUD COMPUTING .....</b>	<b>42</b>
3.1	Introduction to mobile cloud computing.....	42
3.2	Characteristics of mobile cloud computing.....	42
3.3	Challenges of Mobile Cloud Computing.....	43
3.4	Cloud Computing Features.....	43
3.5	Mobile cloud services.....	44
3.6	Latency Challenges .....	45
3.7	Cloudlet Solution.....	46
3.8	Privacy challenges.....	48
3.9	MCC Architecture .....	48
<b>4.</b>	<b>CROSS PLATFORM SOLUTION FOR CLOUD COMPUTING APPLICATIONS.....</b>	<b>50</b>
4.1	Cross Platform Frameworks (CPF) .....	50

4.2	Cross platform framework for solving fragmentation challenges.....	52
4.3	Mobile Web Apps .....	54
4.4	Cross Platform Mobile Tools and Technologies .....	55
4.5	HTML 5.....	56
<b>5.</b>	<b>CRITERIA FOR EVALUATION OF CLOUD CONNECTIVITY OF MOBILE DEVELOPMENT PLATFORMS .....</b>	<b>60</b>
<b>6.</b>	<b>SELECTION OF CROSS-PLATFORM TOOLS .....</b>	<b>62</b>
6.1	PhoneGap .....	62
6.2	Rhomobile .....	64
6.3	MoSync .....	66
6.4	Xamarin .....	69
6.5	Appery.io.....	73
6.6	Appcelerator's Titanium.....	78
6.7	Telerik Platform .....	81
6.8	Kony .....	86
6.9	Parse .....	91
<b>7.</b>	<b>EVALUATION.....</b>	<b>95</b>
<b>8.</b>	<b>CONCLUSIONS .....</b>	<b>99</b>
	<b>REFERENCES.....</b>	<b>102</b>

## LIST OF SYMBOLS AND ABBREVIATIONS

AES	Advanced Encryption Standard
AI	Application Infrastructure
AMI	Amazon Machine Image
API	Application Programming Interface
CaaS	Communication as a Service
Capex	Capital Expenses
CI	Continuous Integration
CPF	Cross Platform Frameworks
CPU	Central Processing Unit
CRM	Customer Relationships Management
CSC	Cloud Service Customers
CSP	Cloud Server Providers
CSS	Cascading Style Sheet (CSS)
DaaS	Database as a Service
DOM	Document Object Model
ERM	Enterprise Resource Management
ERP	Enterprise Resource Planning

GPS	Global Positioning System
HA	High Availability
HaaS	Hardware as a Service
HTML	HyperText Markup Language
IaaS	Infrastructure as a Software
ICE	Integrated Cloud Environment
IPsec	Internet Protocol Security
LDAP	Lightweight Directory Access Protocol
MCC	Mobile Cloud Computing
MFA	Multi-Factor Authentication
MBaaS	Mobile Backend as a service
MWAs	Mobile Web Applications
MVC	Model-View-Controller
NDK	Native Development Kit
NIST	National Institute of Standard and Technology
NTLM	NT LAN Manager
OAuth	Open standard for authorization
OpEx	Operational Expenses
OS	Operating system
Paas	Platform as a Software



QoS	Quality of Service
REST	Representational State Transfer
RMA	Rich Mobile Applications
RPC	Remote Procedure Call
SaaS	Service as a Software
SDK	Software development Kit
SFA	Sales Force Automation
SLAs	Service Level Agreement
SMEs	Small and Medium Enterprises
SSE	Server-Sent-Event
SSL	Secure Sockets Layer
TCO	Total Cost of Ownership
TTM	Time to market
UI	User Interface
UX	User Experience
VM	Virtual Machine
VMM	Virtual Machine Monitor
WAC	Wholesale Application Community

# 1. INTRODUCTION

## 1.1 Background

Smartphone market is rapidly changing where usage of mobile applications is already common today. In addition, the characteristics of mobile phones have dramatically changed. Customers are consciously demanding more advanced features, such as Global Positioning System (GPS), Context-aware-location based application, better UX and advanced multimedia features. This will in turn invigorate the spread of mobile technology which will have a dramatic impact of the lives of individuals. In addition, mobility will provide a freedom of not being bounded by time and space. However, mobile development ecosystem is currently tightly bound and sensitively responsive to individual actors in Mobile ecosystem, where fiercely contesting device manufactures have heavily been investing their proprietary device technologies and services, such as user interface (UI), battery safety and life-cycle, UX superiority and camera design. Mobile OS providers attempt to deliver an effective and developer-centric OS that may invigorate rapid development.

Consequently, service providers, content providers, mobile application developers, OS providers and device manufactures became interdependent actors in mobile ecosystem. The astronomical investment and fierce competitions among device manufactures on one hand and among OS providers on the other hand may have provided a better device models in terms of capabilities and functionalities, such as touch screen, gravity sensors, better battery life-cycle. However, this also created discrepancies among devices and OSes, which in turn made an exceeding hard for developers to deliver hundreds of similar featured applications with various versions for the market consumption. Therefore, one of the biggest barriers in mobile application development is fragmentation [1].

Consequently, mobile application developers are facing ever-growing and enormous challenges directly resulting from countless differences within the same platforms, ever growing device

versions, rapidly changing OSes, diverging screen resolutions, countless specifications and various keypad types. Therefore, one may ask whether cloud computing model will eventually solve mobile fragmentation challenges.

## **1.2 Objectives and restrictions**

The primary purpose of this thesis is to examine the challenges that face mobile application developers to support multiple devices without coding for different platforms also known as the fragmentation challenges and also possible solutions brought by cross-platform mobile application development tools with possible cloud services. As a result, mobile application developers may be able to mitigate the risk of non-interoperability among vendors of mobile devices. Different cloud based cross platform development tools will be tested and compared with each other.

## **1.3 Structure of the thesis**

Section 2 describes opportunities and challenges facing mobile application development as well as mobile development eco-system. Further, it also describes the real actors of mobile application eco-system and their roles. Section 3 covers in detail the concept behind cloud computing as well as types of clouds. Further, Cloud services, such as software as a service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) were also described in detail. In addition, Cloud computing business models were discussed and several other concepts, such as Multitenancy were explained.

Also this section, cloud types such as public, private and hybrid cloud as well as cloud architecture were explained in details. Virtualization technologies which are the basic platform for the cloud technologies were also presented in this chapter. Section 4 was dedicated to

mobile cloud computing (MCC). In this chapter, MCC concepts and technologies behind it are discussed. Further, cloud security and mobile network threats have been treated in this chapter.

Section 5 disclosed the concept behind cross-platform solution for mobile applications, such as mobile web applications and several cross platform frameworks, while Section 6 presents criteria for evaluation different cloud based cross platform tools. Section 7 describes list of selected cloud based cross platform tools, while Section 8 was dedicated for evaluation of selected cross platform based on criteria. Section 9 describes summary of the research while Section 10 handles the conclusion of the collected information as well as suggestions.

## **1.4 Cloud Models**

On-demand services of Cloud computing are composed of three different service models, such as Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS)[4].

SaaS service model enables a complete application solution to be offered as a service to a user, who may access it over the internet [5]. In this way, service provider is responsible for the deployment, maintenance, update, and security of the application. Although, security and confidentiality issues are concerned, SaaS advantages over traditional software distribution include location-agnostic, seamless upgrade, data recovery and backup, instant deployment and zero infrastructure ownership [6]. Additionally, PaaS is an application platform that may offer services such as facilities for application design, application development, testing, and deployment as well as services such as team collaboration, web service integration, and marshalling, database integration, security, scalability, storage, persistence, state management, application versioning, application instrumentation, and developer community facilitate [7]. As a consequence, PaaS user can develop applications without installing any development kit or database server on local machine. The remarkable advantages of PaaS include faster time to market (TTM), reusable data formats, zero upfront cost, simplified deployment and development environment abstraction as well as time and cost effective [8].

Finally, IaaS, provides capabilities to the consumer in order to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications [9].

## 1.5 Vertical fragmentations

The vertical smartphone fragmentation – that is difference between devices that are based with common operating systems – seems to be worrying. Even though, Android has been in the market for shorter period of time, it has five different operating system versions [10]. One of the poignant cases related to this misery is the Angry Bird's tale. After Angry Bird's application game encountered massive malfunctions on different Android devices, Rovio has explained "With our latest update, we worked hard to bring Angry Birds to even more Android devices. Despite our efforts, we were unsuccessful in delivering optimal performance. So far, we have hesitated to create multiple versions of Angry Birds for the Android platform. We are currently developing a lighter solution to run Angry Birds on lower-end Android devices. This does not mean lighter game-play or a lesser amount of levels, but a game experience optimized for devices with less processing power [11]".

According to Larry Page, CEO of Google, 39 manufacturers around the world are currently producing devices based on Android [12]. As a consequence, the more smartphone devices penetrate into the market, the more fragmentations emerge and the exacerbation of mobile application development condition. For instance, smartphone screen sizes for Android devices differ significantly, that is, the values of ImageMaxWidth differ widely. According to Netbiscuits, the value ImageMaxWidth variable of 172 different Android devices has been compared, and the result was staggering. The result showed a broad range, from 220 up to 1280 pixels [13]. In addition to that, JavaScript attributes that provide those values are widely different [14]. Therefore, fragmentation is not limited to specific type of phone. Furthermore, as consequence of fragmentation among devices and OSes lengthens application testing, which in turn increases the cost of development and stretches time to market.

## **1.6 Factors that influence development costs**

In general, development cost is influenced by application portability and maintenance. Even though, mobile application delivery limited to specific and strategic mobile platforms may seem to be cost-effective in short term manner, but definitely not sustainable in long term. What is more, since portability challenges is direct proportional to the number of mobile devices, portability aspects that effects application development costs should be proactively addressed. Fundamental aspects of cost generators include target platform, development/maintenance toolkits and code reusability. Development/maintenance tools for testing, developing, building, packaging and finally deploying should be reusable as much as possible for different target platforms depending on type of mobile framework. Moreover, code reusability refers to utilizing single applications codebase to deliver a mobile application for different platforms. In this way, the higher degree of code reusability, the more cost-effective in terms of development and maintenance time, that in turn effects on other market drivers, such as time-to-market and manageability including authentication, authorization and upgradeability. Moreover, mobile application development in a fragmented ecosystem is a daunting task for developers, not to mention the administrative challenges, due to rapid changes in device features and non-standardized architectures. Additionally, from developers' point of view, the challenge lies not only in device fragmentation, but also fragmentation in skills in programming languages such as Objective-C, Java, and so on [15].

## **1.7 Fragmentation of Mobile Network**

Fragmentation of mobile network as well as other operator assets such as customer intelligence and billing system that can be used in conjunction with device capabilities to offer new payment possibilities, such as in-application billing has increased developer's concern. Some operators, such as Rogers [15] ,Telefonica[16], Orange [17], and Vodafone[18] have opened their network and billing solution through application programming interfaces (APIs) to mobile developers. However, with increasing number of APIs from network operators will increase fragmentation

in network domain and eventually obscure application success. As a consequence, some other solutions has been sought to solve this problem by attempting to standardize network APIs. In this respect, Gsmworld OneApi [19] and Wholesale Application Community (WAC) [20] have attempted to address this type of fragmentation by providing standardized API for billing and common set of specification for the network. Although this seems to be a huge step toward solving fragmentation pain, but application developer has to sign commercial contracts with each operator separately. In this way, this solution has created another fragmentation - fragmented commercial access points, at least developers' point of view. Consequently, de-fragmentation through mobile website or mobile cloud concepts are being discussed in mobile arena. For instance, over 80 percent of mobile app developers are building or planning to build mobile websites [21]. In this respect, mobile web applications (MWAs) can offer compatibility across devices, instant upgradability and easy development, thus, candidate for de-fragmentation solution.

## **1.8 Defragmentation attempt through MWA**

Almost all mobile devices on the market have preinstalled browsers providing extensive opportunities to reach greater audience. In addition to that, Mobile website application technologies, including HTML, JavaScript and Cascading Style Sheet (CSS) are very familiar to application developers. In this respect, MWAs may target mobile devices, such as smartphones, tablets and others with browsers that support APIs to access native capabilities. However, existing browser fragmentation among mobile devices may hinder wide acceptance of MWAs. Additionally, some browser engines may have limited capabilities to render web pages that use advanced technologies, while others may provide high responsiveness of the application due to support of web workers feature which offers web developers an opportunity to offload some tasks to a background process or thread. Furthermore, screen sizes and resolutions fragmentation among devices may still exacerbate portability nightmare. Although, amalgamation of useful techniques, such as abstracting layers via frameworks, CSS media queries via webkit browsers, pre-detection feature and browser type with feature detection libraries including Modernizr [22] will minimize fragmentation impacts on user experience,

more major weaknesses still exist in WMAs, such as discoverabilities issues, portability across devices due to discrepancies among browser and incapability of exploitation of device feature, hence inferior UX both in terms of contextual and implementation.

Though Mobile applications may contextually aware of the users' environment through the different mobile sensors, but due to the existence of several mobile platforms developers need to build several versions of these mobile applications [23]

Furthermore, end users may demand interactivity, such as gaming, personalization and also off-line accessibility. Additionally, some applications may need to use device's processing power, GPS and camera. In this respect, mobile websites may not necessarily offer the ultimate solution for fragmentation challenges. Therefore, alternative solution may be needed, such as cloud computing.



## 2. CLOUD COMPUTING AS A SOLUTION

Cloud computing is a term that can be defined as having IT processing as a service rather than as a product or software and easiest way to visualize it is to compare to electricity [24]. In addition, Cloud computing is an emerging computing model for enabling an on-demand network access to a shared pool of configurable computing resources, such as networks, servers, storage, applications, and services that can be rapidly provisioned and released with minimal management effort or service provider interaction [25]. Therefore, customers may avoid owning computing infrastructure through renting from computing providers. In this way, cloud users may benefit from the reduced cost and the rapid cloud service provisioning, in addition to their capability of expanding or reducing their computing infrastructures efficiently on-demand[26]. Another definition of cloud computing is that it describes a new supplement, consumption and delivery model for IT services based on internet, and typically involves the provision of dynamically scalable and often virtualized resources as a service over the internet[27]. In general, large numbers of cloud customers are small and medium enterprises (SMEs) as in many cases they cannot afford the large capital expenditure of traditional IT [28]. These SMEs will only pay on demand, as they do for other utilities such as water and electricity. Furthermore, as any other technology, cloud computing provides benefits as well as risks need to be mitigated. For instance, cloud computing benefits include increased scalability, pay-as-you-go model, disaster recovery, and increased mobility, while primary risks that may slow cloud adoption include vendor lock-in, security, privacy and confidentiality and poor service level agreements (SLAs).

### 2.1 Service Models

Cloud computing can be broken into three service models, namely, Application, Platform and Infrastructure. Application (Software as a service) is the highest layer in the cloud stack where end customers can purchase a working application [29]. In other words, SaaS solution is composed of some intrinsic elements, such as pay-as-you-go based contractibility, flexibility in

scalability and elasticity, share-ability of the resource, out-of-the box service-based. In this service, applications are transferred to millions of users through browsers [30], where providers and end users may save costs through maintenance and pay-as-you-go respectively. Therefore, consumers don't manage the underlying cloud infrastructure. What is more, no upfront investment or software licenses are needed. Salesforce is one of the best-known providers of this service, especially, Salesforce Customer Relationships Management (CRM) system [31]. In addition, SaaS is commonly used for HR apps and has been worked its way up the food chain to enterprise resource planning (ERP), with players such as Workday [32]. In any case, advantages of using SaaS include proliferation, data integration, exit strategy, and minimal customization [33].

## **2.2 SaaS Model**

SaaS service model enables a complete application solution to be offered as a service to a user, who may access it via user-centric interface. In this way, service provider is responsible for the deployment, maintenance, update, and security of the application. The basic principle of SaaS is letting customers to hire and acquire a deployed service through internet [34].

Further, SaaS can also be defined as a delivery model which offers customers possibilities to use provider's software applications running on a cloud infrastructure.

Although, security and confidentiality issues are concerned, SaaS advantages over traditional software distribution include location-agnostic, seamless upgrade, data recovery and backup, instant deployment and zero infrastructure ownership. Additionally, SaaS is remotely managed and delivered by external provider. In this regard, SaaS usage involves not only security and confidentiality risks but also service level agreements and contracts. Moreover, shared resources of the cloud may enhance economic scale benefits, such as price reduction. For instance, Gartner Group estimates SaaS sales in 2010 have reached \$10B, and are projected to increase to \$12.1b in 2011, up 20.7% from 2010 as SaaS revenue will be more than double its 2010 numbers by 2015 and reach a projected \$21.3b [35].

However, scalability, agility and shortened time to market may be primary drivers for SaaS adoption. In traditional approach, software delivery was based on an application instance for single user according to customer's requirement. Therefore, software application was very expensive, while security issues were addressed. On the other hand, multitenancy feature in SaaS model dramatically reduces costs but unfortunately raised security threats. Furthermore, multitenancy design approach is based on virtual partitions of data and configurations for letting client to work customized instance of the application. Therefore, Multitenancy should not be confused with multiuser. In addition, benefits of multitenancy approach include less complex maintenance of the application, hardware resources utilization and overall lower price. In multitenancy approach multiple organizational data and applications are handled in single instance while no two implementations of SaaS service model are exactly alike even though they're sharing the same centrally hosted system [36]

### **2.3 PaaS Model**

The second cloud model, Platform as a service (PaaS) or development as a service, deliver a solution stack which consumes cloud infrastructure to sustain SaaS [37]. This solution enormously facilitates application development and/or deployment. In addition, capability provided to use consumer is to provision storage, networks, processing and other resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications [38]. Although, consumers don't control underlying cloud infrastructure, but may have control over operating systems, storage, deployed/developed applications, and possibly selection of networking components.

In the context of cloud computing, application infrastructure (AI) embedded and enhanced with features of cloud computing is being offered to customers as a cloud-based products to enable business application and defined as PaaS. Therefore, PaaS is an application platform which provides a complete stack of development, building, operating applications and services, deployment, testing, hosting, storage, security and also versioning.

As a consequence, PaaS user can develop applications without installing any development kit or database server on local machine. Furthermore, cloud based application infrastructure (aka PaaS) solution has more benefits than application infrastructure located on-premises. For instance, Cloud PaaS offers multitenancy provisioning, tracking, and elastically horizontal scaling, customizable configurations, web interface, enhanced privacy and tenant bases security as well as tenant bases self-service administration.

As a result, remarkable advantages of PaaS include faster time to market (TTM), reusable data formats, zero upfront cost, simplified deployment and development environment abstraction as well as time and cost effective. In addition, PaaS shares a porous border with IaaS in terms of terminology and offering, thus increasing misunderstanding of cloud computing concepts. For instance, PaaS may be deployed on IaaS as Heroku on Amazon or may be detached from IaaS such as Force.com. In this regard, PaaS can be defined as a technology services layered between system infrastructure layer (IaaS) and SaaS, thus middleware solution.

Benefit of PaaS model include cost benefits, faster deployment and deployment cycles [39]. Although, some vendors may offer pure PaaS or pure IaaS, others may combine them as IaaS plus middleware which runs on virtual machine and offers as a complete solution, such as Netsuite [40] that markets amalgamation of SaaS and PaaS and isolated PaaS by Force.com [41]. Furthermore, Tools and Software APIs PaaS platform offers are similar to on-premises infrastructure, such as application and database management tools for both tenants to conduct self-service related operations through service layer and providers for managing technology stack of the platform, business process technologies, application servers, enterprise services, as well as application development tools. Finally, Cloud multi tenancy that is based on logically exclusive instances running on a shared environment, where each tenant (user) operates on exclusive resources in computing environment, can be implemented differently according to required type of multi tenancy, whether custom based or not, and the degree of depth of sharing.

Tenancy types, application and enterprise cannot be easily isolated from each other. For the enterprise type, collection of instances of the applications is running in a shared environment, where each instance serves individual platform user. In addition, application type tenant concept

is based on several applications running on shared environment where customers use them from the cloud.

In general, PaaS platform generally comes equipped with ready software developing technologies, such as Python, Java, LAMP and .Net, so as to enable consumers to focus on writing code.

## **2.4 IaaS Model**

Infrastructure as a service (IaaS), the capability provided to the customer is to provision processing, storage, networks, and other fundamental computing resources where the customer is able to deploy and run arbitrary software, which can include operating systems and applications [42]. Therefore, IaaS is an evolutionary process based on hardware delivery, such as network, storage combined with software delivery, such as OS as a cloud service.

Fundamental IaaS characteristics include internet based, on-demand scaling, hardware virtualization and management through orchestration process. Furthermore, this model can be defined as computer infrastructure delivery as a service with usage-based pricing model.

Advantages of IaaS include peak leveling and high scalability, while risks are potential vendor lock-in, privacy and security. Features of IaaS infrastructure that IaaS provider offers to the potential customers include management and networking capability, virtual environments, and persistence storage.

Additionally, IaaS platform offers provisioning and de-provisioning of on-demand computing resources where billing is based on usage metric. Depending on cloud and source type, several IaaS delivery models exist including outsourced private cloud, insourced private cloud and outsourced public cloud. In addition, IaaS generally acts as a computing and storage utilities, where scalable, flexible and on-demand based usage storage capabilities are delivery to the potential customers via standard web interface. Like any other models, IaaS has both advantages and disadvantages. For instance, while advantages include reduced ROI risk, low barriers to entry automated scaling, disadvantages include dependency of vendor's capability potentially greater long-term cost and security [43]

For the IaaS computing resource, there are various delivery models that IaaS provider can utilize depending on customer demands. For instance, IaaS customers who require a guaranteed computing resource will be offered a committed based approach, where provisioned resources are committed to this customer. Due to benefits of IaaS model, consumers are rapidly increasing. For instance, Amazon's EC2 gained increasing IaaS users [44].

Moreover, for customers who have not demanded any guaranteed resources may be offered to a simplified delivery approach, where provider will exert a best effort to allocate demanded resource, but nothing is guaranteed, and finally, middle ground for both approaches where only certain amount of computing resources are guaranteed for the customer to utilize, but customer may also use if there is any extra and unutilized capacity. IaaS components and characteristics include scaling, virtualization, and internet connectivity, automation of administration tasks, utility computing and billing model [45].

In addition, IaaS providers offer service level agreement (SLAs) that cover different types of cloud service elements including network based SLAs for connection and internet availability, Persistence (Storage) SLAs, which covers throughput based storage performance measuring or input/output operation per second. In addition, some organization may have the needs to create a hybrid IaaS cloud by combining an internal IaaS with an external cloud service using orchestrator for automating processes between clouds.

## **2.5 Types of Clouds**

Cloud computing technology consist of four different deployment models, namely, private, public, community and hybrid. Private Cloud is a form of cloud where services have a restricted access or the customer has some control of the service. A various definitions has been applied to a private cloud [46][47][48], which can be summarized as an environment which is not only capable of implementing cloud computing features such as layered services over the network and virtualization, but also applies more stricter policies and requirements, such as security, latencies and usage of datacenter resources. Although, private cloud is generally operated by corporate IT, it can also be off-premises and hosted by a third party while still

remain under the control of company's IT organization. In addition, much of IT enterprises have already moved to private clouds. For instance, large companies such as Bp, Intel, and IBM have virtualized their own resources and reaped the advantages of volume, statistical multiplexing, and utilization [49]. In particular, IBM has saved \$1.5 billion by consolidating its datacenters from 115 to 5 [50]. Consequently, private clouds hold the promise of offering all public cloud advantages while solving security and privacy concerns.

Furthermore, virtualization of servers and data stores is part of cloud computing, but do not by them constitute specific type of cloud.

## **2.6 Private Cloud**

Private clouds can be defined as a data center that has capabilities similar to a public cloud but is operated and owned by business entity [51]. Since private cloud is intrinsically service based, therefore, creation of service catalog and explicitly delivering as a service may eliminate service-level agreement discussions between providers and consumers.

Though, in general, private clouds can be insourced and run on-premises of the organization, but it can also be outsourced, and managed by external entity while providing access via secure network, while it may be operated by the organization, a third party or some combination whether on or off premises [52].

Furthermore, private clouds as well as other cloud types are based on resource virtualization technologies. But virtualization is not cloud computing and cloud computing can happen without virtualization [53]. In addition, several of datacenters also use server virtualization. Therefore, cloud is not synonymous with virtualization.

In addition, architectural elements of private cloud may include utility management, service management and access management. For instance, cloud consumers may be able to access computing utilities easily and seamlessly via web interface. Even though, virtualization is an essential component of private cloud, delivering abstraction of computing resources from

infrastructure and machines can enable for customers to achieve the benefit of private cloud, such as responsiveness, agility, business alignment and focus [54].

### **2.6.1 Advantages and disadvantages of private cloud**

One of the attractive features in private cloud may include the availability of standard services via a self-service interface and seamless deployment of those services that may dramatically decrease time-to-market. Therefore, speed may be regarded as one of the benefit of private cloud. Other key advantages include security because of resource dedication to a single organization, high performance, customizable greater control and deeper compliance, while disadvantages include Capacity Ceiling, higher cost and onsite maintenance [55].

Furthermore, security issues which are paramount concern in cloud business are addressed. As consequence, private cloud could be more expensive due to complexity and management.

### **2.6.2 Key elements of private cloud**

In this respect, key elements of private cloud architecture include centralized and standardized hardware resources, self-service access through web based interface with enabled on-demand service consumption, seamless and automated deployment mechanism to reduce manual administrations and ability to provision and de-provision elastically and on-demand asset scaling, multi-tenancy and pay-as-you-go which is based on chargeback model, in other words, management, automation, security and dynamic provisioning are the building blocks of any private clouds [56].

In order to enhance private cloud solution quality, access management and monitoring mechanism for both infrastructure and software level can be established. For instance, service usage chargeability which is based on chargeback model may deliver many benefits including, enabling cloud consumers to know their computer footprint, visibility into resource utilization and facilitating capacity planning [57].



Moreover, incident management for tracking raised issues and tickets, service SLAs management and its usage report are also necessary key elements for private clouds.

## **2.7 public cloud**

On the other hand, public cloud is a standard cloud computing model, in which dynamic resources are offered to the public via internet. In this way, public cloud is designed around the computing as a utility concept, where resources such as storage, computing, testing and developing platforms are delivered as a service and accessible seamlessly over internet and, by definition, is an external to the consumer as cloud consumers pay as you go (PAYG) model, while providers ensure resource separations, also known as multi tenancy [58].

Furthermore, consumers may able to reduce capex through sharing services usage with other consumers, thus shifting the responsibility to managing and maintaining complex IT platform, which is certainly not the core business model for many enterprises to cloud provider.

### **2.7.1 Advantages and concerns of public cloud**

One of the most attractive benefits of the public cloud is utility price model, where one pays for the computing per the hour. In addition, API access -which allow user to programmatically access server, configure and pay online are the benefits of public over other cloud models [59].

In general, public clouds are usually run by third parties where applications from different customers may be mixed together on the cloud servers as well as networks and storage systems and customer risks are reduced through hosting clouds away from customer premises. In addition, other applications running on the cloud can be made transparent to end users by implementing the cloud with performance, security, and data locality in mind [60].

Therefore, capability of on demand scalability may be smartly utilized by cloud providers as an added value, while shifting the burden of infrastructure risks from the cloud customer to the

provider. Although, public cloud has many advantages like high transaction, wide access to a large range of IT tools with virtually no restriction, but security is one of the drawbacks, since sensitive data are managed and controlled by a third party. For small companies with limited resources, public cloud may be an attractive choice to protect their data than own firewalls. However, for enterprises whose existence depends upon securing trade transactions, classified information and sensitive customer data, public cloud providers may not offer reliable protection. For instance, tremendous risk of multi-tenancy approach due to the possibility of unauthorized access of sensitive data, shared security responsibility between providers and consumers, sensitive data encryption, and changing regulatory requirements on data privacy. Therefore, public cloud solution should satisfy not only business agility and capex elimination but also to reflect privacy and security requirements of consumers. Furthermore, most providers may not meet standards for auditability and comply with legislations, such as Sarbanes-Oxley and the Health and Human Services Health Insurance Portability and Accountability Act (HIPAA) [61].

While each of cloud models (private and public) has an advantage and disadvantage, combining them will minimize the risks and maximize the benefit.

## **2.8 Hybrid cloud**

Hybrid clouds can be defined as infrastructure model that combines different cloud models, such as private and public or any other composition of two or more clouds (private, public and/or community) to match business needs. In addition, Hybrid cloud model can also be defined as private cloud with extension of service public cloud's low cost cloud storage to the enterprise [62]. In this way, remote public resources are integrated into the private cloud, thus hybrid cloud. Furthermore, private resources can be augmented with the dynamic public cloud resources, so that service levels can easily be maintained at the time of rapid workload fluctuations.

Additionally, Hybrid cloud is best suited to handle flash crowds by scaling out whenever the local capacity is exhausted.

In general, private cloud model provides a more optimized and controlled environment for seamlessly provisioning and deploying application workloads, while addressing security issues and eliminate latencies. However, when available resources are exhausted, flexible scalability may become an issue. In this case, public cloud may be used to horizontally scale to compensate exhausted capacity of private cloud.

Moreover, hybrid approach allows organizations to leverage the scalability and cost-effectiveness of a public cloud offering without exposing its critical data to external resources [63].

As a result, hybrid model may offer enterprise applications to the unlimited resources of the public cloud. These available multiple resource pools consequently may create unique challenges that need to be addressed, such as security, latency, cost, and complexity.

### **2.8.1 Benefit of hybrid cloud**

Hybrid benefits include capex reduction without compromising security, agility in business, and elastic scalability. In addition, flexibility is one of the benefits of hybrid cloud especially for companies wanting to capitalize on the benefit of both the private and public cloud approach [64]. Other benefits include saving expenses by deploying complicated processes on public cloud and no need to manage public infrastructure. Moreover, consumer with hybrid solution may run application's web interface in the public cloud due to scalability benefits while securing back-end databases in the private cloud on-premises.

Other hybrid cloud benefits may include cloud bursting, which is leveraging public resources when private resources are limited or price threshold reached, data recovery in case of disaster, and data backup on public storage with strong encryption mechanism as well as separating proprietary contents from public contents. In this respect, existing enterprises may be able to mix and match available services to maximize utilization of company assets, increase agility and level of performance, while enterprise may also easily move processes, services and applications from one geographic location to another [65].

However, there are some challenges in hybrid cloud, such as cost, where company should setup private data centers (e.g. hardware, power, cooling and maintenance) as well as the usage-based cost of cloud provider [66]. Private and hybrid cloud solution may be composed of several different but interacting components including virtual machines, nodes (physical servers), clusters (groups of nodes) racks or Pod (group of clusters), Zones (group of racks or Pod) built on separate islands for high availability and regions (group of Zones), and finally Manager and Controller of resources, such as instance provisioning and on-demand storage allocation.

## **2.9 Community Cloud**

Community cloud is another model, which can be defined as sharing infrastructure for increasing scale and reducing cost. Another form of community may be established by creating a virtual data center from virtual machines instances deployed on underutilized users machines [67]. Community cloud refers as a shared computing environment aimed at group of restricted and like-minded organizations who generally shares similar concerns in terms of security, performance, confidentiality and policy, agility, and elasticity considerations.

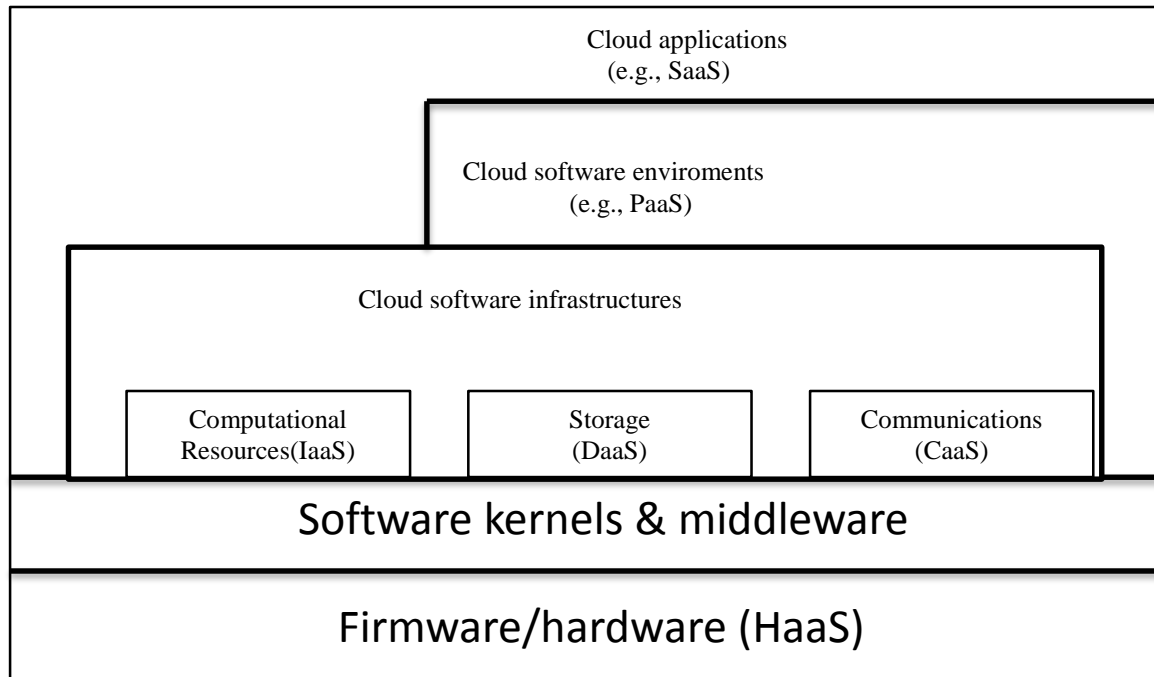
Unlike public cloud, community cloud model provides more security and confidentiality. As a result, this model is more expensive than public, but more cost-effective than private due to spreading cost across community members, such as banks, financial institutions and government agencies, thus enabling economic scale. Similar to hybrid model, community cloud model provides a composite of public and private challenges as well as benefits. In addition, community cloud provides similar services to all members of the community.

In this respect, community cloud model is based on common ownership of cloud facilities among members. Clouds in this model are tightly tailored to the common and shared needs of the community, such as financial agencies, healthcare or any other entity. Members of the community cloud may not only retain most of the benefits of the public ownership model, such as redundancy of data centers, cost reduction, geographic diversity but also maintain closer control over the location, security, data protection and may ensure compliance [68]. From members' perspective, cloud meets specific requirements, such as computing and security, so

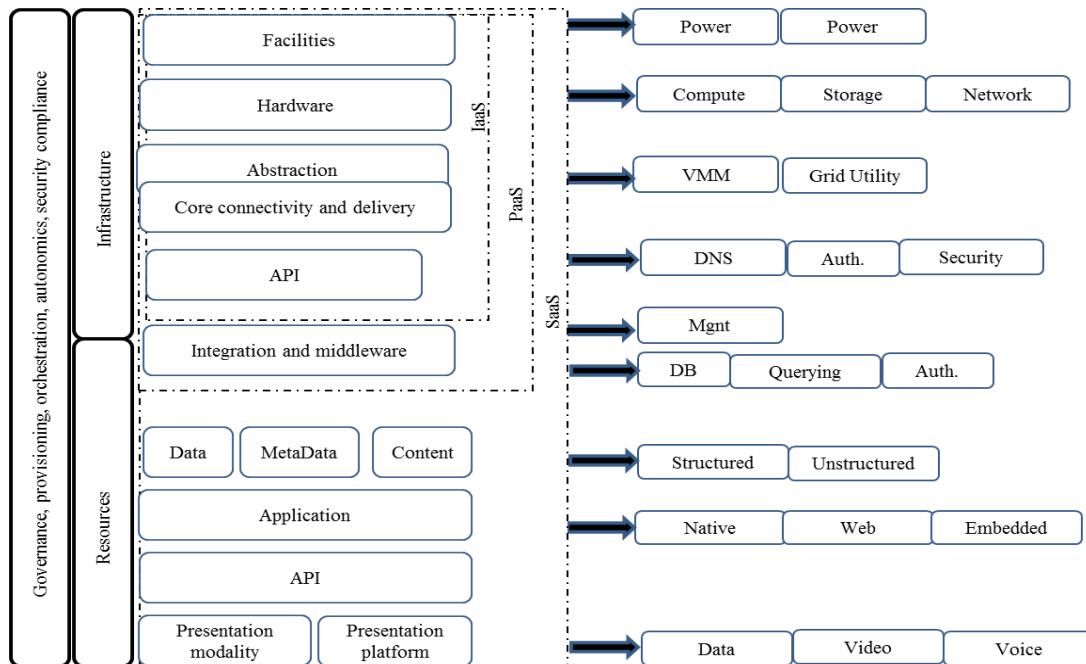
that multiple parties with shared concern may form an interest group or community. Therefore, some advantages of community model are in line with other cloud type, such as lowered capex, scalability, seamless deployment, but community model may include federated security and adaptive OpEx. Other cloud models include personal cloud which refers to seamless access of personalized contents from any location at any time. Further, community cloud is as much a social structure as a technology paradigm [69].

## 2.10 Cloud Layers

In UCSB-IBM cloud ontology, principle of compos-ability, which is an ability to assemble a collection of services to form composite services- from SOA had been used to classify the different layers of the cloud [70]. According to that ontology, there are five layers in the cloud. Application layer, which users can access via web portals and may also be based on pay-as-you-go price model. Platform layer is suitable for implementation and deployment of cloud applications. In this layer, well-defined APIs and various programming-language-level environments are provided to support smooth deployment and flexible scalability required by cloud application. For developers' point of view, this platform does not only simplify cloud application development, but also facilitates an easy integration with other services to their application. Infrastructure layer is the third layer which can be decomposed into three distinct but interdependent components, namely computational resources (IaaS), storage (DaaS), and communications (CaaS) [71]. Furthermore, kernel and hardware layers are the two bottom layers in UCSB-IBM cloud ontology (Figure 1). In addition, Hoff's Model (Figure 2) describes cloud ontology in more details [72]. In this model, infrastructure layer can be decomposed into several components, such as, facility layer which comprises electric power machines and their storage spaces. Hardware – including networking, storage, computing and Abstraction (VM monitors, Grid utility and so on ) are the second and third layers respectively, while Core connectivity, which manages security/accessibility and API (service management) are fourth and fifth layers in Hoff's model respectively. Furthermore, according to the model, PaaS is middleware where cloud application can be developed or integrated with other services. SaaS layer, which comprises of application, API, data/metadata locates on top of all layers.



**Figure 1.** UCSB-IBM Cloud Computing Ontology [71]

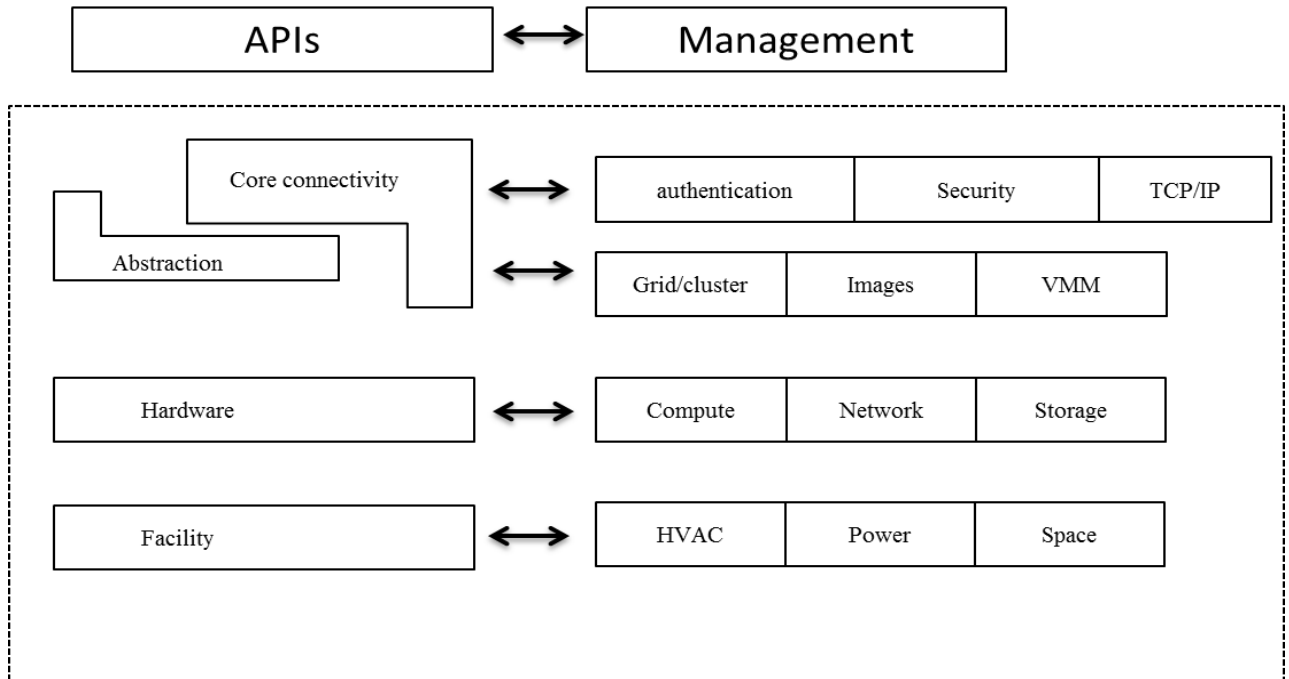


**Figure 2.** Hoff's cloud ontology [72].

## 2.11 Cloud Architecture

Cloud architecture can be defined as designs of applications that use on-demand services through internet. In general, cloud architecture is composed of two end parts, the front-end in which cloud consumers use to access cloud resources through web interface and the back- end which is cloud infrastructure and resources, such as computing and storage.

In addition, cloud applications regularly use the underlying computing architecture on-demand basis. While Cloud Server Providers (CSP) offer service applications as SaaS to Cloud Service Customers (CSC) that address particular needs, they lease computational resources, such as storage, network, computation and security from infrastructure providers presenting their services as IaaS. In cloud platform, large resources, such as physical servers are virtualized and then presented as multiple machines to run multiple OSes and instances. In this way, virtualized resources can be allocated for different customers through shared physical infrastructure. Therefore, virtualization technologies are the building blocks of cloud technology. System virtualization [73] can be defined as the ability to run multiple heterogeneous operating systems on the same physical server [74]. Therefore, virtual machines are containers that are assigned specific resources and the software that runs in the virtual machines is what defines the utility of the cloud computing system [75]. Figure 3 shows virtualized server where API is shown shaded in gray because of being optional component [76]

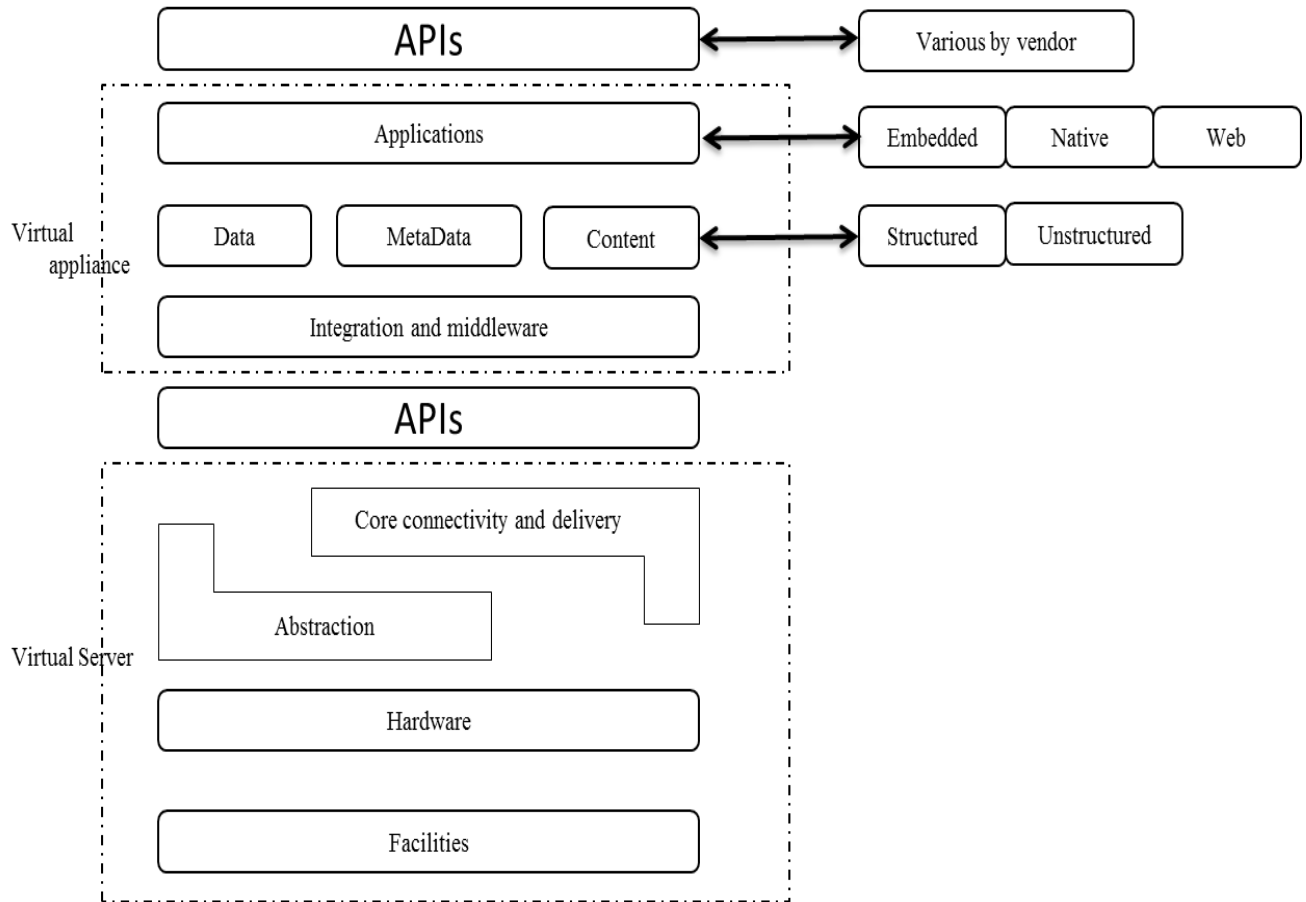


**Figure 3.** Cloud computing stack designated as a server [76]

There are two types of virtual machine monitor (VMM) that can be found in the virtualized resource, namely native hypervisor, such as IBM's CP/CMS, VMware's ESXi and Microsoft's Hyper-V, sometimes called as type 1 or bare-metal hypervisor runs directly on the physical hardware in the most privileged mode and hosted hypervisor, such as VMware's Workstation/Server and Microsoft's Virtual PC/Server, sometimes called type 2 hypervisor, runs directly on a host operating system [77].

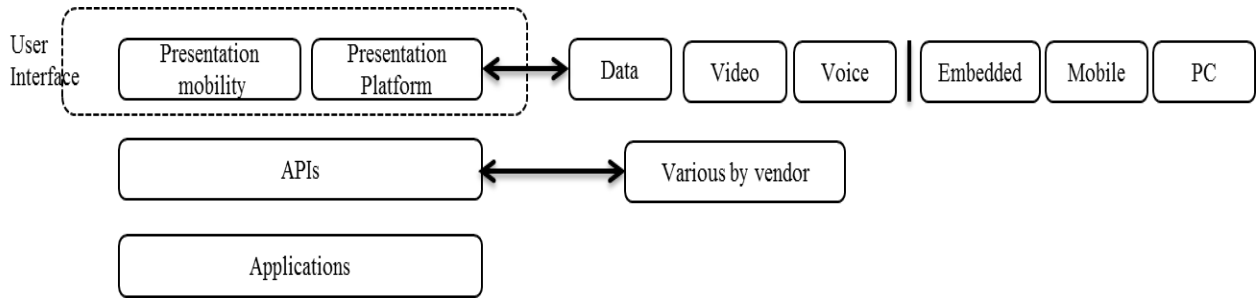
Furthermore, cloud platforms (PaaS), such as Force.dom, Windows Azure and Google Apps/AppEngine are the middleware software layers that provide application developers a flexible environment to develop higher levels of services. Figure 4 shows platform architecture that represents nearly the full cloud software stack, missing only the presentation layer that represents the user interface.





**Figure 4.** Middleware on a virtual machine [71]

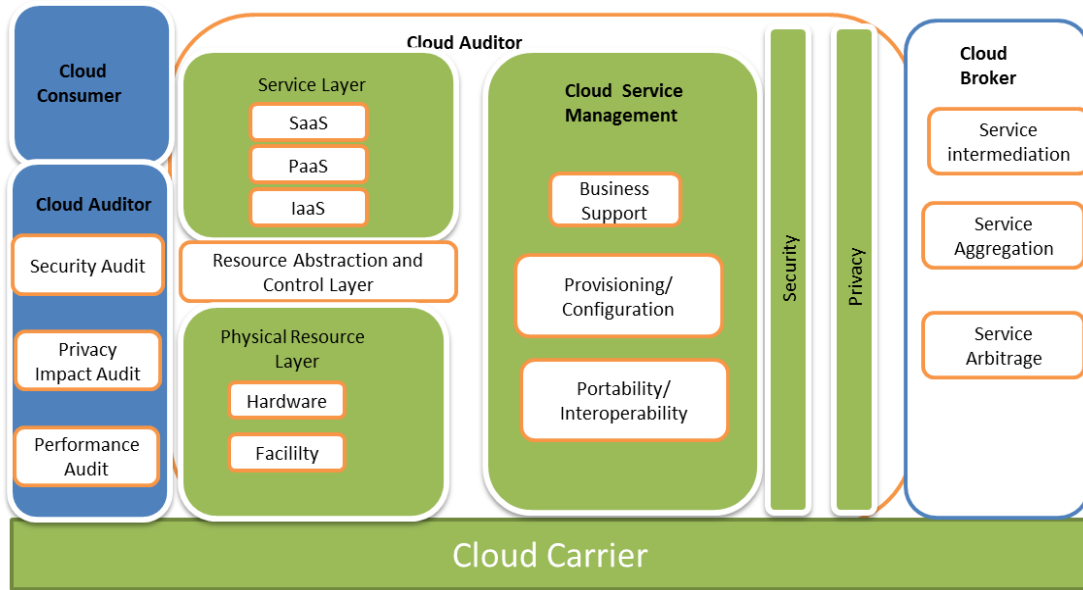
In general, Platforms support developing and testing tools, versioning, local databases and storage tools. These tools are encapsulated in the platform APIs, so that developers may easily interact with the platform services through that APIs. In addition, user interface is not part of the platform any more, but was abstracted from its APIs. Consequently, those services are managed through user interface (UI). Figure 5 shows presentation functionality and API.



**Figure 5.** Cloud computing interface comprises UI and API [71]

Although, Application Programming Interface is useful for communications, security, and data flow control, but they are not standardized as can be seen from Figure 5. Consequently, may in long term cause vendor lock-in.

Furthermore, cloud architecture can couple services running on virtualized physical servers in different locations to offer on-demand services to cloud consumers. According to National Institute of Standard and Technology (NIST), cloud architecture can be divided into five major actors (Figure 6), namely, cloud consumer, cloud provider, cloud carrier, cloud auditor and cloud broker [78]. In addition, NIST defined each actor's role, for instance, cloud consumer defined as a person or organization that maintains a business relationship with, and uses service from cloud providers, which is an entity responsible for making a service available to interested parties. Cloud Auditor conducts independent assessment of cloud service, while cloud broker manages the use, performance and delivery and also negotiates relationships between cloud providers and cloud customers [78].



**Figure 6.** NIST conceptual reference model [78]

Additionally, different services are available for different consumers, for instance, while SaaS consumers may be interested services, including Billing, CRM, Email and Financials, PaaS consumers may demand development, deployment, and testing environments, integration tools, and databases. Furthermore, IaaS consumers may in turn require other services, including computing, storage, and hosting. As Cloud services may sometimes need to be managed for consumers, new cloud entity is needed to handle service usage, management and also negotiation between providers and users. According to Gartner [79], cloud broker provides various services in three service category, namely, service intermediation by providing value-added services to consumers, service aggregation by combining and integrating different services into one or more new service and service arbitrage by selecting services from different agencies according to the quality of available services.

## 2.12 Cloud computing and virtualization

Since cloud computing offers virtualized resources as a service, it is then natural to assume that virtualization technologies are the basic building blocks of cloud infrastructure. In addition, virtualization is the creation of a virtual version of something, such as server, storage device and network resource to allow single physical resource to function as a multiple logical resources as it abstracts compute resources as virtual machines with associated storage and networking connectivity [80].

In addition, security consideration through isolation possibilities of virtual machines, easy disaster recovery and simplification of IT infrastructure management are all virtualization benefits. Furthermore, virtualization may facilitate new software prototypes, testing and efficient delivery, while performance can be monitored in real time and resources can be allocated on demand, thus gaining improved performance.

In general, Cloud computing heavily relies on virtualization, where services are built on top of a virtualization layers which may help the service providers to manage the service and offer standardized platform to the users, therefore, virtualization is another key element of cloud computing [81].

Additionally, virtualization technologies promotes green data center by significantly decreasing data center spaces, cooling and energy requirements. However, security on virtualization solution should be better understood since the impact of an attack on host software aka hypervisor could be very serious since it affects guest operating systems and services running on top of them [82].

## **2.13 Types of virtualization**

### **2.13.1 Hardware virtualization**

This type of virtualization concerns only hardware, such as computer platform where it conceals physical features of the platform. In general, hardware virtualization solution allows sharing hardware resources using host software, aka hypervisor which surrounds an operating system and provides the same inputs, outputs, and behavior that would be expected from an actual physical device [83].

Hypervisor may reside between hardware and operating system and its primary task is to intercept and redirect operating system calls without performance penalties. For instance, hypervisor emulates entire hardware resources in order to create hardware emulations, where guest operating system is installed. Whenever, guest operating system calls hardware resources, hypervisor will intercept and redirect it. In general, hardware emulation can be implemented in two ways using hypervisor software. One approach is to install hypervisor on top of operating system which in turn resides on hardware device. As a result, hypervisor traps guest operating system calls and directs to the underlying operating system. In this solution, hypervisor acts as resource coordinator for the guest virtual machines. The second approach for implementation of the hardware emulation is called bare-metal virtualization approach, which is based on installation of hypervisor directly on hardware device without needing any operating systems [84].

What is more, dependence of hypervisor system call translation may also create a situation where hardware features, such as drivers may not be updated. In general, hardware virtualization technology consists of different virtualization types, including server, partial, full, and hardware-assisted virtualization [85].

### **2.13.2 Partial virtualization**

This type of virtualization is based on performing of partial abstraction of the underlying physical hardware resource, so that only specific guest software can be run on it. Unlike full virtualization, entire operation system cannot be executed on partial type, therefore, much easier to be implemented than full virtualization [86].

The basic idea behind this technology is to provide a virtual machine monitor to abstract individual applications and generate a separate view of the host's resources to each process [87]. In other words, since everything in the target environment is not simulated, installed software programs on the guest operating system cannot be run unmodified [88].

### **2.13.3 Process virtualization**

In this virtualization, memory addresses, central processing unit (CPU) and other system resources are virtualized for each running process by operating system (OS). Furthermore, current process context is managed by running operating system which also ensures that each process to get a share of CPU time using scheduling algorithm. Running application sees virtual machine as address space, registers, and instruction set [89].

Moreover, running processes interact with virtual memory and other resources through standard ABI and APIs, while operating system manages the virtualization and multiplexing of resources under the hood [90].

This technology may provide a way to enable runtime adaptation of software by placing virtualization layer between the running application and the underlying operating system, may enable for inspecting and potentially modifying every instruction that executes on the system, including shared library code or dynamically-generated code [91].

#### **2.13.4 System virtualization**

While process virtualization concerned with virtualizing only processes, entire system can be virtualized through hypervisor. As a result, separate virtual systems known as Virtual Machines (VM) are created which are running in isolated mode. In addition, system virtualization or OS virtualization technique is based on virtualizing OS to decouple application from the OS so that individual applications can be run in a virtualized environment [92].

As a result, all resources of physical resources are managed and shared fairly among VMs by hypervisor, while each VM assumes that physical resources are solely owns by it. Furthermore, VM may manage real resources in various ways, such as allocating resources statically for specific VM or resources are dynamically managed through pooling mechanism, where any resources that are not being used are de-allocated and pooled to be used later. As a result, managing resources dynamically may increase system performance. In general, this type of virtualization is very popular among organizations due to cost-saving and flexibility [93].

In general, system virtualization sees VM not as a space address or register, but an actual machine and interfaces with OS [94].

#### **2.13.5 Paravirtualization**

In this virtualization technique, it presents software interface to VM that is not identical but similar to underlying hardware , for instance, Denali systems uses of x86 for running para-virtualized PC operating system [95]. An advantage of para-virtualization is the suitability for the hardware that does not support virtualization. In general, para-virtualization is based on a hypervisor which multiplexes request accesses from guest machines to underlying resources. For instance, open source type-1 hypervisor, known as Xen installs guest operating system on top of hypervisor which is not included any drivers, such as network drivers. In this regard, when guest operating system requests hardware resources, it accesses through another privilege guest called Domain0 which is modified operating system to assist hypervisor for access controlling to the underlying hardware resource [96].

### **2.13.6 Pre-virtualization**

Pre-virtualization technology combines flexibility with performance and security through decoupling guest code usage from hypervisor via intermediary between hypervisor and the guest code by using neutral interface or offline compilation process. Furthermore, pre-virtualization also promotes guest diversity, by enabling quick virtualization of the guest OS [97]. In short, pre-virtualization is annotating operating system automatically for adapting to specific hypervisor at load time.



### **3. MOBILE CLOUD COMPUTING**

#### **3.1 Introduction to mobile cloud computing**

Cloud computing for mobile domain or, rather, Mobile Cloud Computing (MCC) is one of the hottest topics in technology industry. MCC can be defined as reducing mobile limitation by using cloud computing principles to deliver mobile services to the terminals. By uploading intensive data processing and storage to the cloud, MCC applications can offer features such as image and voice recognition that would not be otherwise possible considering the limited computing power, memory, and data storage available on mobile device. Since, services are running in the cloud, MCC solution may in theory eliminate the problem of terminal and OS fragmentations. In this way, service providers will have an access to a broader audience of wide range of terminals. For instance, according to ABI research , “By 2015, more than 240 million business customers will be leveraging cloud computing services through mobile devices, driving revenues of \$5.2 billion”.

#### **3.2 Characteristics of mobile cloud computing**

Mobile cloud computing has a specific characteristics, such as lack of any requirements towards handheld devices, since all intensive computations are taking place in the cloud and elimination of space- and-time-centric access by enabling users to access what they want from the cloud through internet, while extending cloud computing by providing enhanced service availability and by exploiting information about a user’s location, context and network intelligence, thereby considerably improving user experience [98].

In general, MCC may be divided into two different classes. One refers to carrying out data storage and processing outside mobile device. Therefore, mobile devices are used as cloud terminals for accessing and presenting cloud services. The second class refers to a computation solution where data processing and storage are inside of mobile devices. In this class, mobile

devices share their resources and services among other devices in the cloud environment, where each device has subscribed.

### **3.3 Challenges of Mobile Cloud Computing**

In addition, MCC has many challenges, such as network latency due to intrinsic nature and constraints of wireless network and mobile devices [99]. Latency problem can be solved by minimizing distances between application and users, since is the distance that greatly affects latency. Further, other challenges include a wireless connectivity that meets the requirements of Mobile Cloud Computing with respect to scalability, availability, energy- and cost-efficiency [100]. Since, mobile devices often lack of computing power to execute sophisticated security algorithms, it is really difficult to enforce a standardized credential protection mechanism due to the variety of mobile devices [101].

### **3.4 Cloud Computing Features**

Features of cloud computing include virtualization techniques where resources are virtualized and offered as resource pools to be accessed by consumers through web interfaces, high availability, Rapid Elasticity, Measured service, On demand self-service, Everywhere network access and Resource pooling [102] via easily customizable via configuration, powerful computational power and large scale of storage capacity. In this respect, cloud computing may possibly provide high-performance computing capacity and large storage capacity to the clients, such as mobile devices.

Therefore, combining these two features may lessen mobile computing challenges and enhance its computational power via offloading compute intensive applications onto cloud. Therefore, mobile cloud computing can be described as enhancement of mobile computing challenges and constraints through cloud technologies which in turn based on virtualization technologies. According to 2005 study made in more than 15 countries, primary concern for mobile users is the battery life than any other features such as cameras [103].

Consequently, computation intensive applications running on the device may severely shorten battery life and negatively impact user experience. Furthermore, computation-intensive applications could be moved to the cloud computing platform, while mobile device will connect cloud application and present it to the user, thus mobile cloud computing. Therefore, offloading computations may save energy [104].

### **3.5 Mobile cloud services**

Mobile cloud services are not created equal. Some services that related with platform offer application development and testing platform. For instance, several PaaS providers including Force.com, Appcelerator, and PhoneGap provide mobile application development platform as PaaS. In addition, IaaS providers may offer an infrastructure tools which will be used for building mobile SaaS. As a result, cloud computing attributes such as agility may enhance developers' flexibility for developing mobile application in volatile environment.

Other cloud attributes such as elastic scalability will also be essential for mobile applications due to popularity of mobile devices. Finally, cloud resources, such as computing may be properly utilized by mobile devices through intelligent off-loading their computationally demanding tasks onto the cloud, such as image processing, thus increasing not only application responsiveness but also battery life time. In this paper, MCC computing refers to an infrastructure where both the data and storage and the processing happen outside of mobile device from which an application is launched [105].

According to Vissiongain's report, MCC cloud revenues are derived from multiple services, namely, metered payments (email, VOIP, backup, storage, paid music contents), advertising, direct sale of virtual direct revenues through third part providers (IaaS), metered enterprise app service fee (CRM, SFA, ERM), metered app service fee and revenues from enabling technologies [106].

Furthermore, MCC market will worth \$240 billion by 2016 from \$70 billion in 2010 and growth will accelerate from 2013 to 2016 as LTE network become prevalent [107].

In general, MCC solution can either be mobile performance improvement where cloud infrastructure can be used as an augmentation for resource poverty device execution or mobile application such as e-mails applications. For instance, Zhang et al [108] created elastic applications which augmented mobile devices utilizing cloud computing resources. However, augmenting decisions should carefully be done, since offloading trivial tasks is an inefficient. As a result, several MCC related characteristics include that mobile device being thin client due to offloaded heavy stuff to the cloud, seamless data sharing and regionally unbounded.

### **3.6 Latency Challenges**

In general, the main objective behind mobile cloud computing is to provide not only accurate, but also real-time information for the user regardless of location and time, thus, making computing capacity of mobile device unimportant, even feature phone may utilize cloud capacity resources. In this way, cloud computing may eliminate resource poverty stricken mobile devices from hardware limitations. Furthermore, cloud computing provides resources in elastic way while high performance computing capacity may reduce latency, thus enhancing not only quality of service but also responsive time which in turn creates positive user experience. Cloud computing infrastructure, such as Rackspace [109] or EC2 [110], may significantly enhance seamless data access by providing high bandwidth capacity as well as sophisticated data access mechanism and it may radically change the face of mobile computing by letting resource poverty mobile devices to execute resource-intensive applications. However, there are intrinsic mobile device related challenges, such as limited storage capacity, low battery life time as well as small sized display. In addition, another fundamental obstacle is high WAN latency due to lower network bandwidth. Since application users are acutely sensitive to any jitter and delay, high latency may hurt usability by degrading system response, thus creating negative user experience by reducing user's cognitive engagement. As a result, benefit of cloud computing in mobile world may be limited by WAN. Latency problem can be solved by minimizing distances between application and users, since is the distance that greatly affects latency, although LTE may solve this problem, adopting cloudlets may also eliminate this

challenge [111]. As result, Ericsson has teamed up with Akamai in February 2011 for mobile acceleration solutions that will improve MCC latency challenges [112].

### **3.7 Cloudlet Solution**

Alternatively, the distant communication between mobile device and the cloud may shortened by using resource-rich middleware cloudlet [113] where mobile device seamlessly offload to nearby cloudlet as depicted in Figure 7, which is group of high performance computational infrastructure where all significant resource intensive computations, such as speech recognition, augmented reality, and computer vision occur, thus making mobile devices as a thin client. In this way, mobile device user may able to instantiate instance of required software running on nearby cloudlet's virtual machine. Therefore, cloudlet's physical proximity may drastically reduce latency problem caused by one hop connection to the cloud. In this respect, cloudlets can be considered as micro edition of cloud or extended infrastructure of cloud computing. Further, cloudlets are self-managing computing resources with access control mechanism and can be seen as a black box, which can be installed on public places like train station and coffee shops, like Wi-Fi devices. Therefore, cloudlet is widely dispersed and decentralized infrastructure, trusted and high capabilities computer or cluster of computers, connected to the internet [114].

Although cloudlet contains similar cloud services, but there are differences in terms of ownership and delivery channels, for instance, while cloud infrastructure contains both hard and state, cloudlet supports only soft state, such as cached copies of data [115].

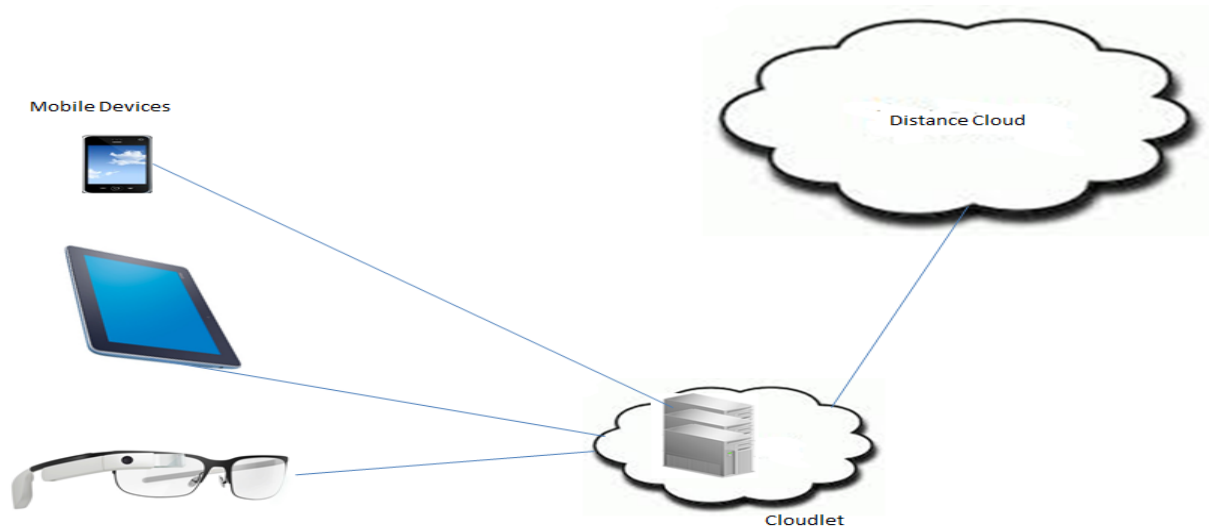


Figure 7: Cloudlet [115]

Furthermore, unlike cloud systems, cloudlet is based on self-management system where ownership is decentralized with LAN based communication network. Alternatively, other solutions can be used to push cloud close to the mobile user. For instance, seamless use of ambient computation of execution engine to augment mobile devices capability by offloading some tasks to nearby computers where they are rapidly executed by using devices' cloned system image and afterwards merging both results in the mobile device's execution may alleviate significant computational burdens from the device.

Therefore, augmented execution solution, which is based on cloning device's system image also known as clonecloud [116] may successfully overcome device's intrinsic hardware limitations. By definition, clonecloud is a service that utilizes cloud infrastructure to offer computing capabilities and other required resources to mobile devices by interacting its clone (copy) running in the cloud. Since battery's life span depends on computational intensiveness, clonecloud may alleviate that burden, thus, transforming resource-poor devices into resource-rich while increasing battery's life time. In this way, fundamental constraints such as short

battery life, latency and limited computation resource, which are intrinsic to mobile devices, may be finally resolved.

Additionally, security aspects in MCC, both in mobile device level and cloud level which is obviously related concurrency, distribution and multi-tenancy may create a real security issue. Furthermore, MCC requires ubiquitous connectivity in order to satisfy mobile cloud computing with respect to scalability, availability, energy and cost-efficiency [117].

### **3.8 Privacy challenges**

Other challenges include privacy and security, since device can easily be lost. In this way, mobile device may amplify aforementioned risks. In addition, security in all cloud models whether SaaS, IaaS, or PaaS, especially services that handle sensitive data may also exacerbate security risks in mobile devices.

Finally, context-, location-aware services that collect and aggregate personal data including user's location at certain time may pose severe privacy challenge.

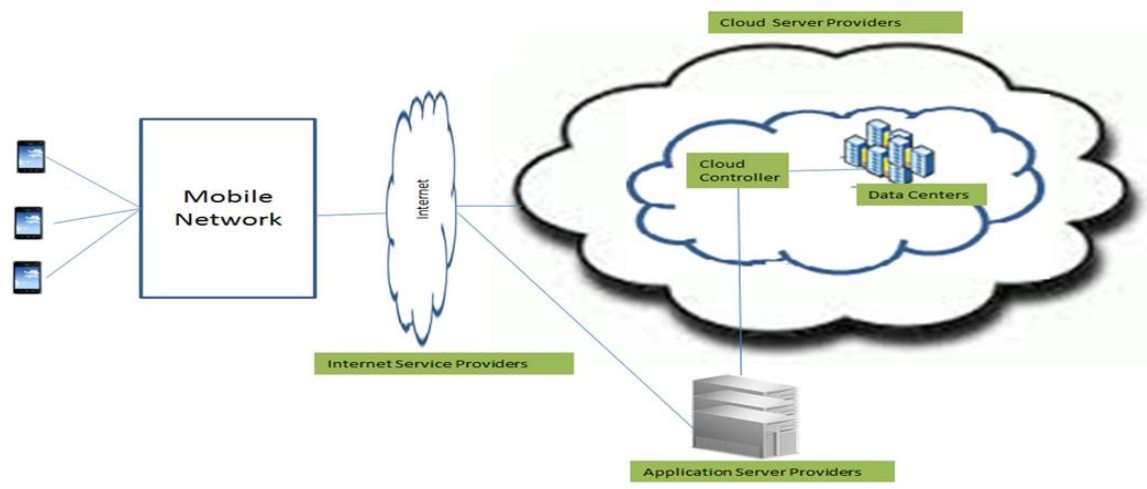
In addition, cloud related security issues include preserving data integrity [118] and [119] and protecting it both in transit and at rest state, digital right management and solid authentication based access management.

.

### **3.9 MCC Architecture**

As depicted in Figure 8, mobile devices, such as mobile phones or tablets are connected to the mobile networks where mobile users' requests and information are transmitted to the central processors and mobile network operators may be able to provide services to mobile users such as authentications based on the home agent (HA) and also subscribers' data stored in mobile operator's databases. In addition, mobile subscribers' requests are delivered to a cloud through the Internet, where cloud controllers process the requests to provide mobile users with the

corresponding cloud services [120].



**Figure 8:** Mobile Cloud Computing Architecture [120]



## **4. CROSS PLATFORM SOLUTION FOR CLOUD COMPUTING APPLICATIONS**

Dealing with multiple mobile platforms known as a fragmentation is a real challenge of mobile development [121]. In addition, different mobile platforms come with different languages, tools and distribution channels. In this way, mobile application that was developed on particular platform may not run on another platform. As a result, application developers need to master multiple programming languages and platforms for multiple devices. Furthermore, each development project targeting specific set of devices will be different than other projects, in terms of requirement management, analysis, design, implementation, test, and configuration management. Even though, mobile web applications (MWA) may solve fragmentation nightmare in terms of portability challenges, but mobile application requirement may inhibit web solution as a viable choice.

### **4.1 Cross Platform Frameworks (CPF)**

Cross platform frameworks (CPF) were introduced in to the market, where mobile application developers would only concentrate business logic of the application rather than portability issues. In general, Cross-platform mobile frameworks may enable mobile application developers to create resident mobile application (RMAs) that can be seamlessly deployed to different mobile platforms. Quality of these RMA applications developed with cross-platform framework can be as high as native SDKs. Furthermore, Cross-platform frameworks due to expansive APIs provide deep integration with device's resources, such as camera, GPS, and accelerometer. As a result, Mobile application developers seamlessly interact with APIs provided by cross-platform framework, while supported APIs can possibly be extended if required features are not supported. Even though, several cross-platform frameworks exist in market, only some frameworks may leverage other frameworks, such as web mobile framework and native SDKs for constructing application UI and building, respectively. These frameworks

leverage different frameworks to generate deliverable RMAs, hence called hybrid frameworks, such as Oracle MAF [122].

In addition, as the name suggests, mobile frameworks leverage WebView controls for rendering UI applications using latest web related technologies, such as CSS3, JavaScript, and HTML5 and also accessible from any mobile device that supports web browser. In general, mobile frameworks are best suitable when application portability is concerned, while other advantages include simpler maintainability, reduced learning curve due to familiarity of web technologies.

Mobile framework may even leverage other programming sources, such as C# [123], Java [124], and even Lua [125], which consequently reduce the need for new language and increase time to market. By using WebView controls, mobile application developers are capable of accessing several device resources, for example, one may access hardware sensors on Android devices [126].

However, mobile framework's slow performance due to external functionalities, such as JavaScript, may create a significant drawback that may have a negative impact on user experience.

Generally, using cross-platform frameworks, developers are capable of creating portable native mobile application, also called resident mobile application (RMA). However, CPF are not created equal and varies in terms of portability issues, for instance, some CPF combine web technologies with native code to generate RMAs, while others may use proprietary technologies.

Platforms that combine web with natives are also called hybrid frameworks and support state-of-art web technologies, such as CSS3, HTML5 and JavaScript, while developers may leverage native webview control or browser controller, which is part of many native SDKs.

Furthermore, JavaScript is generally used for accessing underlying device resources. Even though, mobile application developers can use hybrid framework for creating portable RMAs for different target audience, some frameworks are only suitable for certain targets, for instance, PhoneGap is primarily suitable for public mobile users, whereas Sybase Unwired platform is intended for developing enterprise mobile applications [127].

In addition, other frameworks only use proprietary source language such as Lua for Corona framework [128] or ActionScript for Flex framework [129]. Finally, some frameworks such as Titanium Mobile [130] provide an API that is mapped to the target platform native API in such a way that mobile developers may use any familiar source languages they may deem.

## **4.2 Cross platform framework for solving fragmentation challenges**

Cross platform framework solved many fragmentation challenges and drastically alleviated developer's burden. However, cross platform frameworks still depend on native tools, for instance, building and compiling. In addition, emulators and/or simulators may also be needed for testing generated RMAs. These testing tools in turn require accessing native SDK. As a result, some framework vendors attempted to mitigate this dependencies and created remarkable solution by offering cloud for building and compiling RMAs. For instance, PhoneGap vendor Nitobi provided a build cloud [131] for building and compiling mobile applications built with web technologies. In general, cross platform frameworks enhances user experience of the mobile applications. In some cases, frameworks are capable of utilizing native experiences, such as GPS, camera, and compass to deliver expected user experiences. Furthermore, framework may render user interface for application either through abstraction or emulation. With abstraction, framework uses underlying UI controls of the native, while in emulation case, framework attempts to independently construct UI by mimicking underlying UI control. Consequently, abstraction method increases UI richness while emulation decreases UI quality. Therefore, frameworks with expansive APIs support and leverage proprietary language such as Corona or Titanium Mobile use abstract model for constructing UI. In contrast, frameworks that leverage web technologies use emulation model for constructing UI [132].

In addition, even though almost all CPF have capabilities of accessing device features, but some frameworks, such as Corona, which supports expansive APIs provide deep access to the native

capabilities. In general, frameworks that use proprietary provide greater support than others, such as hybrid frameworks. Unlike WMA, CPF enables application developers to seamlessly access device features, such as camera, accelerometer. Additionally, almost all frameworks have capabilities of offering developers to create mobile applications that may address application requirements. According to Gartner [133], framework capabilities fit in to three categories, namely, tooling, execution and application programming interface. Tooling part may include programming language, use interface builder, development platforms, simulator, profiler, SDK management, and build [133].

Platform execution capabilities may include runtime engine that may possibly support several runtime platforms for potential broader reach. In addition, execution capabilities may also include platform's runtime container that encapsulates underlying platform complexities from mobile application. Furthermore, runtime container may offer other capabilities, such as data synchronization or progressive enhancement [133].

For progressive enhancement capabilities, application developers are offered the luxury of focusing on application's business logic, while framework will handle rendering user interfaces for different mobile devices. While performance may have an impact on user experience, data synchronization capabilities may be enhanced into resource-aware capabilities by configuring available framework to conduct synchronization process based on availability of high network bandwidth. Generally, developer's challenges are not based on platform availabilities, but selection of type of platform that is suitable for fulfilling current requirements. For instance, selection criteria may be affected by performance requirements, device resources, such as camera, GPS, on/offline capabilities, data synchronization, application portability, discoverabilities as well as time to market. For instance, native RMAs which can only execute specific code that was compiled on specific platform may have better performance due to running directly atop of the mobile platform than hybrid platform. Therefore, advantages of native RMAs include maximum performance, providing native SDKs that enable seamless access to the device resources. However, code compiled with this platform cannot be reused for other platforms, which means specific codebase for specific platform, thus increasing platform fragmentation.

### 4.3 Mobile Web Apps

Mobile web apps refer to internet-connected applications that can be accessed with mobile devices, such as mobile phones and tablets using browsers. In addition, mobile web application development allows developers to escape fragmentation challenges in mobile devices to reach widespread audience.

In this way, application development will be independent of the fragmented mobile platforms, thus, saving time and resources. Furthermore, mobile web is easier to develop and maintain than native apps which depend on device platform and its regular updates. Mobile web apps generally leverage latest web technologies including family of HTML 5 technologies to deliver richer web UX experience. In addition, mobile web apps may use JavaScript in the browser and can communicate server-side process through RESTful interfaces or web services.

Even though, mobile web applications are suitable for portability across variety of mobile devices, browser discrepancies among devices must be addressed. For instance, several browsers are incapable of constructing pages that were built by using advanced web technologies, like html5. As a result, several mobile web development frameworks recently emerged into the market, including JQTouch, Sencha Touch, and JQuery Mobile.

Furthermore, browser fragmentation will unfortunately exist, thus, complicating future MWA application development. In addition, device fragmentation may also exist in device capabilities, such as resolution and screen sizes which cannot be updated. As a result, MWA application developers must accommodate these issues. MWA can be upgraded easily and both portability and application manageability is quite high, thus enabling broader reach and reduction of time to market.

However, MWA has a significant drawback, including access limitation of device resources and slow performance due to heavy usage of JavaScript. In addition, several solutions have been introduced to mitigate fragmentation challenges among device browsers. For instance, developers can leverage media queries in CSS to mitigate browser fragmentation, specially

screen sizes and resolutions. Some solutions are based on design strategies rather than framework based solution, such as progressive enhancement techniques which enables developers to design mobile applications easily accessible through browsers that don't support latest technologies. In this design strategy, developers will be able to proactively customize features to be added into the application. For instance, mobile browsers that support latest web technologies will be added more features than other browsers. Additionally, application developers may also be able to detect whether specific features are supported in target browser through using JavaScript.

#### **4.4 Cross Platform Mobile Tools and Technologies**

Mobile application development for various mobile devices is a tedious and daunting task due to architectural fragmentations among mobile devices. Furthermore, several frameworks have been designed to neutralize fragmentation challenges to alleviate developer's pain. In addition, some considerations should be exercised prior to tool selection, for instance, mobile application developers should be able to trade safety based on application commerciality of selecting native SDKs for less productivity due to supporting specific platform.

Other issues to be carefully considered include, supported features, cost affordability, commercial viability, accessibilities to necessary APIs, possibility of extension, addressability of application development life cycle, such as testing, debugging and building, learning curve, vendor's customer orientation, and lastly platform ecosystem. In general, almost all mobile application platform frameworks intended for developing hybrid apps support latest web technologies, such as HTML 5.

## 4.5 HTML 5

HTML 5 consists of family of web technologies comprises of HTML, CSS and JavaScript and is being standardized by W3C and Web Hypertext Application Technology Working Group (WHATWG). In addition, HTML5 is the fifth revision of html language standard for structuring and presenting contents for the web. Further, html5 comprises of a collection of technologies, APIs and features, including CSS3 and series of JavaScript APIs to enable creation of sophisticated and complex web application. According to Juniper Research, the number of cloud computing subscribers is expected to grow rapidly in the next five years as the cloud-based mobile market will generate annual revenue of \$9.5 billion in 2014 from \$400 million in 2009 [134]. Furthermore, by 2016, more than 2.1 billion mobile devices will have HTML5 browsers, up from just 109 million in 2010 [135]. Moreover, cloud-based apps will not only offer an improved offline usage through data caching where disconnection from internet won't interrupt apps usage, but also reduce server load demands which will make cloud-based mobile connectivity as a means to improve access of locales with poor coverage [136].

Additionally, HTML5 technology is already strongly emerging in internet, such as video sites, like YouTube's video format H.264, which can be viewed by browsers that support HTML 5 or Gears project by Google replacement to HTML 5.

Mobile Application developers who adopted HTML 5 technology may be enabled not only seamlessly port their apps to several devices, but may also bypass opaque processes and constraints imposed by app stores.

Originally, the web was originally designed to support static hypertext documents, then, web browser plug-ins, such as applets, flash, were created to support dynamic applications. However, more browser limitations were discovered, where HTML 5 capabilities may address, for instance, current browser limitations include peer-to-peer communications or performance issue in client/server communication, push notifications as well as local storage for offline usage and high quality graphic support. In addition, HTML 5 canvas feature enables application developer to easily embed external multimedia, such as video, images directly in to the application. Further, apps developed in HTML 5 allow users to launch web pages in offline

mode and can store files and other information in the documents on the mobile devices. In addition, HTML5 standard supports Canvas element combined with API which supports both 2D and 3D drawing elements. Like Canvas element, CSS3 also offers graphic capabilities but with better performance. In addition, HTML5 also came with new other features, such as Offline Storage, Semantics, Audio/video, Application Cache, Automatic SSE, and Geolocation.

**Offline Storage:** Multiple new APIs were introduced in HTML5 spec to address offline storage challenge. WebStorage [137] API is based on key-value pairs where localStorage object provides a Storage Object for an origin. For instance, storing current movie under a key “currentMovie”, and giving value “movieState”, localStorage will provide storage object, localStorage.setItem(“currentMovie”, “movieState”). In addition, by loading back the value, localStorage.getItem(“currentMovie”). Furthermore, sessionStorage also stores data on the client side. The difference between sessionStorage and localStorage is that while sessionStorage stores data for one session, localStorage object stores it with no expiration date. In addition, offline storage supports other APIs such as WebDatabase which can be used to manipulate client-side databases using SQL [138], IndexedDB which performs advanced key-value data management by using transactional databases to store keys and corresponding values by implementing through B-tree data structures [139] and FileSystem API [140] for creating folders and storing potentially huge files.

**Semantics:** prior to HTML5, structuring pages such as footers, headers, columns or any other sections had been defined with div elements, for instance, <div id=“footer”>. Fortunately, HTML5 introduced new elements to represent different sections in pages. For instance, one can use <header> or <footer> instead of <div> element.

**Audio/video:** HTML5 introduced new media elements [141] which facilitate embedding video/audio elements in standard way on a web page for playing videos or movies and audio files with captions. For instance, embedding video is a straight forward as

```
<video width="400" height="300" controls="controls">
```

```
<source src="mymovie.ogg" type="video/ogg" />
```

```
<source src="mymovie.mp4" type="video/mp4" />
```



```
<source src="mymovie.mpg" type="video/mpg" />
```

video tag not supported in your browser!!!

```
</video>
```

<video> element supports more than one <source> element which can link to different video formats.

**Application Cache:** HTML5 introduced a way to cache web application to be accessed when there is no internet connection. Moreover, Application Cache comprises a set of cached resources consists of one or more resources identified by URLs falling into one or more following categories master entries, the manifest, Explicit entries and Fallback entries [142]. In addition, application cache gives web application three benefits, speed and performance, offline browsing and reduced server load.

**Server-Sent-Event (SSE):** HTML5 also introduced SSE to enable web servers to push data to the client over HTTP by creating an *EventSource object* and registering an event Listener. In this way, web client will never need to regularly poll for updates, since any updates will pushed to it. For instance, twitter, news feeds, sport results updates will automatically be pushed to web client. There are some other useful APIs such as *PageVisibility API*, *Battery Status API*, and *getUserMedia API*. In the case of Pagevisibility API, it is application programming interface which enables web application to determine the visibility of the current page. It is defined in HTML 5 specification that visibilityState attribute returns one of these four states, *hidden*, *visible*, *prerender* or *unloaded* [143].

```
partial interface Document {
    readonly attribute boolean hidden;
    readonly attribute DOMString visibilityState;
    // "hidden", "visible", "prerender", "unloaded"
};
```

Even though Battery API scope is very narrow, it is an important API. With this API, developers are enabled to determine the status of the battery of the mobile device and proactively conduct desired decision to improve user experience rather than making assumptions. In addition, JavaScript getUserMedia API provides application developers to

access underlying device capabilities, such as camera, microphone, through calling rather than installing plugins. Since HTML 5 is distributed through device browser, application *discoverability is rather high. For instance, almost all browsers in the market support HTML 5* capabilities, even though browser fragmentation may preclude web apps to seamlessly run across different browsers.

However, emerging HTML 5 technology had already had a substantial impact on existing web application providers. For instance, Amazon's kindle format 8, which is principal format for electronic books, supports HTML 5 [144], adobe's acquisition of Nitobi software which developed PhoneGap and abandoning Flash on mobile platform may also promote HTML5 domination in application platform markets. Furthermore, Financial Times's FT application was developed with HTML5 and within 10 months passes 2m end users [145], thus proving the penetration capacity of this technology. In addition, Microsoft Corporation has also demonstrated windows 8 apps developed with HTML 5 and JavaScript to access the full power of the PC [146]. What is more, in Nov. 2011 HTML5 platform creator, called Strobe, was acquired by Facebook [147]. Therefore, all these acquisitions and demonstrations may prove that HTML5 technology family will dominate web application platforms, including mobile platforms. As a result, several mobile platforms already emerged into the market, including PhoneGap, Verivo and so on.

## **5. CRITERIA FOR EVALUATION OF CLOUD CONNECTIVITY OF MOBILE DEVELOPMENT PLATFORMS**

The objective for this section is to set criteria for evaluating cloud connectivity of cross-platform development. Evaluation of these criteria will be based on testing.

These criteria include:

1. Defined as a cross platform development framework
2. Access to cloud in development Phase
3. Collaboration between developers through Cloud
4. Testing application in the cloud
5. Building and deployment via cloud
6. Distribution to Apps Store via cloud

Defined as a cross platform development framework criterion is a bench mark for the selection where development tool is identified as a cross platform framework. In this selection, mobile development tool is analyzed whether it supports multiple platforms. In order to be selected, mobile application development platform should support at least more than two platforms.

Access to cloud in development Phase criterion covers cloud connectivity of the cross platform, especially in the development phase, where required APIs are easily available in the cloud and development can be implemented on the cloud.

Collaboration between developers through cloud criterion examines whether mobile developers can easily collaborate development through cloud. For instance, developers on different locations can cohesively work on development of mobile application.

Testing application in the cloud criterion covers whether implemented mobile application can be tested on the cloud and whether testing on cloud can be done using different mobile platforms.

Building and deployment via cloud criterion covers whether implemented mobile application can easily be built and deployed to the cloud.

Distribution to Apps Store via cloud criterion, distributions of applications to Apps Stores via cloud are examined and whether modification and upgrade can easily be done through cloud.

## 6. SELECTION OF CROSS-PLATFORM TOOLS

Selection of cross-platform tools are primarily based on cloud connectivity support of the tool. Even though, there are dozens of mobile application development tools that can be defined as cross-platform, but they may not necessarily support cloud connectivity, thus unselected to this research. Further, another reason for selecting these tools are based on usability and flexibility of the tool as well community behind it. In addition, the nature of cloud connectivity support is examined and rigorous analysis have been made which was based on both available literature as well as hands-on testing.

### 6.1 PhoneGap

PhoneGap is an HTML5 application platform that leverages web technologies, such as HTML, CSS and JavaScript to be used as a developing language for mobile native applications. Furthermore, PhoneGap is an open source which can be created mobile applications that can run on multiple devices using HTML5 for the layout, JavaScript for accessing device functionality, such as, storage and camera, and CSS3 for rich look and feel without changing the source base. As result, PhoneGap may bridge the gap between web and native technologies. Moreover, mobile applications created with PhoneGap framework can access device hardware, such as Accelerometer, GPS, Vibrations, Sound and Geolocation through SDK API which abstracts native device platform to facilitate cross platform deployment. PhoneGap supports multiple mobile platforms, such as iOS, Android, Windows Phone, BlackBerry, WebOs, Symbian and Bada [148].

As a result, PhoneGap uses different SDKs for different mobile devices. For instance, iPhone application development will be needed iPhone SDK, Xcode, Mac Os and Intel-based Apple Computer, while Android applications need Android SDK or (NDK) and IDE environment like Eclipse IDE. Furthermore, application developers can seamlessly interact with PhoneGap's APIs to programmatically access underlying device capabilities. For instance, Accelerometer API will enable application developers to access into the device's motion sensor, while Camera

and Capture API may enable possibility of capturing photo and media files, such as video, audio, using mobile devices capabilities. Another important API, Storage API, which allows access to the storage of the devices, is not based only on W3C Web Storage API, but also W3C Web SQL Database specifications [149].

Furthermore, Geolocation API, which is based on W3C Geolocation API specification, offers location information, such as longitude and latitude for the mobile devices using several sources, such as GPS GSM/CDMA cell IDs. Even though, several devices, such as BlackBerry, Windows Phone, Android, WebOs, Bada and Tizen already implemented Geolocation API Specification, other devices may leverage PhoneGap implementation which is closely adheres to the specification of W3C. In addition, Notification API which handles device notification whether audible, visual and/or tactile provides notifications to mobile application developers. In general, methods in this API include *alert*, *beep*, *confirm* and *vibrate*. For the notification alert function, Cordova implementations intrinsically use native alert dialog box, but other platforms may alternatively use the less customizable browser's alert function.

PhoneGap platform does not support write once, run everywhere concept, therefore, developing mobile application on PhoneGap application framework may require different native SDK for different mobile devices for compiling and testing. However, PhoneGap addressed this inconvenience by implementing cloud build [150] service where SDKs, compilers and hardware are not needed at all. Suffice developed applications to be uploaded to the build service and compiled code will run on different mobile devices. PhoneGap build also supports multiple platforms including Android, iOS, Windows Phone, Blackberry and Symbian. In order to compile and test, applications should be packaged according to open packaging model which is based on W3C widget Packaging specifications [151].

Required file types, such as html, css, js, and images are then uploaded onto the Build either as zip format or via git repository as shown in Figure 9.

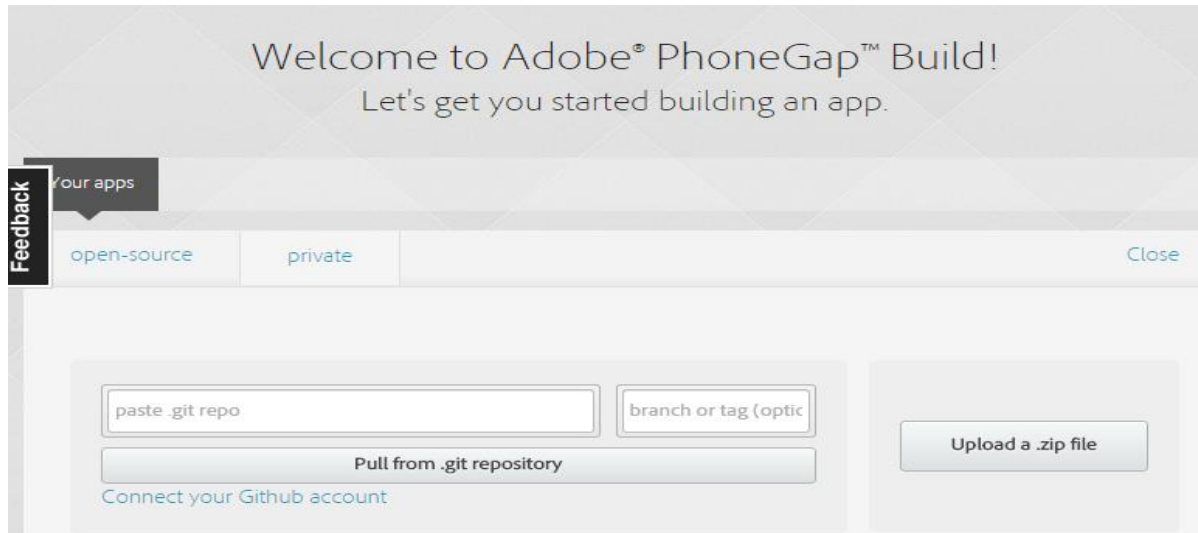


Figure 9: PhoneGap solution for building mobile applications on Cloud [152]

Since PhoneGap Build enables creating cross-platform mobile apps, any native files, like .java should not be uploaded. Furthermore, PhoneGap Build provides an API which is designed for integrating into familiar IDEs, app builders and allows to programmatically access to not only creating and building, but also updating PhoneGap applications using PhoneGap Build web service [153]. In addition, PhoneGap Build also provides a tool called Hydration, which notifies any updates by pushing directly to PhoneGap apps installed on device [154]. In this way, any updates made by app developers will instantly be notified to the testers of the application. Further, PhoneGap build enables developers to easily collaborate application development [155] where remote application developers can modify previous implemented application code and testers can only download.

## 6.2 Rhomobile

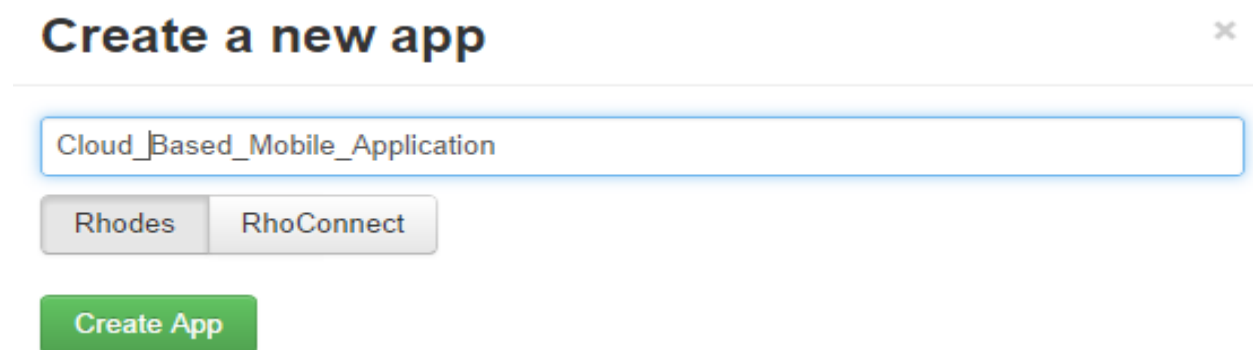
According to Motorola, Rhomobile suite, which comprises of RhoConnect, RhoStudio, RhoElements, Rhodes and RhoGallery allows creation of hardware and OS agnostic applications that can be run on any mobile device. In addition, Rhomobile supports enterprise-

and consumer-class operating systems in Windows Embedded Handheld, Windows CE, Windows Phone 7 Series, Apple iOS, Android and BlackBerry [156].

Consequently, one may have a complete control over how mobile applications may be deployed and behaved in different mobile devices. Since Rhomobile works as a mobile application container, developers with common skills can easily write cross platform code. Further, RhoMobile application container also provides API libraries for accessing device level capabilities, such as GPS, camera and so on via JavaScript and Ruby interfaces. In addition, RhoMobile container provides cloud platform called RhoHub which integrates with RhoStudio- Eclipse plug-in- for not only coding but also seamlessly building and running in the cloud. Further, RhoMobile enables users to create either Rhodes mobile apps or RhoConnect apps for mobile application integrations as depicted in Figure 9 and 10.

What is more, RhoMobile solution also enables developers to collaborate easily by adding collaborator name and her role.

Additionally, based on target customers, developers can use different development platforms for different applications. For instance, RhoElement is a cross platform framework for developing enterprise applications while Rhodes cross platform is suitable for developing consumer applications.



**Create a new app** ×

Cloud\_Based\_Mobile\_Application

Rhodes RhoConnect

Create App

Figure 9: new application creation using RhoMobile cloud solution [157]



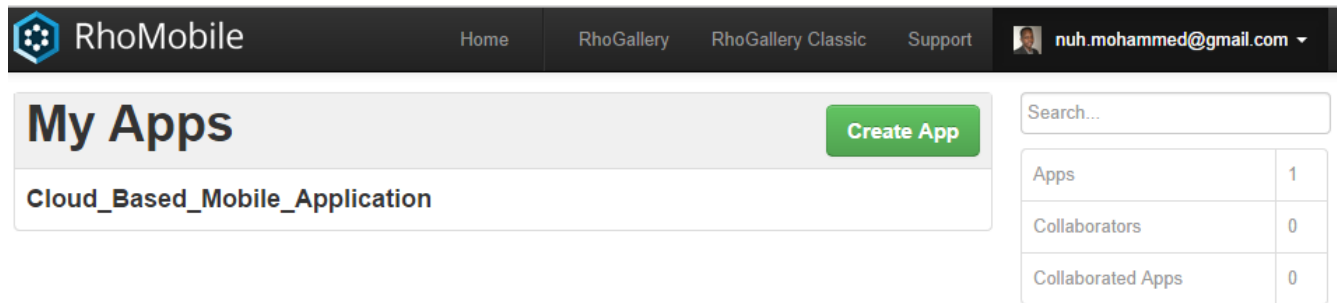


Figure 10: new application created using RhoMobile cloud solution [157]

### 6.3 MoSync

MoSync is a cross-platform solution for developing mobile applications. It consists of two different solutions, namely MySync Reload which can be developed native mobile application using JavaScript and HTML5. In addition, developers using this tool can easily access advanced features of the device, such as sensors, camera, location, file systems and so on. In this tool, one may develop mobile native applications for several operating systems including iOS, Android and Windows Phone 7 using familiar HTML/JavaScript technologies [158].

Further, MoSync SDK is another tool for rich cross-platform application environment which also enables mobile applications developers to easily code, build, and compile for up to nine different platforms at once, using C/C++ or HTML5/JavaScript [159].

In addition, mobile developers using MoSync SDK may be able to get access native device features, such as sensors, locations, file systems, and so on by just using set of APIs. For instance, getting current location, one may easily call `getCurrentPosition` method on geolocation object from integrated JavaScript API called Warmhole, which provides location related information, such as longitude and latitude. Since, SDK tool is tightly integrated with the MySync Reload, one may develop mobile native apps for several platforms using web technologies. Additionally, there is another C/C++ API for SDK developers, who may develop native mobile applications using C/C++.

MoSync does not provide cloud solution for application development by itself, but developers can use Cloudbase.io's helper class for MoSync cross-platform SDK [160].

For instance, mobile application developers may first sign up using cloudbase.io portal and then start creating mobile apps by giving application name and the password for that application Figure 11 and then, cloudbase.io will generate a unique code for that application Figure 12. Additionally, developers may add more functionality code, such as cloud functions [161] where application code can access Cloud Database to seamlessly manipulate and manage.

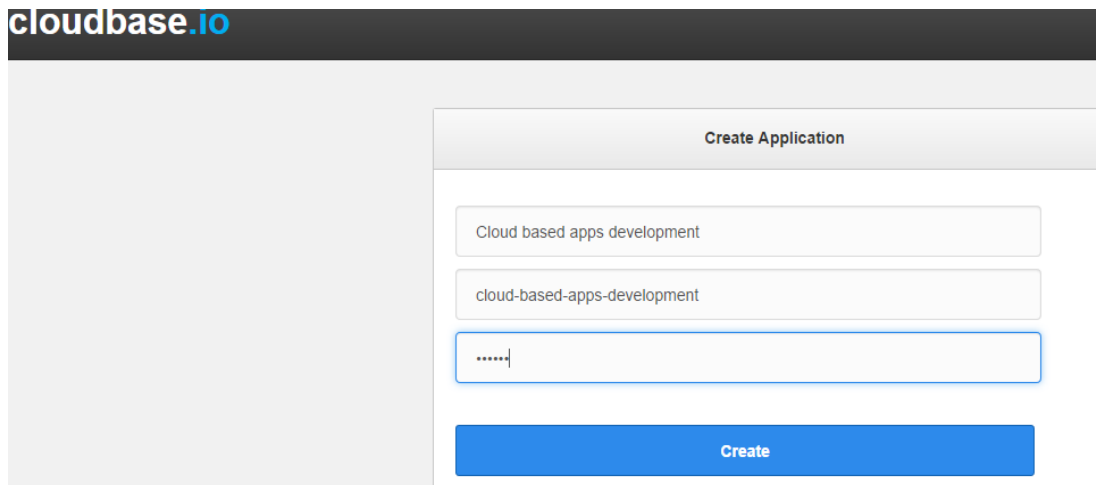
The image shows a web interface for Cloudbase.io. At the top left is the 'cloudbase.io' logo. Below it, on the right side, is a 'Create Application' form. The form has a title bar 'Create Application'. It contains three input fields: the first is labeled 'Cloud based apps development', the second is labeled 'cloud-based-apps-development', and the third is a password field with a masked input '.....'. Below these fields is a blue button labeled 'Create'.

Figure 11. New Application creation using Cloudbase.io

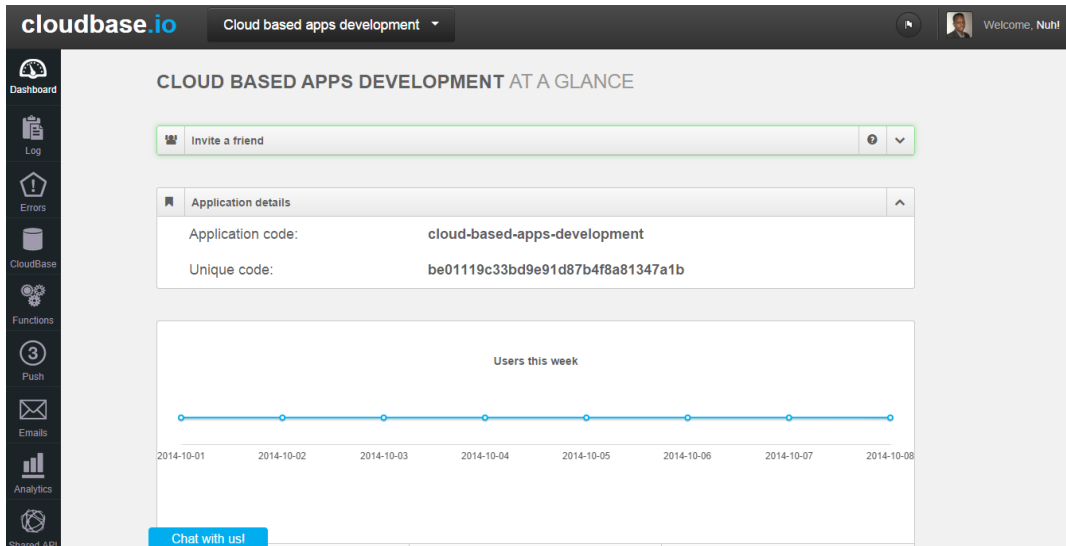


Figure 12. Cloudbase.io generates unique code

Developers using MoSync platform for mobile application development can easily access Cloudbase.io services via helper class provided by Cloudbase.io. MoSync mobile developers may easily get those classes from github and if android helper is needed, then, it may be got from cloudbase.io as a jar file [162]. Developers can programmatically access cloudbase.io services via helper class. For instance, Android mobile application developers can get cloudbase.io helper class, import and initialize a new helper object as depicted in the following code.

```
import com.cloudbase.helper.*;
CBHelper NuhHelperObejct = new CBHelper(
    "cloud-based-apps-development",
    "be01119c33bd9e91d87b4f8a81347a1b",
    this.getActivity());

NuhHelperObejct.setPassword(md5("myPassword"));
```

After initialization, developer may be able to add more functionality, such as sending notifications, sending email, collecting statistics of application usage, store objects to the cloud database or even interacting with PayPal services.

## 6.4 Xamarin

Using Xamarin platform, one may be able to develop native iOS, Windows, and Android applications using one single language, C#. Further, Xamarin apps are built with standard, native user interface control and can easily access spectrum of functionality exposed by underlying platform and device [163].

Moreover, Xamarin framework solution provides code sharing and code reuse that may reduce time-to-market. Further, developers using Xamarin platform may entirely code mobile application with C# while accessing other platform's APIs and may deliver device specific user interface where large portion of application logic code can be shared across device platforms as depicted in (Figure 13).

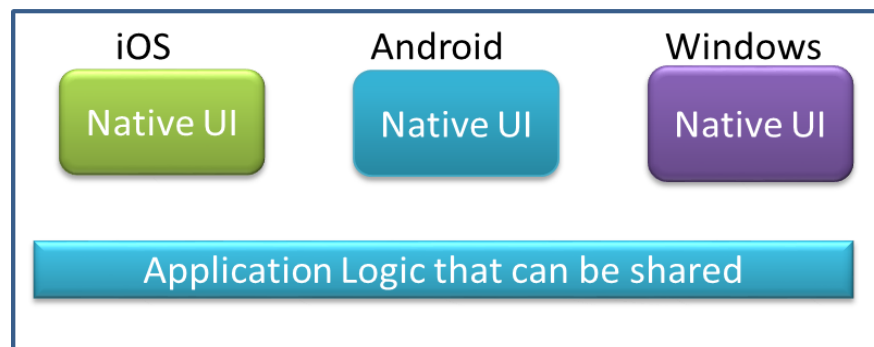


Figure 13. Xamarin device specific UI with shareable application logic [164]

Developers using Xamarin may easily develop native user interfaces while utilizing platform-specific elements and at the same time may be able to share their specific codes, including DB communication integrations, and web service implementations.

Further, Xamarin Platform consists of different components including component store, which are searchable components both paid and free that consists of third part libraries, native UI controls, development IDEs, native compilations, SDK bindings for different platforms[165] and so on (Figure 14 )

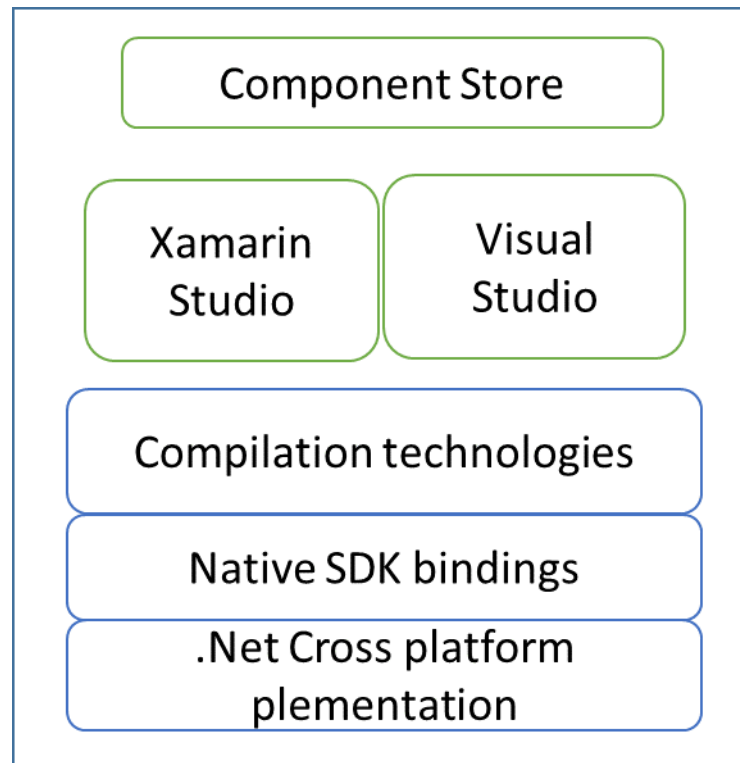


Figure 14: Xamarin platform [165]

Furthermore, Xamarin platform also include development IDEs with extension of component catalogs where developers may easily search desired component to embed into the code.

In addition, developers may easily test their mobile applications on several mobile devices using Xamarin test cloud [166], which is cloud based service, which provides an easier way to do UI acceptance automatically. Further, testers can focus on testing rather than maintaining testing environment.

In general, locally created mobile application and tests are uploaded to Test Cloud, which starts installation of mobile apps and run tests against hundreds of physical devices and notify results to the developer as depicted in Figure 15.

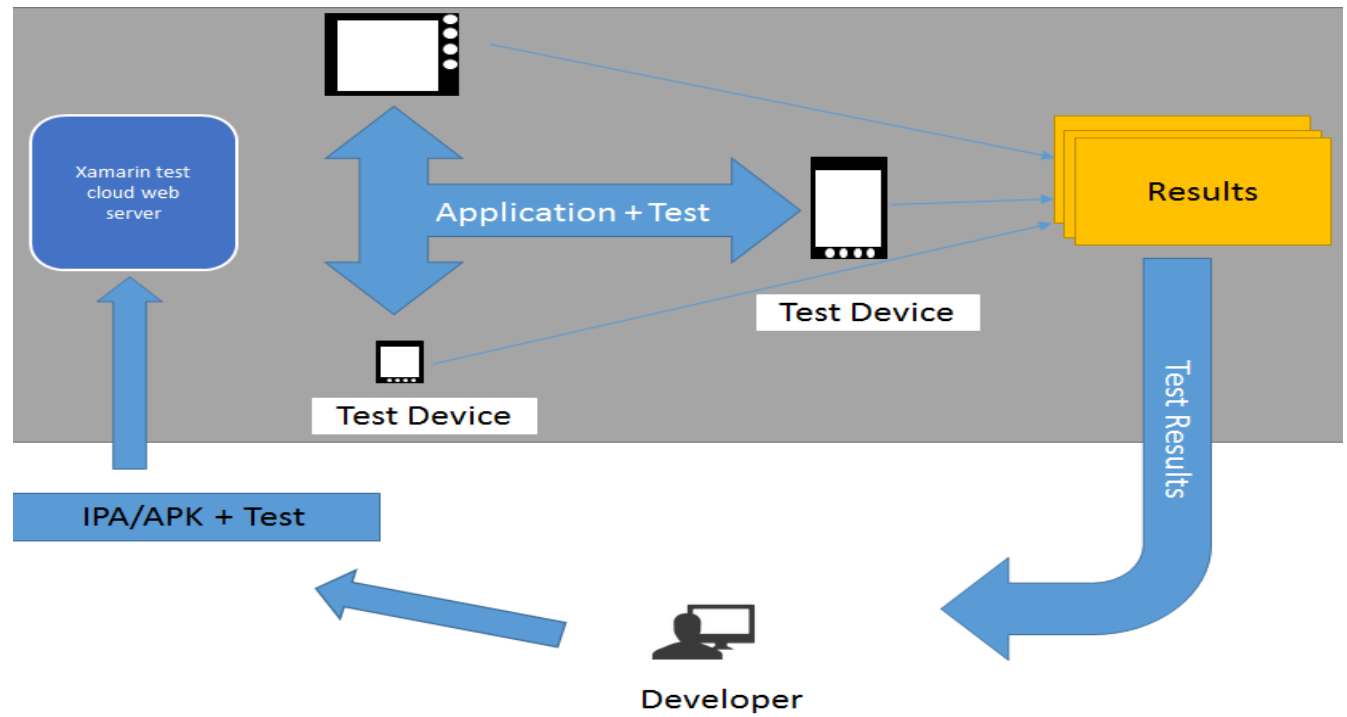


Figure 15. Xamarin cloud testing lifecycle [167]

In addition, in order to test using Xamarin test cloud environment, one must first register and sign in to the system as depicted in figure 16.

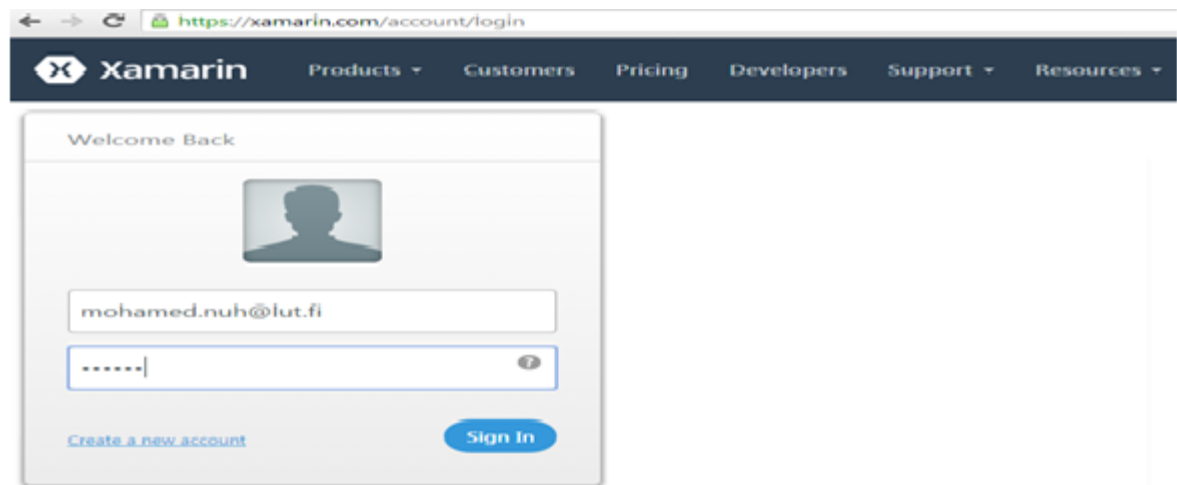


Figure 16. Xamarin cloud testing login page

After registration into the system, developers may able to run application tests against more than 1000 devices [168] as shown in the following Figure 17.

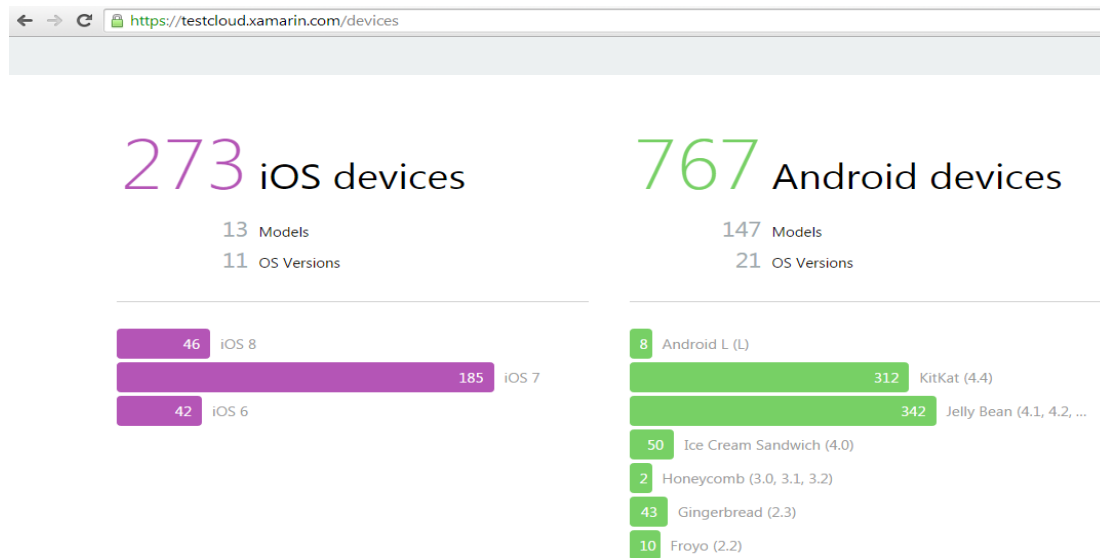


Figure 17. Xamarin cloud test devices [168]

In this way, developers may conduct fragmentation test, such as using different operating systems with different screen sizes and resolutions. Further, tests can be written with different programming languages, for instance, C# or Ruby Calabash for all applications (hybrid and native). Additionally, applications running in the test cloud can directly be run from continuous integration (CI) applications, such as Jenkins, while application performance can be monitored.

Finally, Xamarin automated UI test cases can be written either with Xamarin.UITest framework or Calabash Framework which provides developers possibilities to write test cases in Ruby using Cucumber which is well suited to Behavior Driven Development (BDD) [169].

## 6.5 Appery.io

Appery.io is a cloud based platform that mobile application developers can utilize to build mobile applications using HTML5/jQuery Mobile and can easily connect backend services and even third part APIs. In addition, Appery.io app builder uses Apache Cordova (or PhoneGap) for building mobile applications for different platforms including Android, iOS, and Windows Phone [170]. Further, since Appery.io builder is a cloud based, developers don't need to install any tool for mobile application development (Figure 18 and 19). In addition to development, builder running in the cloud easily compiles and builds the binary file which can easily be published to the app stores.

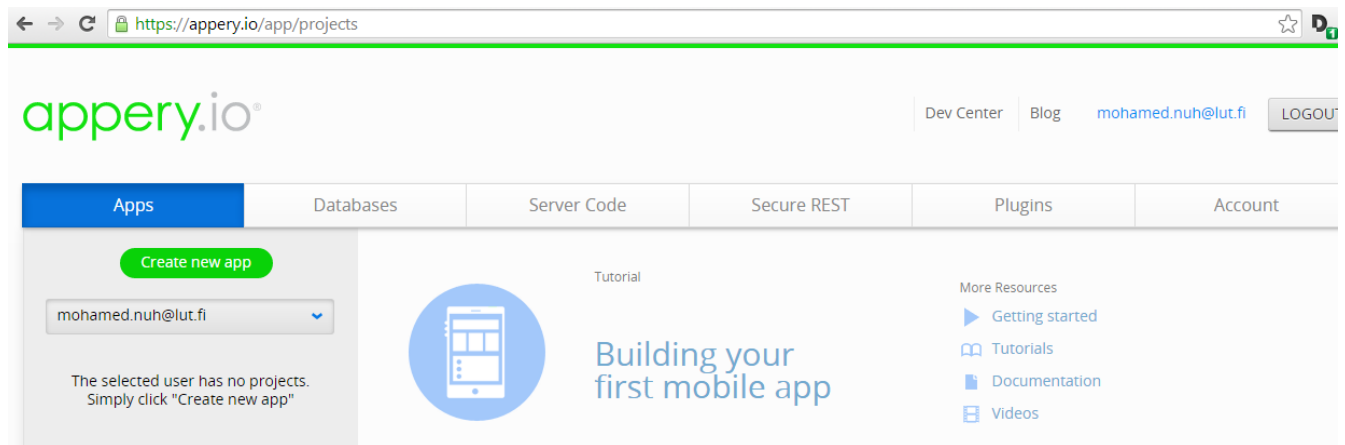


Figure 17. Appery.io application platform

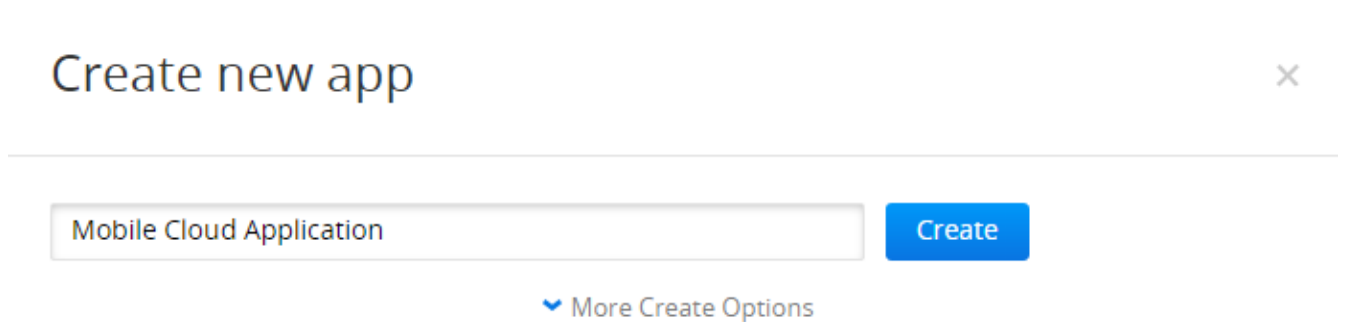


Figure 18. Creating mobile apps on the cloud



In addition, mobile application developers can easily create flexible Mobile UI using provided components, pages and UI themes [171] as depicted in Figure 19.

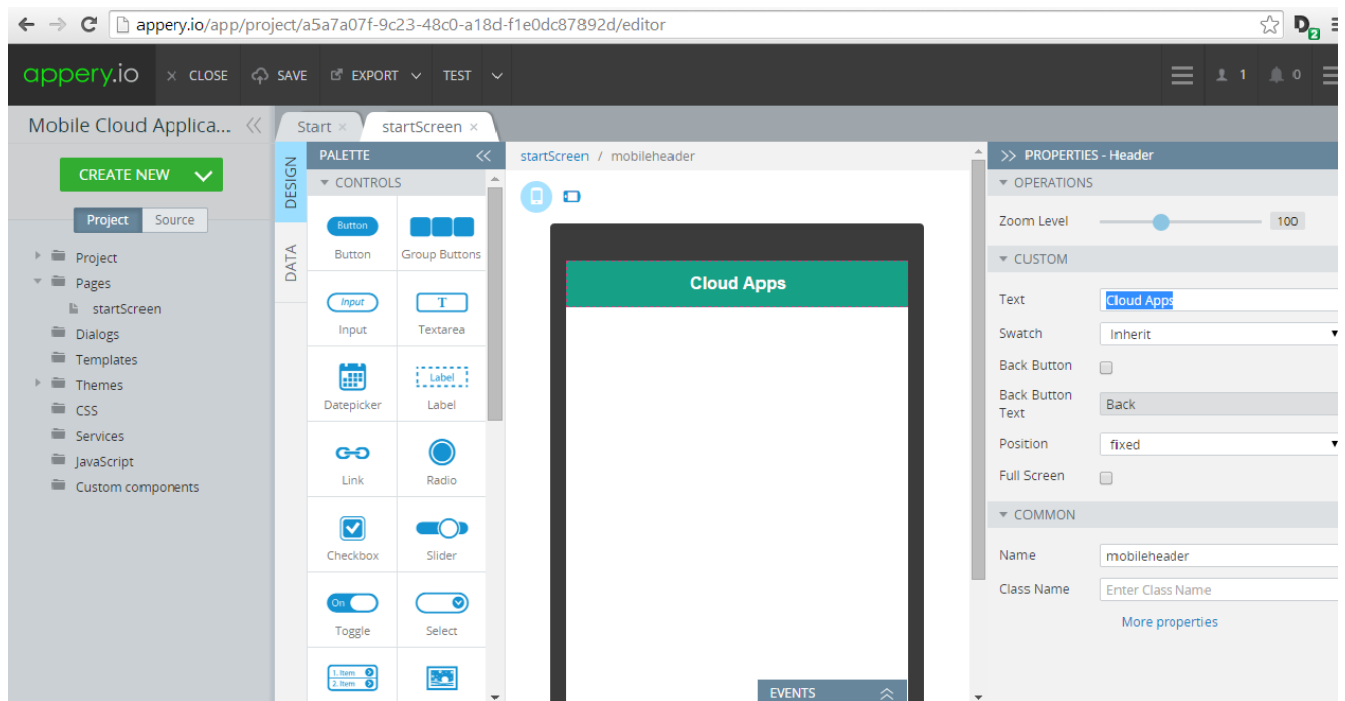


Figure 19. Appery.io Mobile UI Designer

Appery.io also provides a seamless way to bind developer's data using mapping feature which is very flexible (Figure 20) while cloud database can be used for storing application data, such as customer information, locations and so on, where mobile application communicates with this cloud database through simple REST API (Figure 21)

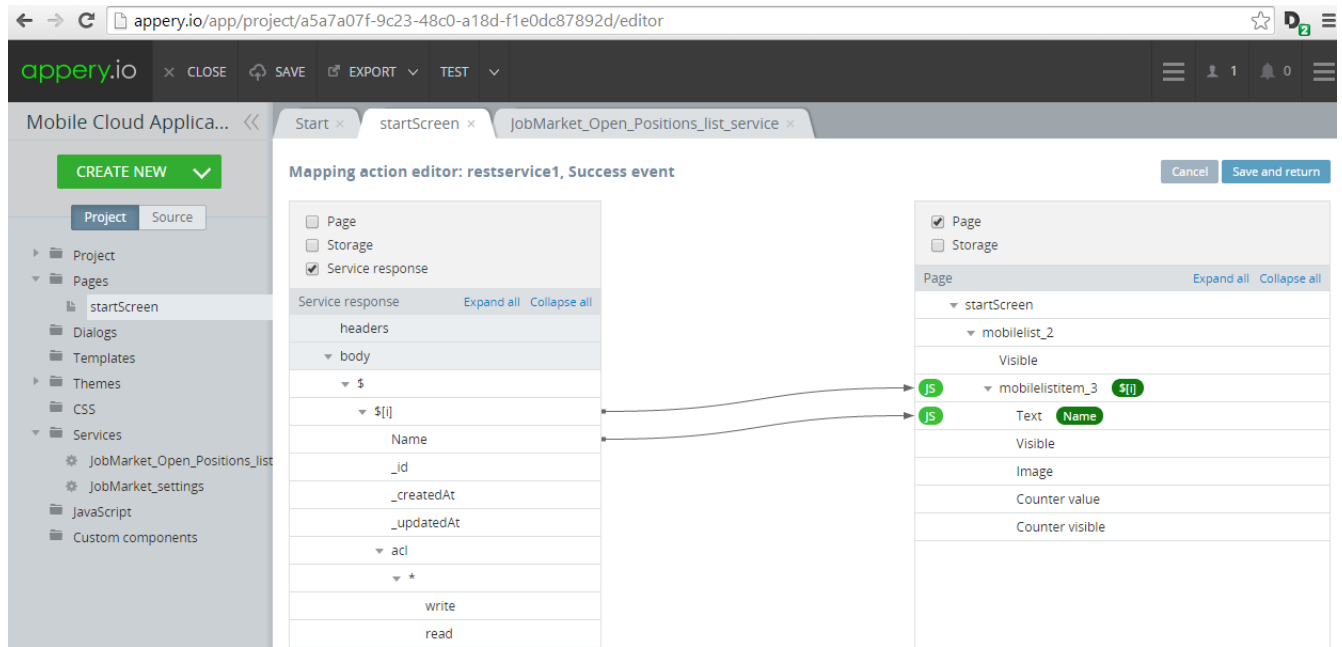


Figure 20. Appery.io cloud database

To test these features, author has created mobile application called Mobile Cloud Application using Appery.io cloud database called Job Market where collection of open positions, such as Information technology, Business Consultancy, and Financial Sector are added (Figure 21). This mobile application seamlessly communicates to cloud database using REST API. Author then built and run completed mobile application on mobile emulator (Figure 22).

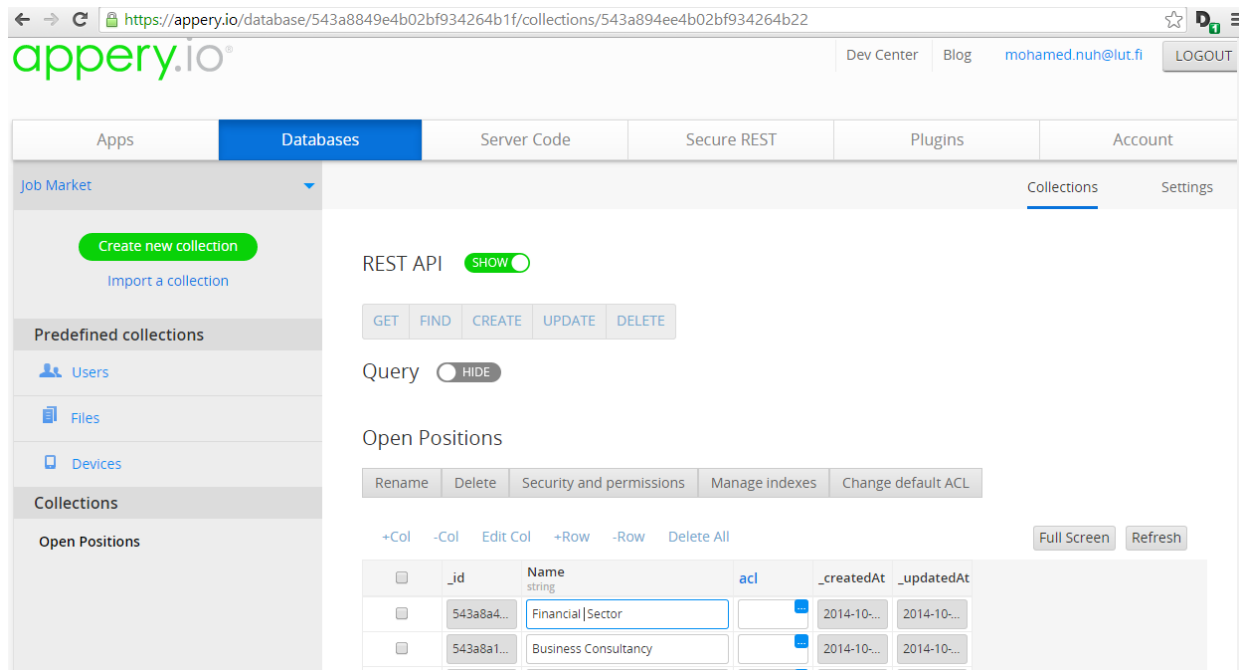


Figure 20. Appery.io Cloud Database Collections

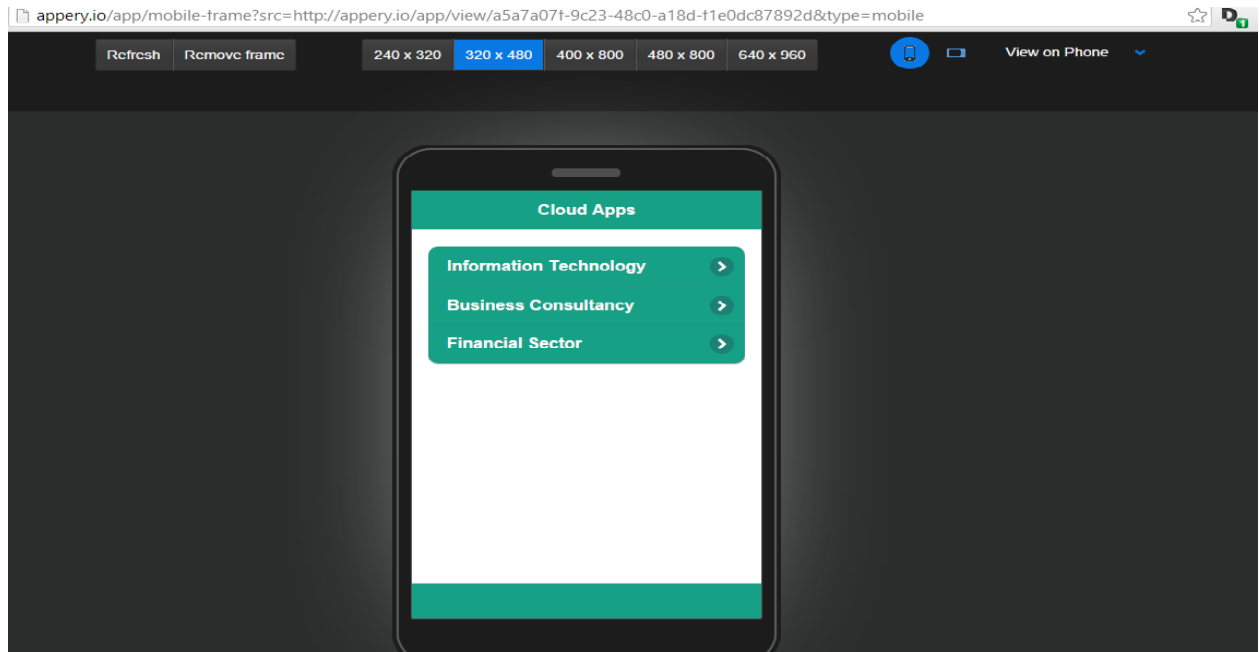


Figure 22. Mobile Cloud Application

In addition, Appery.io provides a way to easily publish completed mobile application to any App Stores, whether, Google, Apple or Windows as depicted in Figure 23.

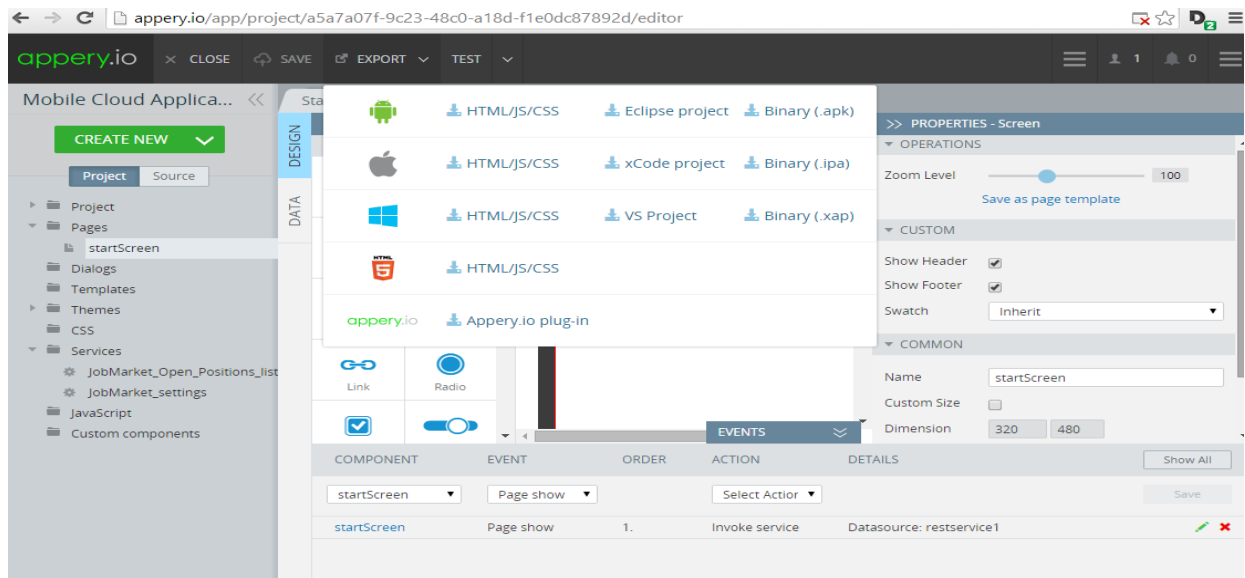


Figure 23. Mobile Application Publishing through Appery.io Platform

Another benefit for the cloud platform is that collaboration and document sharing among stakeholders is real time Figure 24. For instance, customers can quickly give instance feedback to developers. Appery.io also supports application testing either on website or phone if application is based on HTML5 mobile app while native app can be tested with Mobile App Tester.

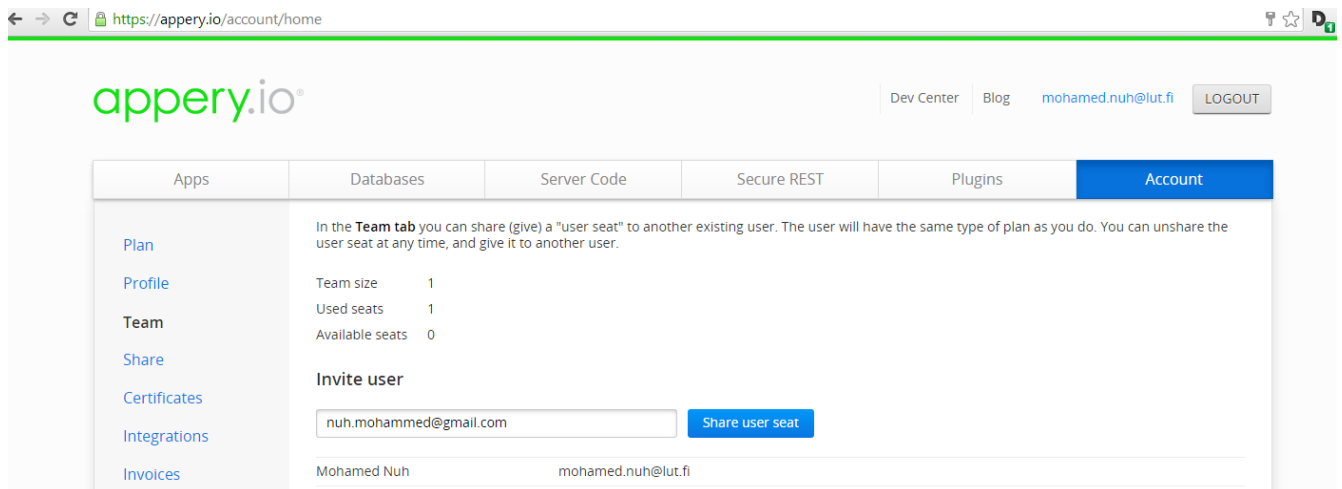


Figure 24. Appery.io Platform collaboration services

Appery.io platform gives mobile application developers possibilities to add more functionality to their mobile application using out-of-the-box plug-ins. For instance, developers may add sms functionality in mobile application using plug-in APIs, such as Twillo SMS API [172].

In summary, Appery.io is a cloud based mobile development environment integrated with backend services and third part plug-in APIs that enormously simplify development of cross-platform mobile applications, thus, dramatically alleviating fragmentation nightmare. Further, Appery.io also supports drag and drop functionality within visual editor, where non-programmers can simply drag and drop ready-made elements, while highly experience developers may use more advanced features which are also included in the editor. In this platform, one may build real mobile applications which are based on html5, jQuery Mobile for iOS, Android and Windows Phone.

## **6.6 Appcelerator's Titanium**

Mobile application developers may easily deliver cross-platform applications using Appcelerator which is an open cloud-based platform. Further, applications written in JavaScript will be able to be run as native in every major platform, such as Android, Windows Phone and iOS. In addition, this platform has three primary parts, namely, Apps [173], which provides creating, building and testing native cross-platform applications from single JavaScript codebase without manipulating HTML DOM, and the second part is Analytics part [174], which offers a real-time view of application quality, adoption, and performance and third part is Appi [175]

Further, Appcelerator's Cloud Services offers an easy integration of compelling features such as push notifications, user login, and photo uploads [176]. Appcelerator's API builder has features that enable accessing any data source while making consumable for mobile devices. For instance, data optimization feature reduces payload size for better mobile performance, while transformation of data converts from XML to JSON. API builder also offers both private and

public data sources for connecting third party applications such as Facebook [177], Salesforce [178] and SAP [179].

In addition, Appcelerator cloud platform provides layers of mobile security including data encryption, both in motion using SSL and IPsec as well at rest (server side) using AES-256, while user management, such as authentication /authorization services such as Active Directory, OAuth and also LDAP are supported in mobile security services [180].

In addition, using this cloud platform's eclipse-based IDE with mobile SDK enable developers to access for compelling features , such as code analyzer, support for variety of mobile operating systems, including Android, Tizen, and iOS, while encouraging robust software development by offering MVC architecture [181] which is combined with test automation. Furthermore, this platform offers mobile application developers a compelling feature called mobile analytics for enhancing user experience by providing immediate insights into mobile metrics, such as quality, retention and engagement.

To use Appcelerator cloud platform, one has first to register as depicted in Figure

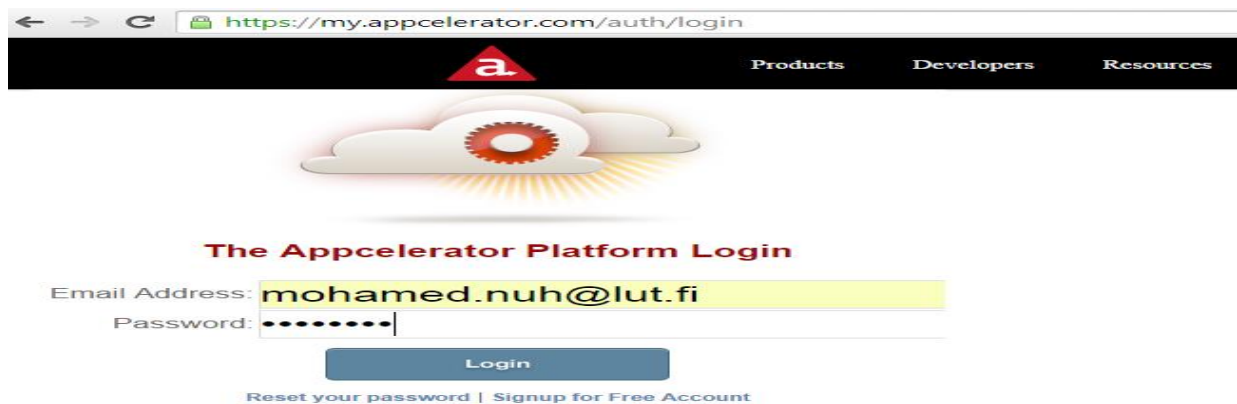
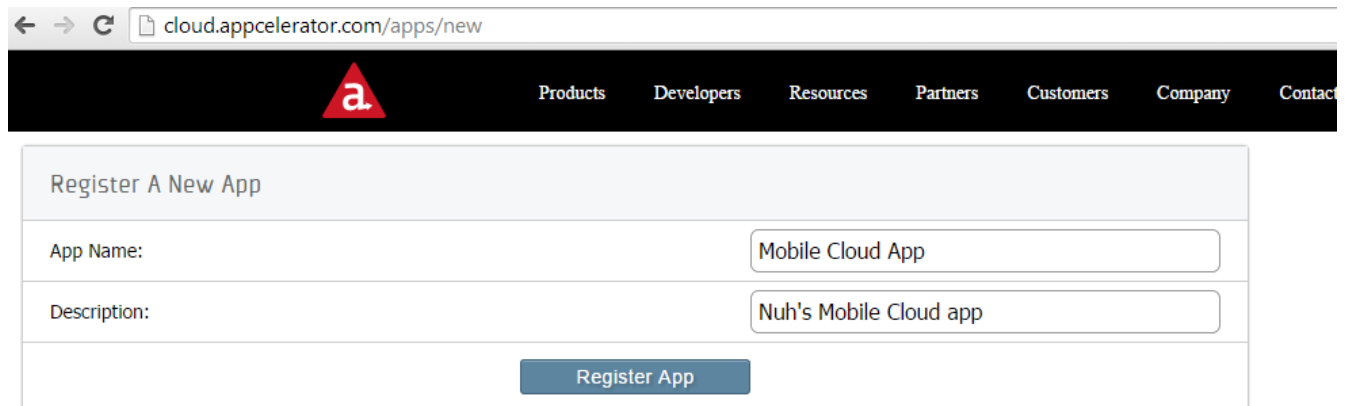


Figure 25. Appcelerator Platform login page

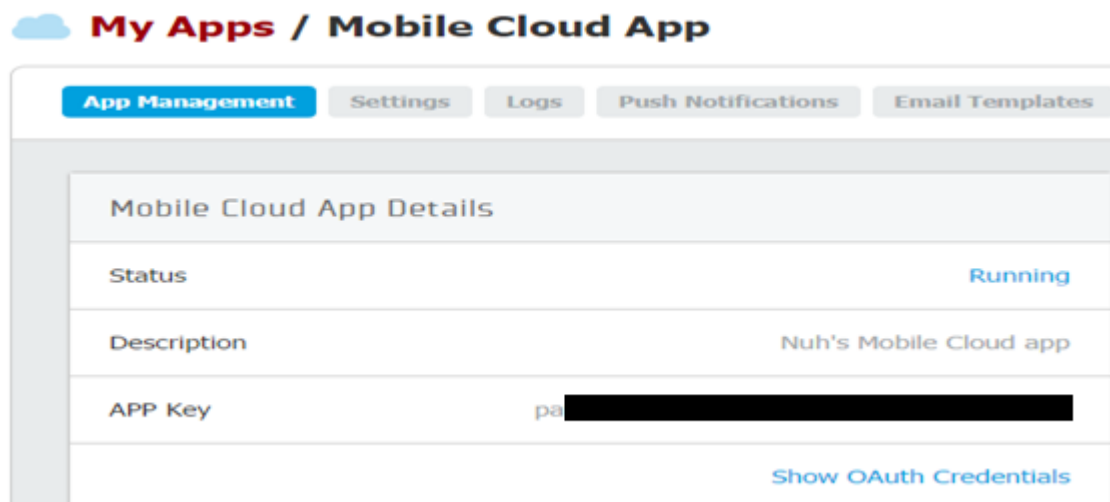
And start registering cloud application by giving application name and description Figure 26.



The screenshot shows the Appcelerator web interface for registering a new app. The browser address bar displays 'cloud.appcelerator.com/apps/new'. The page has a dark navigation bar with the Appcelerator logo and links for Products, Developers, Resources, Partners, Customers, Company, and Contact. Below the navigation bar is a form titled 'Register A New App'. The form contains two input fields: 'App Name' with the value 'Mobile Cloud App' and 'Description' with the value 'Nuh's Mobile Cloud app'. A 'Register App' button is located at the bottom of the form.

Figure 26. Appcelerator Platform login page

After mobile application has been created, one may able to call cloud methods through provided SDKs, such as Android SDK, Titanium SDK and iOS SDK. Further, using SDKs, one may able to enable cloud services in Titanium project. For instance, enabling cloud service using Android SDK, one may download appcelerator-sdk-android-1.0.0.jar and copy into the Titanium Studio project's libs folder while adding internet permission into the AndroidManifest.xml, such as `<uses-permission android:name="android.permission.INTERNET"/>` and calling `APSServiceManager.getInstance()`, while passing application context and unique key application. For instance, following is an example of passing application context and unique application key of the created mobile cloud apps as depicted in Figure 27 and 28.



The screenshot shows the 'My Apps / Mobile Cloud App' page in the Titanium Cloud interface. The page has a header with a cloud icon and the title 'My Apps / Mobile Cloud App'. Below the header is a navigation bar with tabs: 'App Management' (active), 'Settings', 'Logs', 'Push Notifications', and 'Email Templates'. The main content area is titled 'Mobile Cloud App Details' and contains a table with the following information:

Status	Running
Description	Nuh's Mobile Cloud app
APP Key	pa [REDACTED]

At the bottom right of the table, there is a link that says 'Show OAuth Credentials'.

Figure 27. Mobile application created in Titanium Cloud

```

import com.appcelerator.aps.APSServiceManager;

public class MainActivity extends Activity {
    try {
        APSServiceManager.getInstance().enable(getApplicationContext(), "ps[REDACTED]");
    } catch (RuntimeException e) {
        e.printStackTrace();
    }
}

```

Figure 28. Enabling cloud services via Android SDK

After cloud service is enabled in Titanium Studio using any provided SDK [182], mobile application may able to call Titanium Cloud Services using Appcelerator Platform Services API, such as APSAnalytics, APSPhotos APSPlaces, APSChats [183] and so on. In addition, mobile applications can be uploaded, test and deploy to specific mobile devices.

## 6.7 Telerik Platform

Telerik is a cross platform mobile development environment which combines a rich set of UI tools with powerful cloud services for developing not only web, hybrid but also native apps [184]. In this platform, developers may able to seamlessly design mobile application with drag and drop UI interface Figure 29.



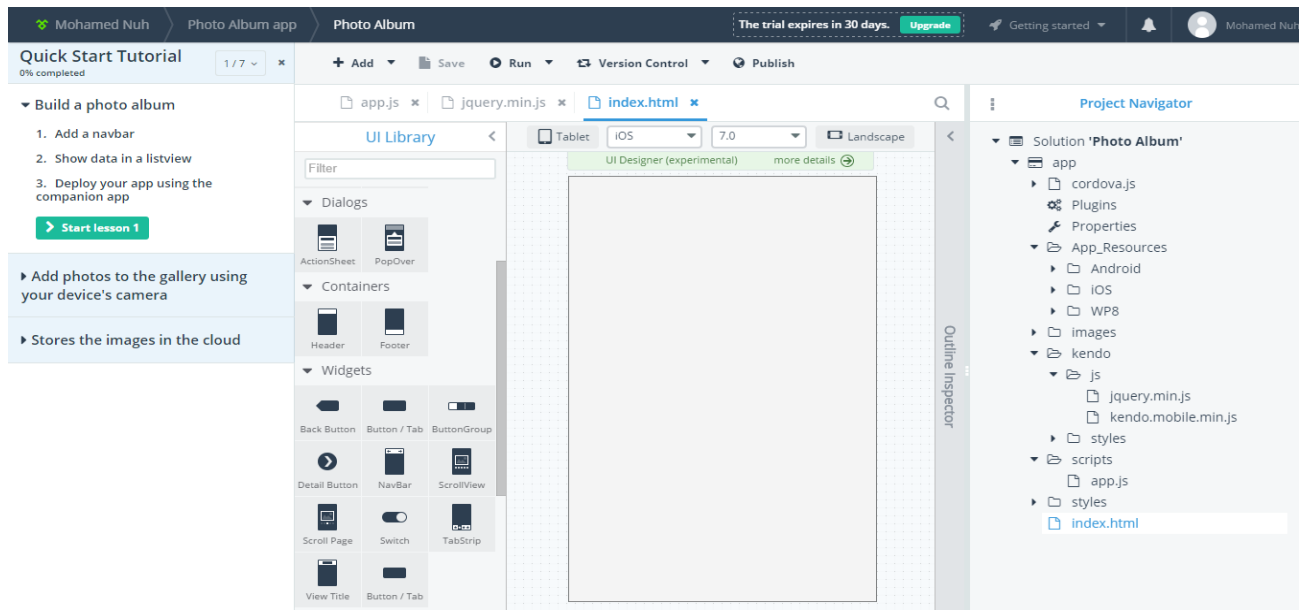


Figure 29. Telerik Platform Designer

Further, stakeholders can easily collaborate in real-time through cloud service by inviting other stockholders as depicted in Figure 30.

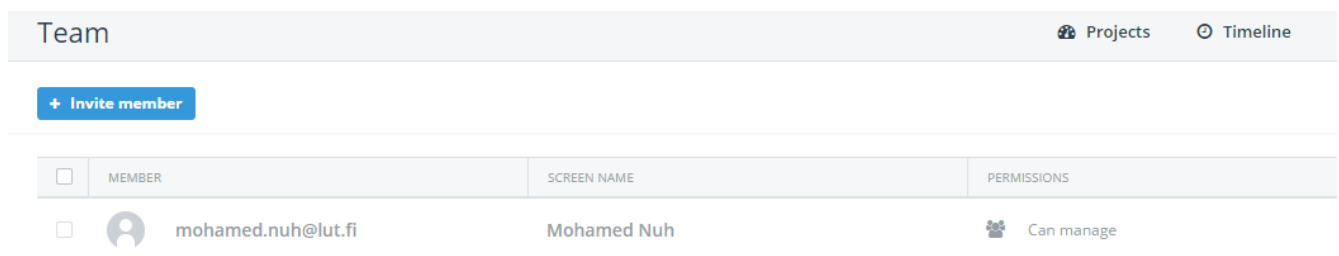


Figure 30. Telerik Platform Team Collaboration

In addition, Telerik Platform provides application security services, such as authentication, data services and offers seamless integration to backend data.

Automated mobile application testing can be optimized with cross-platform automated testing, while application packaging and deployment can easily be done. Moreover, Telerik enables application developers to monitor and measure consumer base by gathering demographic data and application usage [185].

In addition, Telerik platform provides cloud services for management, delivery optimization, and analysis of applications across all screens with a single, unified platform [186].

Telerik is modularity based platform, where cloud services and UI tools are rigorously combined. For instance, mobile developers who intend to design and create prototypes may easily utilize AppPrototyper module. Further, AppBuilder module (previously known as Icenium) which is an Integrated Cloud Environment (ICE) provides mobile developers an opportunity to seamlessly build hybrid mobile applications for various platforms Figure 31,

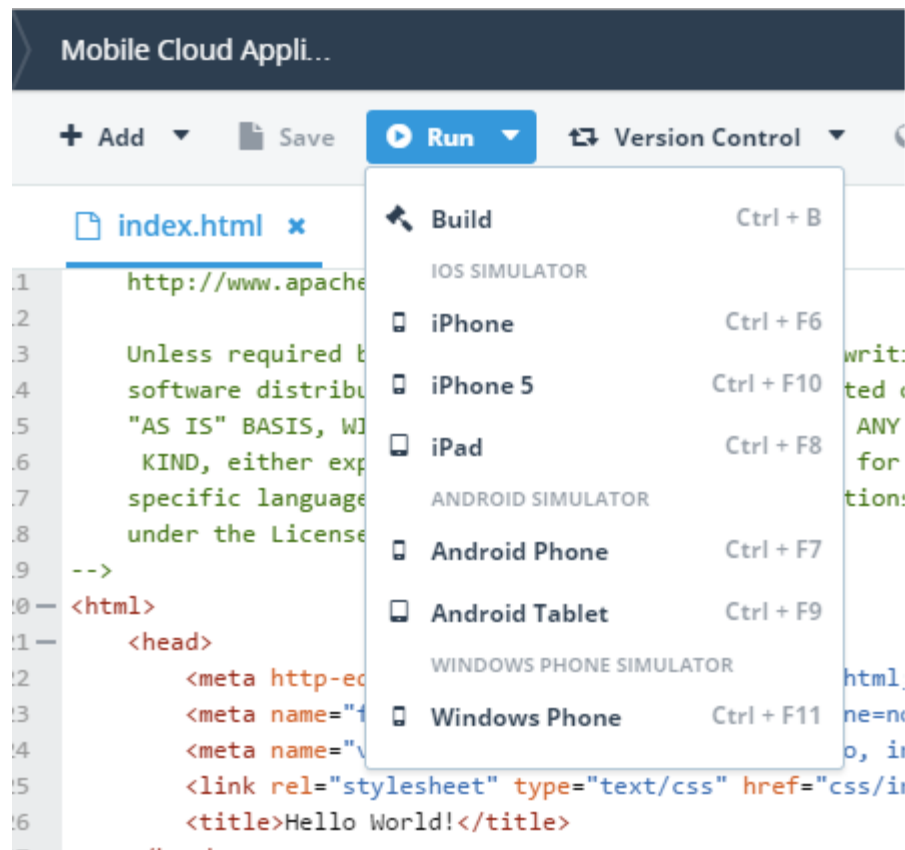


Figure 31. Telerik Building mobile applications for various platforms

Where complete mobile application can seamlessly be installed onto the target device by simply scanning provided QR code as shown in Figure 32.

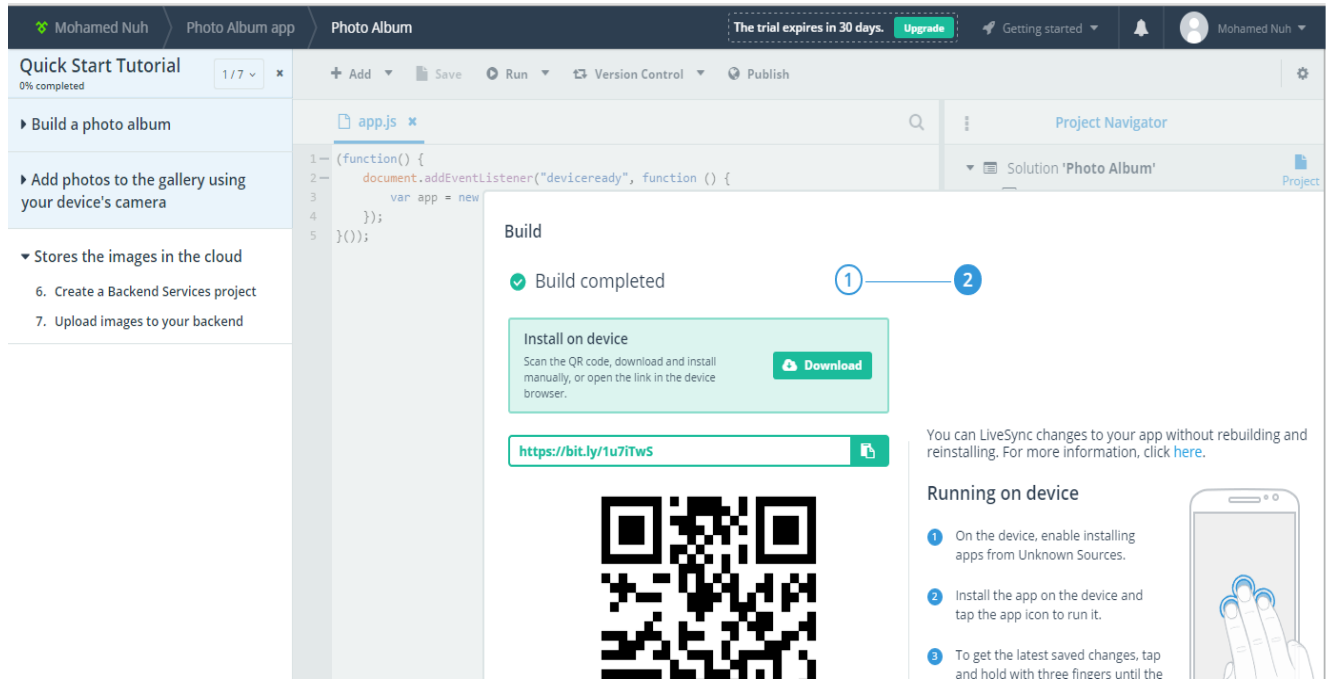


Figure 31. Telerik Platform Cloud Building Service

Further, Telerik also facilitates deployment and publishing complete application to the App Store or Google play as depicted in Figure 32.

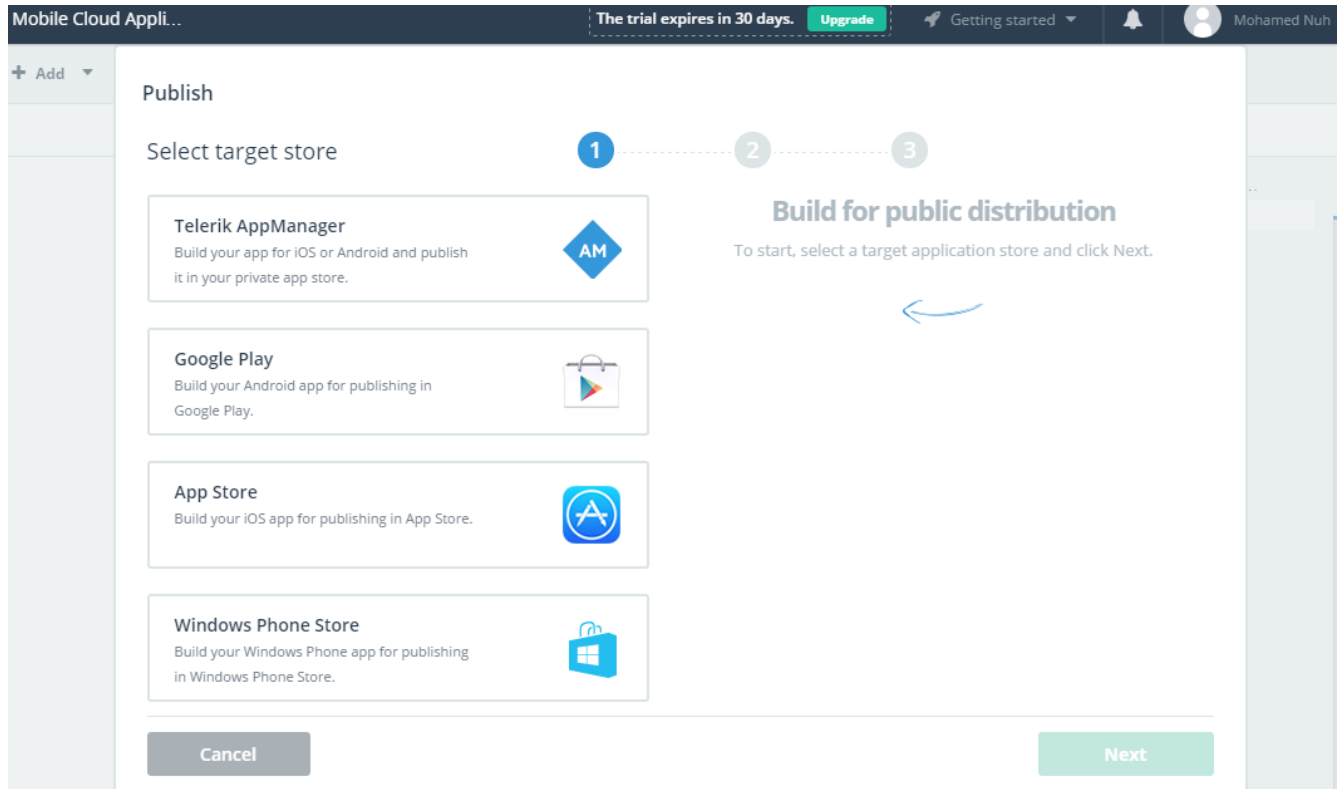


Figure 32. Telerik Platform can publish app to different App Stores

In addition, since ICE is a cloud based platform, it has capabilities to abstract all platform related dependencies, thus enabling app developers to concentrate on application developments. ICE internally uses Apache Cordova that creates an opportunity for web developers to use their existing skills such as HTML, JavaScript, and CSS to develop compelling mobile applications [187].

In addition, Telerik cloud-based backed may provide a compelling user-management feature that can seamlessly be utilized by application developers, while back-end portal offers an interface for interacting data, such as permission setting authentication management and even code debugging. Furthermore, Mobile application code can be executed using Telerik's cloud code functionality and can be used from all mobile platforms including Android, Windows phone and iOS. Telerik user may able to use automated testing for native, web apps and even hybrid applications. Including Android and iOS. What is more, Telerik also provides mobile application management for securely distributing mobile application to a private store. For

instance, developers can seamlessly not only deploy apps, but also manage and update if needed. In addition, Telerik platform provides developers another service called AppFeedback which enables beta users to send a feedback directly from mobile applications [188].

## 6.8 Kony

Kony provides a very sophisticated Studio for mobile application developers to create not only rich, multi-edge but also connected apps using visual app designer where designing for native and web can be done simultaneously [189].

Further, Kony also provides an interpose service integration designer through Kony MobileFabric where developers are enabled to discover all available services and business objects within enterprise systems such as SAP, Oracle, IBM and possibly map all backend field directly to mobile app [190].

In addition, Kony MobileFabric provides developers a possibility to select any development framework, such as open development platforms including, Sencha, JQuery, PhoneGap and even native tools such as Android and iOS for mobile development and then seamlessly and securely connect to enterprise system and access back-end services through REST, SOAP, XML, and JSON [191]. Further, using Kony cloud Platform is extremely simple, suffice to register into the Cloud and add a new Kony Cloud as depicted in Figure 33.

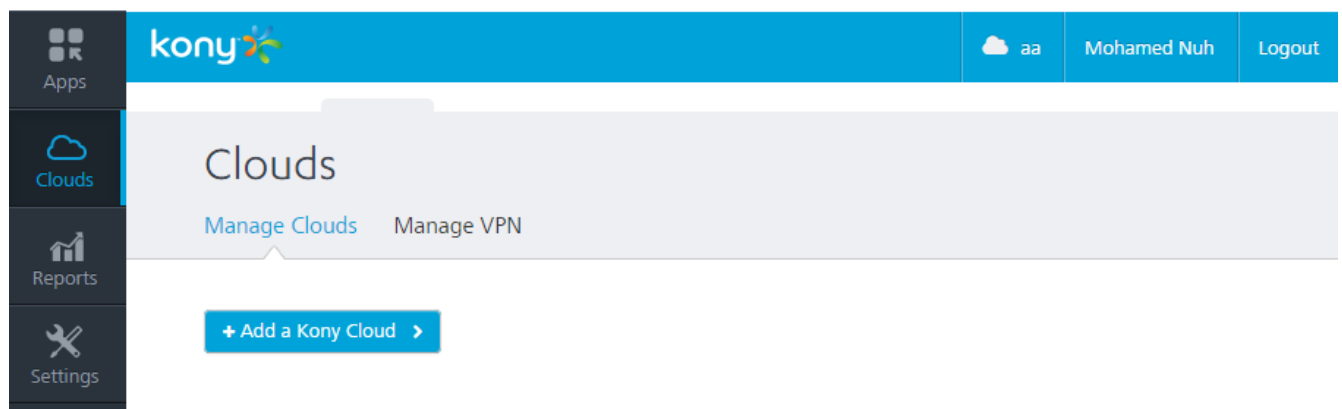


Figure 33. Creating Kony Cloud

Further, Kony also enables not only publishing but also sharing prototypes and application designs between designers through the cloud to instantly preview on iOS, Android and Windows while changes are shared in real time [192]. In addition, developers can easily publish mobile apps to the cloud for download and review. Team collaboration on the cloud has been depicted in Figure 34.

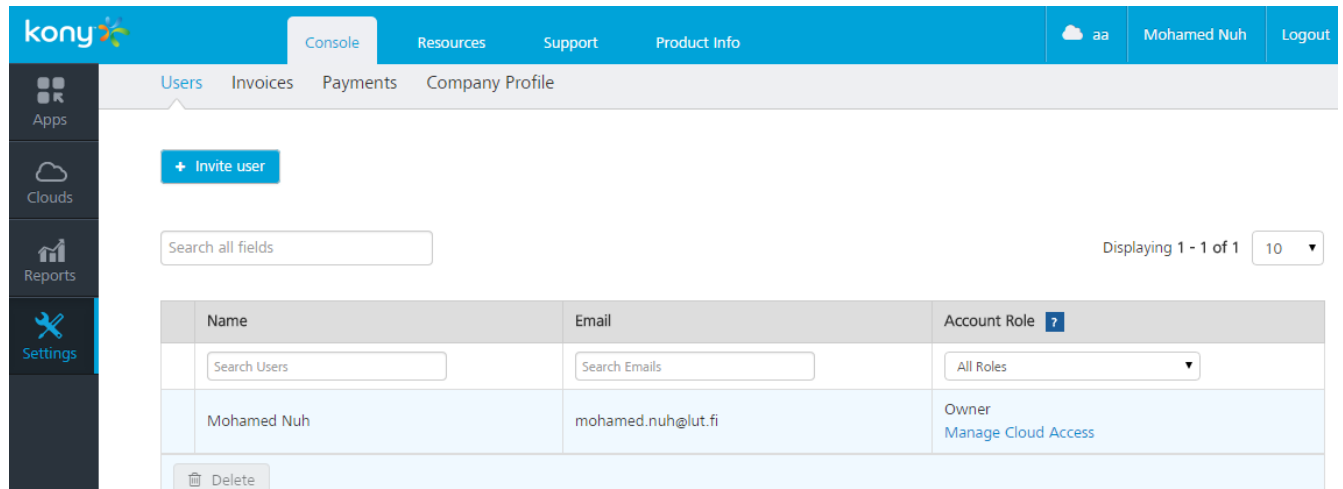


Figure 34. Team Collaboration in Kony Cloud

Further, Kony Experience Cloud is composed of three different solutions, namely, Kony Visualization cloud where users are enabled to define and design prototypes and seamlessly collaborate with other stakeholders and fully integrated with Kony Development Cloud for not only developing, but also testing and easy deployment and lastly Kony Management Cloud for managing mobile, devices and data [193].

Kony only provides an integrated platform that supports the entire application software development phases and empowers enterprise to quickly design, build, deploy and manage multi-channel app experiences [194]

What is more, Kony user may leverage their code by using drag and drop tools and widgets while built apps can seamlessly be deployed and published to public app stores. In general, Kony Cloud is comprised of various cloud types with different services and functionalities, namely Visualization cloud for designing and prototyping, Development cloud for developing apps from single code base and Management cloud for managing apps, devices [195].

Further, Kony platform provides messaging and notification platform which covers Android, iOS, Windows and Blackberry with single RESTful API as shown in Figure 35, where messages are delivered by context and even create and manage multi-channel marketing campaigns [196].

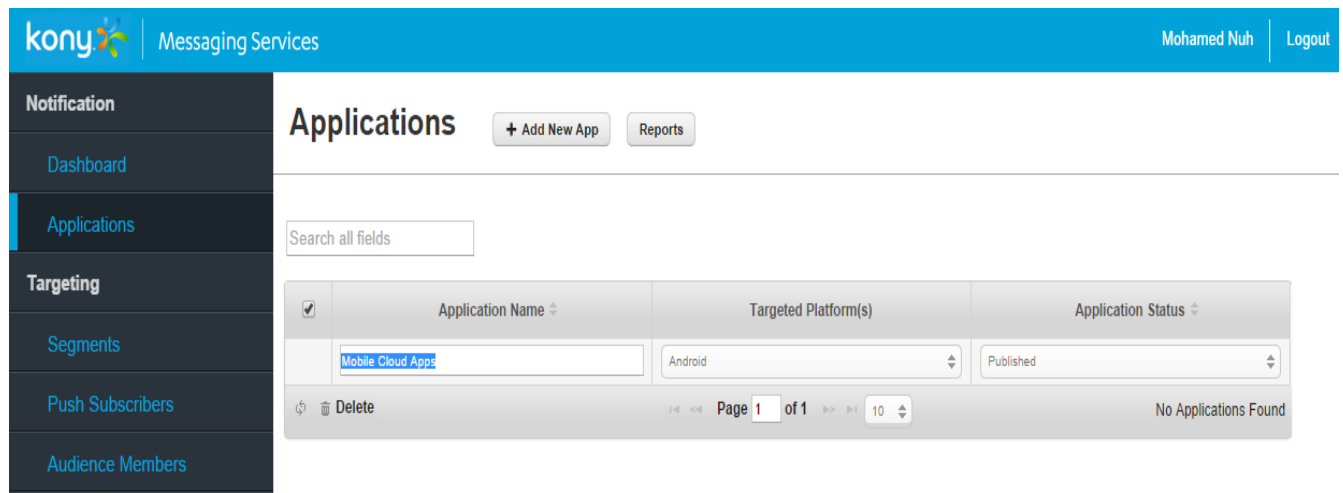


Figure 35. Kony Platform Messaging and Notification service

Additionally, Kony also provides Reporting and Analytics depicted in Figure 36, to enable developers to analyze notification, schedule reports and even push analytics data to Adobe Omniture, IBM Coremetrics, Google Analytics, Webtrends and TeaLeaf [197].

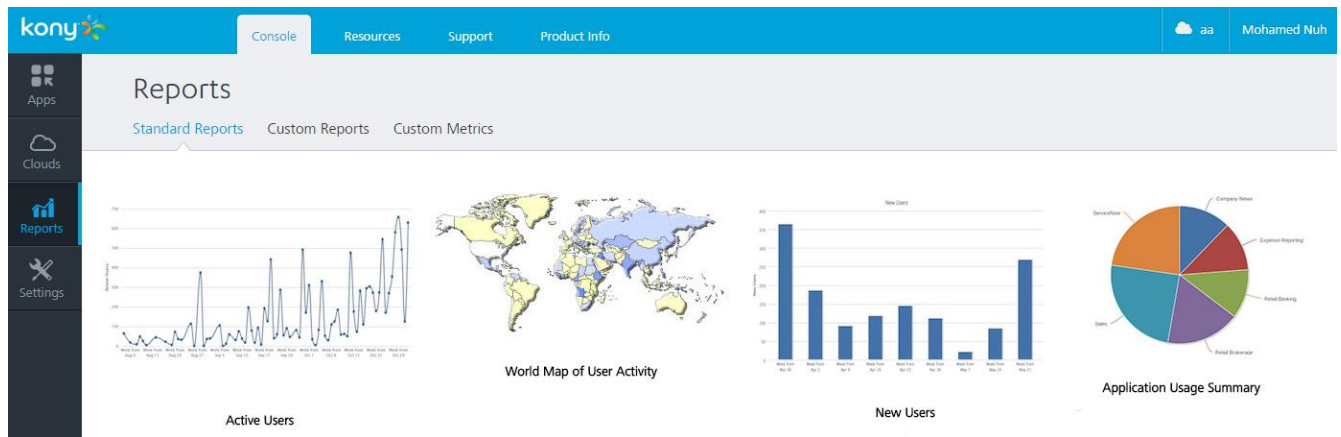


Figure 36. Kony Platform's Reporting and Analytics

Further, developers can utilize an easy to use portal for managing and controlling apps while getting information about application behavior through analytics services. Kony's architecture is comprised of three layers as depicted in Figure 37.

The first layer is the development platform where developer can choose to create enterprise mobile application where leveraging MBaaS services, like authentications and notifications through simple APIs like RESTful and even connect to third party web services through JSON and so on.



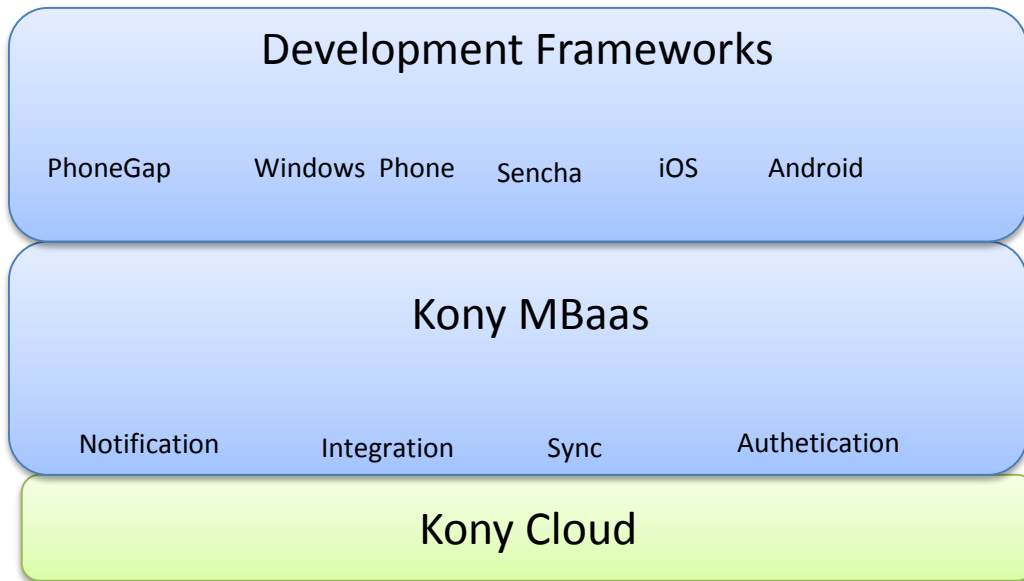


Figure 37. Kony platform [198]

In additions, Kony also supports various types of deployment models, such as deployment to the public Cloud where one may securely integrate to Kony backend services enterprise data center and utilize business intelligence services to view reports and analytic, while deployment to private cloud one may able to choose from virtual private cloud or dedicated physical hardware options and even create a secure VPN connection to your enterprise data center [199].

Further, Applications can also be deployed onto the private Cloud where developers can use virtual cloud and connect securely to the enterprise data. Platform users may choice to deploy apps on local server rather than Kony Cloud.

What's more, Kony Cloud uses Amazon's AWS data center infrastructure and also supports High Availability (HA) by running two instances of every component than runs customer cloud application [200].

## 6.9 Parse

Parse is a cloud application development platform for creating device-agnostics applications which can run on multiple operating systems, including Android, Windows Phone and iOS [201]. Further, mobile developers can easily access scalable backend services, such as data storage, notification, and third party integration through Cloud Code. Parse's platform consists of three separate solutions, namely, Parse Core, Parse Push and Parse Analytic [202].

Parse Core provides an easy way to securely save and query data in the cloud and enables Parse Core enables apps code to be executed in Parse Cloud using JavaScript SDK where Parse Dashboard is used for data manipulation and even sending push notification [frame-39].

Furthermore, with Parse Push, apps developers can easily send notifications to the mobile applications through various channels such as REST API or SDK [203].

What is more, Parse offers a way to measure a real-time application usage using API request which is based on REST verbs. Parse Analytics is very easy to use in terms of code complexity, For instance, using Android device, developer can easily create any important metrics to be monitored with few lines of code.

```
Map<String, String> GameMetrics = new HashMap<String, String>()
GameMetrics.put("HighScoreRange", "1500-3000"); GameMetrics.put("customerSegment",
"youth");                                     ParseAnalytics.trackEvent("search", GameMetrics);
```

All analytics data can be viewed on Parse Dashboard and displayed with multi-graphs. REST API is used to seamlessly interact with Parse platform, for instance, JavaScript code can access Parse data to perform metric analytics [204]

Further, Parse also provides Cloud Code solution where developers are enabled to abandon using physical servers for building any mobile application.

In order to use parse cloud, one has to initially register and then create a mobile application as depicted in Figures Figure 38 and Figure 39.

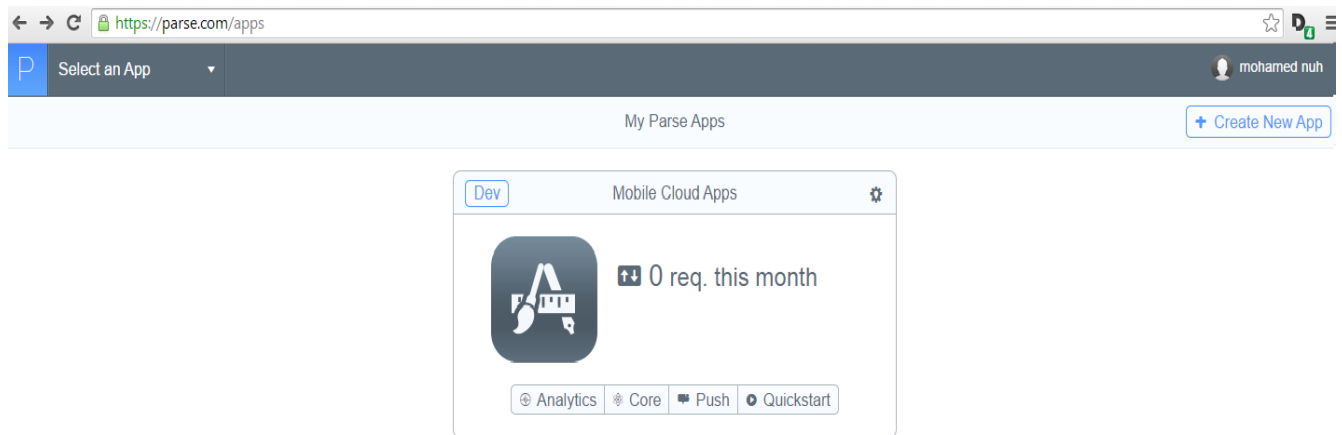


Figure 38. Created Parse application

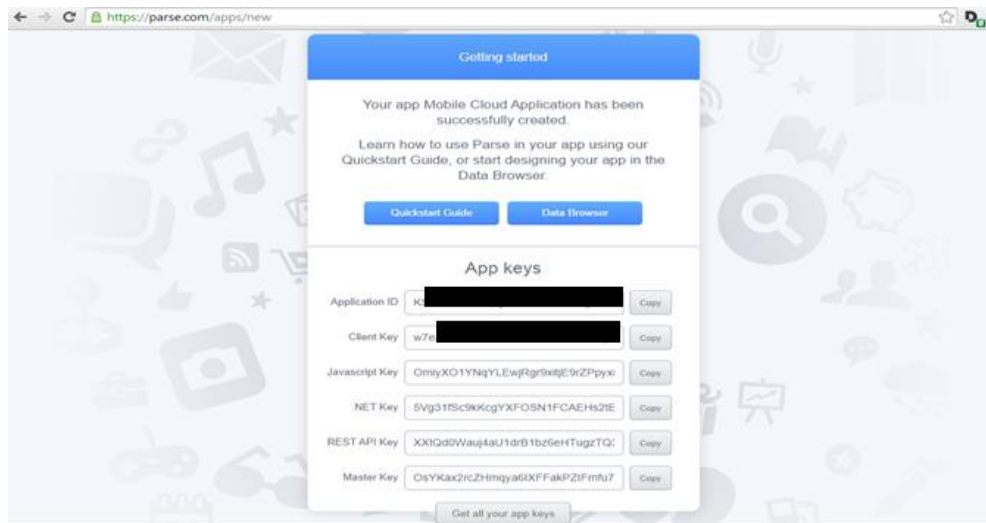


Figure 39. Application ID and Client Key Generation

In addition, Parse offers mobile application developers a smart way to develop mobile applications without dealing with any physical servers and any code that may not be suitable for running in mobile device, will be deployed and run in the Cloud Code.

Using Parse command tool, mobile developers may able to write and deploy and run on the cloud. For instance, after downloaded Parse Command tool, code directory that will run on the cloud is created by this command *parse new*.

JavaScript code can be locally written as shown in Figure 40 and easily deployed to the cloud code by *parse deploy*.

```
// Use Parse.Cloud.define to define as many cloud functions as you want.
// For example:
Parse.Cloud.define("MobileApplication", function(request, response) {
  response.success("Nuh Mobile Cloud Application!");
});
```

Figure 40. JavaScript written in cloud code

Deployed code can be seen and hosted on the cloud as shown in Figure 41.

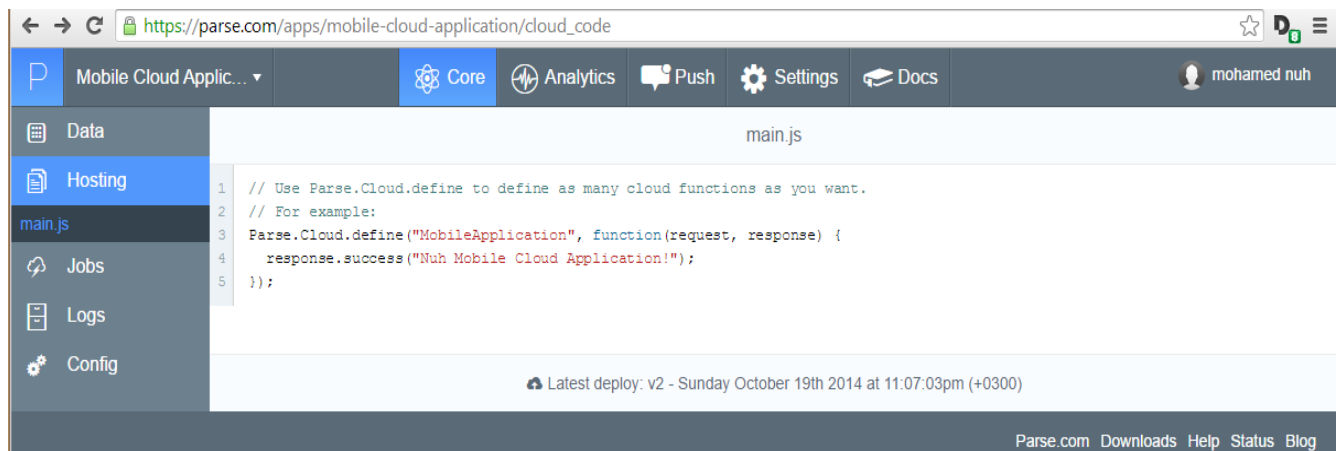


Figure 41. Deployed code is hosted in the Parse cloud

In addition, Parse Cloud Platform enables developers to seamlessly share mobile apps with someone else who may get a permission to get an access by simply inviting as shown in Figure 42. Further, shared apps may also easily unshared.

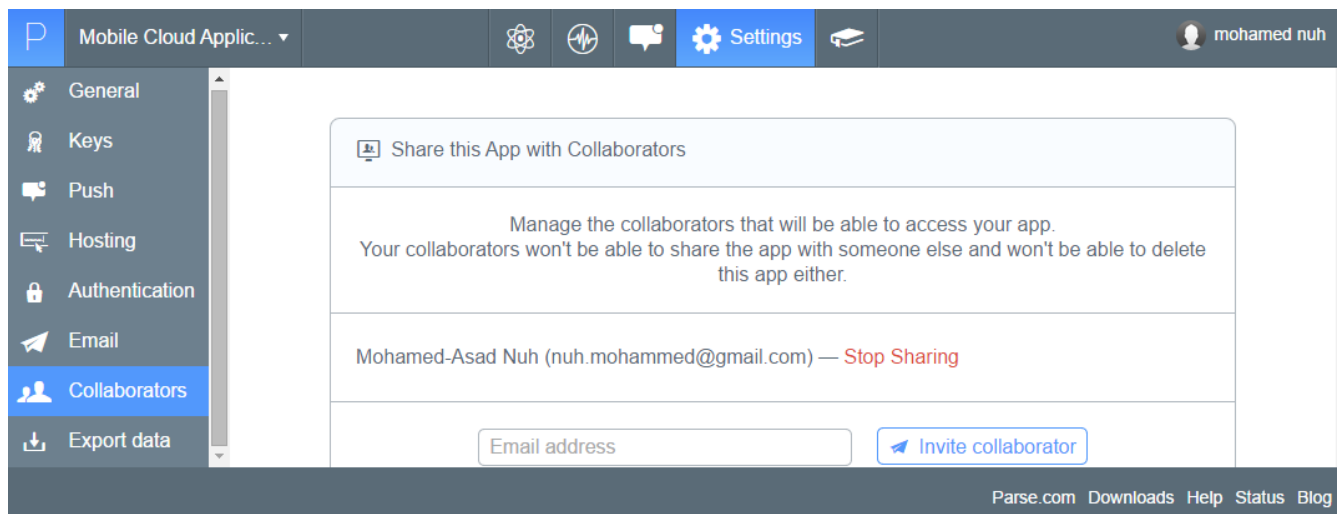


Figure 42. Collaborators may easily share or un-share mobile application

## 7. EVALUATION

In this chapter, cloud based cross platform development framework will be evaluated according to the evaluation criteria mentioned above. In this evaluation, six different criteria have been selected to measure how platform is integrated to the cloud. Further, results of these evaluation are based on testing and using cloud-based features in the target platform in order to understand and compare with other tools, while developer's concern were attempted to be addressed. For instance, can the cloud service be accessible in the development phase? How about the collaboration among stakeholders through cloud? Can mobile application testable on the cloud? Is it possible to build and deploy onto cloud? Can distribution to App Store be done via cloud?

Even though, there are dozens of other tools that may provide cross-platform solutions, most popular tools with possible cloud integration have been selected to be evaluated. In general, cross platform mobile application development tools are different in terms of provided features, capabilities, flexibilities and also integration to cloud services, which may provide essential back-end services such as authentication, collaboration, notification and messaging, not only for public mobile applications, but also enterprise applications.

Further, mobile application developers may have faced burning but relevant questions, such as:

- What is the best way to minimize code duplication?
- What is the best way to develop one application on multiple devices with different operating systems and architecture?

Those questions have exacerbated the demand for cross platform application development tools in the market.

While concept of cross platform mobile development may refer to a solution where a single codebase is written for apps that may be deployed on different devices with different operating systems, this may at least in theory solve fragmentation problems. In addition, developers constantly demand flexibility, collaboration among stakeholders and stable single code base for different mobile applications. For instance, while PhoneGap, Kony and Rhomobile cross

platforms provide cloud access in development/testing phases and also support cloud building, but developers using these tools may not be able to distribute mobile application via them.

Consequently, developers must find another way to distribute their mobile applications outside of these tools. Therefore, these above mentioned cloud-based tools may not cover the complete mobile application development life cycle, namely, design/development/testing/building and distributing.

In this evaluation, selected platform tools will be categorized in terms of cloud connectivity to full mobile application development life cycle phases. For instance, PhoneGap, Kony and Rhomobile platforms support all mobile application life cycle phases except distribution phase defined in criteria list. Since, they support fifth out of six criteria, these tools will be categorized as type-5.

Type-5 tools provide mobile application developers code reusability, where single code base can be utilized for different devices and operating systems. Therefore suitable for solving fragmentation challenges.

We have also evaluated Type-4 tools that support four out of six criteria list. For instance, we have observed that Titanium mobile application development platform provides mobile application developers a possibility to develop mobile application for different devices and platforms while development can be done in the cloud. Further, developers who may use Type-4 platforms are enabled to test mobile application in the cloud, while developed mobile applications can seamlessly be not only built but also deployed to the cloud. Further, Type-4 platforms may also alleviate the fragmentation pain. However, type-4 users may not be able to collaborate with other stakeholders, which is obviously a big drawback for this type. Further, Type-4 platforms don't allow deployed and built applications to be distributed to Apps Stores via them.

Since Xamarin supports code reusability where single code base (C# based) can be used to develop for different mobile platforms and operating systems, it can be defined as cross-platform. However, the cloud connectivity for this tool is limited only to mobile application

testing, where more than one thousand mobile devices are provided to be tested against mobile application, thus, may also solve the fragmentation challenges. Therefore, Xamarin can be categorized as Type-1 platforms, since it fulfills only one out of six criteria list.

Other similar platforms include MoSync, which is classified as a cross platform, provides application developers to access cloud features in development phase, but only through third party called Cloudbase.io.

Further, full cloud based mobile cross platforms in terms of cloud connectivity, and cloud back-end services, like notifications, authentication, database, collaboration and also distribution to app stores through cloud include platforms such as Appery.io, Telerik and Parse. These platforms enable cloud connectivity to each mobile application development life cycle, whether in designing, development, testing, building and also deployment phase, while completed mobile applications can seamlessly be distributed to Apps Stores. These platforms are categorized as Type-6 platforms. These platforms enable developers a unique way to collaborate early in the development phase, where code/design can be rigorously reviewed to ensure application quality.

In addition, Type-6 platforms also enable developers to access back-end servers, such as security features, messaging and notifications to application users.

In this paper, we have acknowledged that mobile application developers are faced with fragmentation challenges, but we have also observed that cloud based mobile application platforms can seamlessly solve fragmentation challenges, while some of them may still provide back-end services that may shorten time to market and enhance application quality and security.

Therefore, author recommends Type-6 cloud based mobile application platforms to be used for mobile application development not only to mitigate fragmentation risks, but also for security, user engagement via back-end services, application quality via cloud collaboration and short time to market.

Further, evaluations of all platform types are depicted in the following table 1.



**Table 1.** Evaluation results for cloud connectivity of cross platform mobile development tools

Cross platform Criteria	Cross-platform	Cloud in development Phase	Collaboration between developers in the Cloud	Testing application in the Cloud	Building and deployment in the Cloud	Distribution to Apps Store via Cloud
PhoneGap	yes	yes	yes	yes	yes	no
Rhomobile	yes	yes	yes	yes	yes	no
Xamarin	yes	no	no	yes	no	no
Appery.io	yes	yes	yes	yes	yes	yes
Titanium	yes	yes	no	yes	yes	no
Telerik	yes	yes	yes	yes	yes	yes
MoSync	yes	yes	no	no	no	no
Kony	yes	yes	yes	yes	yes	no
Parse	yes	yes	yes	yes	yes	yes

## 8. CONCLUSIONS

Since smartphone market is rapidly changing where usage of mobile applications have dramatically increased, the characteristics of mobile phones have also changed.

Mobile users have been demanding more advanced services and features, such as Context-aware-location based application, better UX and advanced multimedia features.

In addition, mobile OS providers' attempt to deliver developer-centric OS that may invigorate rapid development. Competitions among device manufactures on one hand and among OS providers on the other hand created discrepancies among devices and OSes, which in turn made an exceeding hard for developers to deliver hundreds of similar featured applications with various versions for the market consumption.

Therefore, one of the biggest barriers in mobile application development is mobile fragmentation, which causes mobile application developers to face unbearable challenges from rapidly changing OSes, diverging screen resolutions and various keypad types. Further, developers in fragmented ecosystem are undoubtedly facing huge challenges, not to mention the administrative challenges, due to rapid changes in device features and non-standardized architectures, but also fragmentation in skills in programming languages.

Therefore, alternative solution may be needed, such as cloud computing, especially mobile cloud computing (MCC). In addition, one of the drawbacks of MCC may be related to latency problem which can easily be solved by minimizing distances between application and users, since is the distance that greatly affects latency through cloudlet, which may contain similar cloud services but has differences in terms of ownership. Further, latency problems can also be solved through cloning device's system image known as clonecloud which is based on augmented execution solution.

Attempts to alleviate fragmentation pain in mobile application development have been tried through Cross Platform Frameworks (CPF) where mobile application developers would only concentrate business logic of the application rather than portability issues. Some platforms such as hybrid frameworks may even be able to combine web with natives and support state-of-art web technologies, such as CSS3, HTML5 and JavaScript.

CPF may enable developers to design and develop resident mobile application (RMAs) that can be seamlessly deployed to different mobile platforms and also due to expansive APIs, CPF also provide deep integration with device's resources, such as camera, GPS, and accelerometer. In this way, developers may seamlessly interact with provided APIs.

CPFs may have alleviated many fragmentation challenges and drastically alleviated developer's burden. However, cross platform frameworks still depend on native tools for building and compiling, while emulators and/or simulators may also be needed for testing.

As a result, some framework vendors attempted to mitigate this dependencies and created remarkable solution by offering cloud for building and compiling RMAs.

In general, CPF enables application developers to seamlessly access device features, such as camera, accelerometer. Additionally, almost all frameworks have capabilities of offering developers to create mobile applications that may address application requirements.

In addition, developer's challenges are not solely based on platform availabilities, but selection of type of platform that is suitable for fulfilling collaboration among stakeholders, building seamlessly on the fly, testing against real devices, backend services utilization and high performance, while device resources, such as camera, GPS, on/offline capabilities, data synchronization, application portability, and discoverabilities can be accessed through APIs.

Further, in this thesis, deep analysis about cloud based CPFs have been made and tested against well-defined criteria, while comparing to each other accordingly. As a result, not all cloud based CPFs completely integrate to the cloud, such as accessing cloud environment not only in the development but also testing phase, while stakeholders can seamlessly collaborate. In addition, well integrated cloud based CPFs provide application building and deployment on the

cloud against several device platforms, while some of them even provide distribution services to app stores.

Finally, this thesis has also provided not only a deep analysis about existence of fragmentation challenges in mobile application development, but also proved that fragmentation challenges can also be solved with CPFs available in the market, while cloud based CPFs are superior in terms of back-end services, such as notifications, security, collaborations, building, deployment and distributions to app stores.

## REFERENCES:

- [1] Joseph Sacha, 2013, Web Development in a Fragmented World,  
<http://groups.adobe.com/index.cfm?event=post.display&postid=46458> [online],  
 Accessed: July 2014
- [2] Fact Sheet, Introduction to Cloud Computing, [https://www.priv.gc.ca/resource/fs-fi/02\\_05\\_d\\_51\\_cc\\_e.pdf](https://www.priv.gc.ca/resource/fs-fi/02_05_d_51_cc_e.pdf)
- [3] Peter Mell, Timothy Grance , Sep,2011, Recommendations of the National Institute of Standards and Technology, The NIST Definition of Cloud Computing,  
<http://faculty.winthrop.edu/domanm/csci411/Handouts/NIST.pdf>
- [4] What are service models in cloud computing? , <http://www.cloud-competence-center.com/understanding/cloud-computing-service-models/> [online],. Accessed: October 2014
- [5] Software as a Service (SaaS), Feb, 2009, Blackbaud,  
[https://www.blackbaud.com/files/resources/downloads/DataSheet\\_SaaS-FAQ.pdf](https://www.blackbaud.com/files/resources/downloads/DataSheet_SaaS-FAQ.pdf)
- [6] SaaS: Software as a Service, <https://www.salesforce.com/saas/> [online], Accessed: October 2014
- [7] Platform as a service, [http://en.wikipedia.org/wiki/Platform\\_as\\_a\\_service](http://en.wikipedia.org/wiki/Platform_as_a_service), [online],  
 Accessed: October 2014
- [8] Shyam Kotecha, Platform as a Service(PaaS),  
[http://www.ieee.ldrp.ac.in/index.php?option=com\\_phocadownload&view=category&download=4:pdf&id=1:workshop&Itemid=216](http://www.ieee.ldrp.ac.in/index.php?option=com_phocadownload&view=category&download=4:pdf&id=1:workshop&Itemid=216)
- [9] Cloud Computing: What is Infrastructure as a Service, technet Magazine October, 2011,  
<http://technet.microsoft.com/en-us/magazine/hh509051.aspx> , [online], Accessed: Sep, 2014

- [10] NetBiscuits, “Mobile Web Metric report H2/2011”  
[http://www.netbiscuits.com/external\\_content/marketing/Netbiscuits\\_Mobile%20Web%20Metrics%20Report\\_2011-H2.pdf](http://www.netbiscuits.com/external_content/marketing/Netbiscuits_Mobile%20Web%20Metrics%20Report_2011-H2.pdf) (accessed 10.12.2011)
- [11] Rovio Blog, “Angry bird’s Android feedback”,  
<http://www.rovio.com/en/news/blog/51/angry-birds-android-feedback/> (accessed 10.11.2011)
- [12] Google Official Blog, “Supercharging Android: Google to Acquire Motorola Mobility”, August 15, 2011,  
<http://googleblog.blogspot.fi/2011/08/supercharging-android-google-to-acquire.html>, (accessed 12.12.2011)
- [13] NetBiscuits, “Mobile Web Metric report H2/2011”  
[http://www.netbiscuits.com/external\\_content/marketing/Netbiscuits\\_Mobile%20Web%20Metrics%20Report\\_2011-H2.pdf](http://www.netbiscuits.com/external_content/marketing/Netbiscuits_Mobile%20Web%20Metrics%20Report_2011-H2.pdf), , [online], Accessed: Dec, 2011
- [14] *ibid.*
- [15] RogerRob, “Rogers Catalyst Beta: Open APIs to create more opportunities”, RedBoard, 12.Nov.2010, <http://redboard.rogers.com/2010/rogers-catalyst-beta-open-apis-to-create-more-app-ortunities/>, [online], Accessed: Dec, 2011
- [16] BlueVia, “Payment API”, <https://bluevia.com/en/page/tech.APIs.PaymentAPI>, [online], Accessed: Dec, 2011
- [17] Orange Partner “an API to pay for your services on the Orange bill”, 20.Nov.2012, <http://www.orangepartner.com/content/api-click-buy>, [online], Accessed: Dec, 2011
- [18] Vodafone, <http://developer.vodafone.com/>, [online], Accessed: Dec, 2011
- [19] GSMA, “gsmworld OneApi”, <http://oneapi.gsmworld.com/> [online], Accessed: Dec.2011)
- [20] WACAPI “WAC”,<http://wacapis.wacapps.net/index.html> [online], Accessed: Dec.2011)
- [21] Olafur Ingthorsson, “Can the mobile cloud resolve fragmentation?”, Data Center Knowledge, 31.May.2011,

- <http://www.datacenterknowledge.com/archives/2011/05/31/can-the-mobile-cloud-resolve-fragmentation/> [online], Accessed: Dec.2011)
- [22] <http://modernizr.com/> [online], Accessed: Dec.2013)
  - [23] Shaubik Roy Choudhary, 2014 Cross-Platform Testing and Maintenance of Web and Mobile Applications. DOI = 10.1145/2431211.2431216  
<http://dx.doi.org/10.1145/2591062.2591097>
  - [24] Peeyush Mathur, Nikihil Nishchal, iCloud Computing: New Challenge to the entire computer industry, 1st International Conference on Parallel, Distributed and Grid computing (PDGC-2010),978-1-4244-7674-9/©2010 IEEE
  - [25] Malathi, M., "Cloud computing concepts," Electronics Computer Technology (ICECT), 2011 3rd International Conference on , vol.6, no., pp.236,239, 8-10 April 2011 doi: 10.1109/ICECTECH.2011.5942089
  - [26] Dong Xu, "Cloud Computing: an Emerging Technology", International Conference On Computer Design and Applications (ICCD 2010), Volume-1, Pgs (100-104)
  - [27] Ibid.
  - [28] Malathi, M., "Cloud computing concepts," Electronics Computer Technology (ICECT), 2011 3rd International Conference on , vol.6, no., pp.236,239, 8-10 April 2011 doi: 10.1109/ICECTECH.2011.5942089
  - [29] D. Durkee, "Why Cloud Computing Will Never Be Free," Comm. ACM, vol. 53, no. 5, 2010, pp. 62–69.
  - [30] Shuai Zhang, Shufen Zhang, Xuebin Chen, Xiuzhen Huo, "Cloud Computing Research and Development Trend", Second International Conference on Future Networks, 2010 IEEE
  - [31] Dong Xu, "Cloud Computing: an Emerging Technology", International Conference On Computer Design and Applications (ICCD 2010), Volume-1, Pgs (100-104)
  - [32] Infoworld, special report, cloud computing deep dive, September 2009, pp.9-10.

- [33] Malathi, M., "Cloud computing concepts," Electronics Computer Technology (ICECT), 2011 3rd International Conference on , vol.6, no., pp.236,239, 8-10 April 2011 doi: 10.1109/ICECTECH.2011.5942089
- [34] Hong He, "Applications deployment on the SaaS platform," Pervasive Computing and Applications (ICPCA), 2010 5th International Conference on , vol., no., pp.232,237, 1-3 Dec. 201 doi: 10.1109/ICPCA.2010.5704104
- [35] Pang Xiong Wen; Li Dong, "Quality Model for Evaluating SaaS Service," Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference on , vol., no., pp.83,87, 9-11 Sept. 2013 doi: 10.1109/EIDWT.2013.19
- [36] Kulkarni, G.; Khatawkar, P.; Shelke, R.; Solanke, V.; Waghmare, R., ""Multi-tenant SaaS cloud"," AFRICON, 2013 , vol., no., pp.1,4, 9-12 Sept. 2013 doi: 10.1109/AFRCON.2013.6757684
- [37] Peeyush Mathur, Nikihil Nishchal, iCloud Computing: New Challenge to the entire computer industry, 1st International Conference on Parallel, Distributed and Grid Computing (PDGC-2010),978-1-4244-7674-9/©2010 IEEE
- [38] NIST Definition of Cloud Computing (September 2011), National Institute of Standards and Technology Special Publication 800-145.
- [39] Mitesh Soni. 2014. Cloud computing basics—platform as a service (PaaS). Linux J. 2014, 238, pages.
- [40] NetSuite, SuiteCloud Platform, The Ultimate Business Cloud Platform, <http://www.netsuite.com.hk/portal/common/pdf/ds-suitecloud.pdf>
- [41] Salesforce, PaaS: Platform as a Service, [online], <https://www.salesforce.com/paas/overview/>
- [42] NIST Definition of Cloud Computing (September 2011), National Institute of Standards and Technology Special Publication 800-145



- [43] Dan Orlando, "Cloud computing service models, Part 1: Infrastructure as a Service" Jan 2011[Online], <http://www.ibm.com/developerworks/cloud/library/cl-cloudservices1iaas>[accessed: 2014]
- [44] Amazon Web Services, "Case studies," Amazon web site, Aug 2014, [Online] <http://aws.amazon.com/solutions/case-studies/> [online], Accessed: 2014
- [45] Pankaj Arora, Rubal Chaudhry Wadhawan, Er. Satinder Pal Ahuj "Cloud Computing Security Issues in Infrastructure as a Service" , International Journal of Advanced Research in Computer Science and Software Engineering, Jan 2012
- [46] D. Tinnelvely. "Microsoft and HP to Tap into Overlooked Private Cloud Market". <http://cloudcomputing.sys-con.com/node/1249863> [online], Accessed: Dec 2011
- [47] [http://www.computerworld.com/s/article/353080/The\\_Bumpy\\_Road\\_to\\_Private\\_Clouds](http://www.computerworld.com/s/article/353080/The_Bumpy_Road_to_Private_Clouds) (accessed 29.11.2011]
- [48] ibid.
- [49] P. Hofmann and D. Woods, "Cloud Computing: The Limits of Public Clouds for Business Applications", presented at IEEE Internet Computing, 2010, p.93.
- [50] Ibid.
- [51] Lars Arne Skaar, Ruth Lennon, E. Michael Maximilien, and Arne Jorgen Berre. 2010. Building the right cloud solutions. In Proceedings of the ACM international conference companion on Object oriented programming systems languages and applications companion (SPLASH '10). ACM, New York, NY, USA, 287-288.DOI=10.1145/1869542.1869616 <http://doi.acm.org/10.1145/1869542.1869616>
- [52] Peter Mell,Timothy Grance, September 2011 The NIST Definition of Cloud Computing, Recommendations of the National Institute of Standards and Technology, <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>

- [53] Mark Vaughn, Key differences between virtualization and cloud computing,  
[http://docs.media.bitpipe.com/io\\_10x/io\\_100683/item\\_428385/Intel\\_sServerVirtualization\\_IO%23100683\\_E-Guide\\_062811.pdf](http://docs.media.bitpipe.com/io_10x/io_100683/item_428385/Intel_sServerVirtualization_IO%23100683_E-Guide_062811.pdf)
  
- [54] Microsoft Private Cloud, Cloud Power, August 2011,  
[http://www.microsoft.com/investor/downloads/events/Microsoft\\_Private\\_Cloud\\_Whitepaper.pdf](http://www.microsoft.com/investor/downloads/events/Microsoft_Private_Cloud_Whitepaper.pdf)
  
- [55] AeroHive network, Public or Private Cloud: The Choice is Yours,  
<http://www.aerohive.com/pdfs/Aerohive-Whitepaper-Public-or-Private-Cloud.pdf>
  
- [56] Melillo Consulting, Infrastructure : Private Cloud Building Blocks,  
<http://www.mjmc.com/infrastructure/private-cloud-building-blocks>
  
- [57] Cisco, Managing the Real Cost of On-Demand Enterprise Cloud Services with Chargeback Models,  
[http://www.techdata.com/content/tdcloud/files/cisco/Cloud\\_Services\\_Chargeback\\_Models\\_White\\_Paper.pdf](http://www.techdata.com/content/tdcloud/files/cisco/Cloud_Services_Chargeback_Models_White_Paper.pdf)
  
- [58] Shamim Hossain, 2014, Thoughts On Cloud, Cloud computing basics,  
<http://thoughtsoncloud.com/2014/02/cloud-computing-basics/#idc-container>
  
- [59] Mike Klein, Three Benefits of public cloud computing, Online Tech, OTBlog, 28, April, 2011, <http://resource.onlinetech.com/three-benefits-of-public-cloud-computing/> [online], Accessed: Dec.2011
  
- [60] Sun microsystems, Introduction to cloud computing architecture, white paper, 1st edition, June 2009,  
<http://ereseach.wiki.otago.ac.nz/images/7/75/Cloudcomputing.pdf>, page 9, [online], Accessed: Dec.2011
  
- [61] Hofmann and D. Woods, "Cloud Computing: The Limits of Public Clouds for Business Applications", presented at IEEE Internet Computing, 2010, p.91
  
- [62] P.C. Heckel, Hybrid Clouds: Comparing Cloud Toolkits, Seminar Paper Business Informatics, University of Mannheim, 2010.

- [63] Bob Violino, 2013, Why the Hybrid Cloud Offers Companies a Leg Up in Enterprise IT , <http://www.lunarpages.com/uptime/why-hybrid-cloud-offers-companies-leg-enterprise-it>, [online], Accessed: June 2013
- [64] VmWare Bluerock, Security in the hybrid cloud, putting rumors to rest, <http://www.frost.com/prod/servlet/cpo/272112250>, Accessed: Dec.2013
- [65] Antony Savvas, 2014, The benefits of hybrid cloud computing, <http://www.itproportal.com/2014/05/14/the-benefits-of-hybrid-cloud-computing/>
- [66] Sujay.R June, 2011, Hybrid Cloud: A New Era, <http://www.ijcst.com/vol22/2/sujayR.pdf>
- [67] Briscoe, G., & Marinos, A. (2009). Towards community cloud computing. Proceedings of the IEEE Digital Ecosystems and Technologies (DEST 2009), Istanbul, Turkey
- [68] Garlick, G., "Improving Resilience with Community Cloud Computing," Availability, Reliability and Security (ARES), 2011 Sixth International Conference on , vol., no., pp.650,655, 22-26 Aug. 2011 doi: 10.1109/ARES.2011.100
- [69] Y. Benkler, "Sharing nicely: on shareable goods and the emergence of sharing as a modality of economic production," The Yale Law Journal, vol. 114, no. 2, pp. 273-359, 2004.
- [70] Ahson, Syed A., and Mohammad Ilyas (eds). "Chapter 1 - Understanding the Cloud Computing Landscape". Cloud Computing and Software Services: Theory and Techniques. Auerbach Publications. © 2011. Books24x7.  
<<http://common.books24x7.com/toc.aspx?bookid=36915>> [online], Accessed: June, 2012
- [71] Sosinsky, Barrie. "Chapter 3 - Understanding Cloud Architecture". Cloud Computing Bible. John Wiley & Sons. © 2011. Books24x7.  
<<http://common.books24x7.com/toc.aspx?bookid=40611>> [online], Accessed: December 2012
- [72] Ibid.

- [73] G. J. Popek and R. P. Goldberg, "Formal requirements for virtualizable third generation architectures", *Communications of ACM*, vol. 17, no. 7, 1974, pp. 412-421
- [74] P. Barham et al, "Xen and the art of virtualization", in *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP'03)*, New York, USA, 19-22 October 2003, pp. 164-177.
- [75] Sosinsky, Barrie. "Chapter 3 - Understanding Cloud Architecture". *Cloud Computing Bible*. John Wiley & Sons. © 2011. Books24x7.  
<<http://common.books24x7.com/toc.aspx?bookid=40611>> [online], Accessed: Feb, 2012
- [76] Ibid.
- [77] Won Kim. Cloud architecture: a preliminary look. In *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services*, New York, NY, USA, 2011. ACM.
- [78] Fang Liu et al., "NIST Cloud Computing Reference Architecture (September 2011)", National Institute of Standards and Technology Special Publication 500-292.
- [79] Gartner, Press Release "Gartner Says Cloud Consumers Need Brokerages to Unlock the Potential of Cloud Services", July 9, 2009,  
<http://www.gartner.com/it/page.jsp?id=1064712>. [online], Accessed: Feb.2012
- [80] Intel It center, 2013, Planngng Guide, Virtualization and Cloud Computing,  
<http://www.intel.com/content/dam/www/public/us/en/documents/guides/cloud-computing-virtualization-building-private-iaas-guide.pdf>
- [81] Jan Kremer, Cloud Computing and Virutalization,  
<http://jkremer.com/White%20Papers/Cloud%20Computing%20and%20Virtualization%20White%20Paper%20JKCS.pdf>
- [82] Pek, G., Butty ´ an, L., and Bencs ´ ath, B. 2013. A survey of security issues in hardware virtualization. *ACM ´ Comput. Surv.* 45, 3, Article 40 (June 2013), 34 pages. DOI:  
<http://dx.doi.org/10.1145/2480741.2480757>

- [83] Pearce, M., Zeadally, S., and Hunt, R. 2013. Virtualization: Issues, security threats, and solutions. *ACM Comput. Surv.* 45, 2, Article 17 (February 2013), 39 pages. DOI = 10.1145/2431211.2431216 <http://doi.acm.org/10.1145/2431211.2431216>
- [84] wing, Types of Virtualization Technology-Advantages vs. Disadvantages - <http://securitywing.com/types-virtualization-technology/>, [Online] Accessed: Jan, 2014
- [85] Wikipedia contributors, " Hardware virtualization," Wikipedia, The Free Encyclopedia, [http://en.wikipedia.org/wiki/Hardware\\_virtualization](http://en.wikipedia.org/wiki/Hardware_virtualization) [Online], Accessed: Feb 2014]
- [86] Crunchertronis, Partial virtualization – type of hardware virtualization, <http://www.crunchertronics.com/partial-virtualization-type-hardware-virtualization/> [Online], Accessed: Feb 2014]
- [87] Federico Pareschi, 2013, Applying Partial Virtualization on Elf Binaries Through Dynamic Loaders, [http://amslaurea.unibo.it/5065/1/pareschi\\_federico\\_tesi.pdf](http://amslaurea.unibo.it/5065/1/pareschi_federico_tesi.pdf), [Online], Accessed: March 2014]
- [88] Virtualization softwares, What is Paravirtualization, <http://www.virtualizationsoftwares.com/what-is-paravirtualization/> [Online] Accessed: March 2014
- [89] Smith, J. and Nair, R. *Virtual Machines*, Morgan Kaufmann, 2005
- [90] Heradon Douglas and Christian Gehrman. Secure virtualization and multicore platforms state-of-the-art report. SICS Technical Report, T2009:14A:1-71, Jan 2010
- [91] Kim Hazelwood. 2011. Process-level virtualization for runtime adaptation of embedded software. In *Proceedings of the 48th Design Automation Conference (DAC '11)*. ACM, New York, NY, USA, 895-900. DOI=10.1145/2024724.2024924
- [92] Oren Laadan and Jason Nieh. 2010. Operating system virtualization: practice and experience. In *Proceedings of the 3rd Annual Haifa Experimental Systems Conference (SYSTOR '10)*. ACM, New York, NY, USA, Article 17 , 12 pages. DOI=10.1145/1815695.1815717

- [93] Wikipedia contributors, "Paravirtualization ," Wikipedia, The Free Encyclopedia, <http://en.wikipedia.org/wiki/Paravirtualization> [Online] Accessed: March 2014]
- [94] Brandon D. Shroyer ,  
[http://www.cs.rochester.edu/~sandhya/csc256/seminars/bshroyer\\_virtualmachines.pdf](http://www.cs.rochester.edu/~sandhya/csc256/seminars/bshroyer_virtualmachines.pdf)
- [95] Wikipedia contributors, " Virtual machine," Wikipedia, The Free Encyclopedia, [http://en.wikipedia.org/wiki/Virtual\\_machine](http://en.wikipedia.org/wiki/Virtual_machine) [Online] Accessed: Feb, 2014
- [96] SUSE Linux Enterprise Server Virtualization with Xen , 2012,  
[http://doc.opensuse.org/products/draft/SLES/SLES-xen\\_sd\\_draft/index.html](http://doc.opensuse.org/products/draft/SLES/SLES-xen_sd_draft/index.html), [Online]  
Accessed: May, 2014
- [97] J LeVasseur, V Uhlig, B Leslie, M Chapman, G Heiser, "Pre-virtualization: uniting two worlds", Proceedings of the twentieth ACM symposium on Operating systems principles.
- [98] OLAFUR INGTHORSSON, 2011, Cloud Computing, The Mobile Cloud: What it Is& Why it Matters,  
<http://www.datacenterknowledge.com/archives/2011/05/09/the-mobile-cloud-what-it-is-why-it-matters/>
- [99] Guan, Le; Ke, Xu; Song, Meina; Song, Junde, "A Survey of Research on Mobile Cloud Computing," Computer and Information Science (ICIS), 2011 IEEE/ACIS 10th International Conference on , vol., no., pp.387,392, 16-18 May 2011 doi: 10.1109/ICIS.2011.67
- [100] Chetan S., Gautam Kumar, K. Dinesh, Mathew K., and Abhimanyu M.A. CloudComputing for Mobile World. 2010; Available from: <http://chetan.ueuo.com/projects/CCMW.pdf>.
- [101] Sheng Xiao and Weibo Gong. Mobility Can Help: Protect User Identity with Dynamic Credential. in Mobile Data Management (MDM), 2010 Eleventh International Conference on. 2010

- [102] Cloud computing advices, 5 Essential Features of Cloud Computing,  
<http://cloudcomputingadvices.com/cloud-computing-features/> [Online] Accessed: Feb 2014]
- [103] K. Yang, S. Ou, and H.H. Chen, "On Effective Offloading Services for Resource-Constrained Mobile Devices Running Heavier Mobile Internet Applications," IEEE Comm. Magazine, vol. 46, no. 1, 2008, pp.
- [104] Allen Liao, 2010, Cloud Computing for mobile users: can offloading computation save energy?,[http://voiplab.niu.edu.tw/ppt/98Internet\\_Exchange/R9843005.pdf](http://voiplab.niu.edu.tw/ppt/98Internet_Exchange/R9843005.pdf)
- [105] Kim, S.; Yonglk Yoon; Minkyoo In; Kangchan Lee; Seungyun Lee; , "The evolution of standardization for mobile cloud," ICT Convergence (ICTC), 2011 International Conference on , vol., no., pp.578-579, 28-30 Sept. 2011
- [106] Visiongain report "MOBILE CLOUD COMPUTING INDUSTRY OUTLOOK REPORT: 2011-2016", p.126 <http://www.visiongain.com/Report/737/Mobile-Cloud-Computing-Industry-Outlook-Report-2011-2016> [Online] Accessee, Feb, 2014
- [107] Ibid.
- [108] Zhang et al., "Securing elastic applications on mobile devices for cloud computing," Proceedings of the 2009 ACM workshop on Cloud computing security, 2009.
- [109] Rackspace, <http://www.rackspace.com/> , [Online] Accessed: May, 2014
- [110] Amazon EC2, <http://aws.amazon.com/ec2/> [Online] Accessed: May, 2014
- [111] Mahadev Satyanarayanan; Bahl, P.; Caceres, R.; Davies, N, "The Case for VM-based Cloudlets in Mobile Computing," Pervasive Computing, IEEE , vol.PP, no.99, pp.1,1, doi: 10.1109/MPRV.2009.6
- [112] Ericson alliance with Akamai " new approach to mobility" , Feb,14,2011,  
<http://www.akamai.com/ericsson> [Online] Accessed: Feb, 2013
- [113] Mahadev Satyanarayanan, Bahl, P., Caceres, R., Davies and N., "The Case for VM-Based Cloudlets in Mobile Computing", vol. 8, no. 4, pp. 14-23, Oct.-Dec., 2009

- [114] Jararweh, Y.; Tawalbeh, L.; Ababneh, F.; Dosari, F., "Resource Efficient Mobile Computing Using Cloudlet Infrastructure," Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on , vol., no., pp.373,377, 11-13 Dec. 2013 doi: 10.1109/MSN.2013.75
- [115] Mahadev Satyanarayanan. 2011. Mobile computing: the next decade. SIGMOBILE Mob. Comput. Commun. Rev. 15, 2 (August 2011), 2-10.  
DOI=10.1145/2016598.2016600
- [116] Byung-Gon Chun, Sunghwan Ihm, Petros Maniatis, Mayur Naik, and Ashwin Patti. 2011. CloneCloud: elastic execution between mobile device and cloud. In Proceedings of the sixth conference on Computer systems (EuroSys '11). ACM, New York, NY, USA, 301-314. DOI=10.1145/1966445.1966473
- [117] Chetan S., Gautam Kumar, K. Dinesh, Mathew K., and Abhimanyu M.A. "Cloud Computing for Mobile World. 2010 <http://chetan.ueuo.com/projects/CCMW.pdf>.
- [118] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling public verifiability and data dynamics for storage security in cloud computing," in European Symposium on Research in Computer Security (ESORICS) 2009, Saint Malo, France, Sep 2009
- [119] C. Wang, K. Ren, W. Lou, and J. Li, "Towards Publicly Auditably Secure Data Storage Services," IEEE Network Magazine, vol. 24, no. 4, pp. 19-24, July/August 2010.
- [120] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," Wireless Communications and Mobile Computing, vol. 13, no. 18, pp. 1587–1611, 2013
- [121] Joorabchi, Mona Erfani, Ali Mesbah, and Philippe Kruchten. "Real challenges in mobile app development." Empirical Software Engineering and Measurement, 2013  
ACM/IEEE International Symposium on. IEEE, 2013.
- [122] Oracle Mobile Application Framework, <http://www.oracle.com/technetwork/developer-tools/maf/documentation/oraclemaf-datasheet-2220993.pdf>



- [123] C Sharp (programming language),  
[http://en.wikipedia.org/wiki/C\\_Sharp\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/C_Sharp_(programming_language))
- [124] Java (programming language),[http://en.wikipedia.org/wiki/Java\\_\(programming\\_language\)](http://en.wikipedia.org/wiki/Java_(programming_language)) [Online]  
Accessed: Feb, 2014
- [125] Lua, <http://www.lua.org/about.html>, [Online] Accessed: Feb, 2014
- [126] WebView for Android, <https://developer.chrome.com/multidevice/webview/overview>,  
[Online] Accessed: May, 2014
- [127] Sysbase “Sysbase unwired platform” ,  
<http://www.sysbase.com/products/mobileenterprise/sysbaseunwiredplatform> , [Online]  
Accessed: May, 2014
- [128] Corona SDK, <http://www.coronalabs.com/products/corona-sdk> , [Online] Accessed:  
May, 2014
- [129] Flex platform “One code base, multiple devices”,  
<http://www.adobe.com/products/flex.html> [Online] Accessed: Feb, 2014
- [130] Titanium mobile “Native iOS, Android, Hybrid, and Mobile Web Application  
Development” <http://www.appcelerator.com/platform/titanium-sdk/> [Online] Accessed:  
March, 2014
- [131] PhoneGap Build, “ Adope Phonegap Build”, <https://build.phonegap.com/> [Online]  
Accessed: Nov.2014)
- [132] Gartner Technical Professional Advice: Cross-Platform Mobile Development  
Frameworks Published: 14 November 2011
- [133] Gartner Technical Professional Advice: Mobile Application Frameworks Published: 15  
March 2012, p.7-23
- [134] Cloud Times: “Mobile Cloud”, <http://cloudtimes.org/mobile-cloud>, [Online] Accessed:  
Jan, 2014

- [135] ABI research,” 2.1 Billion HTML5 Browsers on Mobile Devices by 2016 says ABI Research”, <http://www.abiresearch.com/press/21-billion-html5-browsers-on-mobile-devices-by-201>, [Online] Accessed: Mar, 2014
- [136] Cloud Times: “Mobile Cloud”, <http://cloudtimes.org/mobile-cloud> , [Online] Accessed: Feb, 2014
- [137] W3C “Web Storage”8.Dec.2011, <http://www.w3.org/TR/webstorage/> [Online] Accessed: Apr, 201
- [138] W3C “Web SQL Database, 18.Nov.2010, <http://www.w3.org/TR/webdatabase/> [Online] Accessed: Apr, 2014
- [139] W3C “Indexed Database API “, 24.May.2012, <http://www.w3.org/TR/2012/WD-IndexedDB-20120524/>[Online] Accessed: Apr, 2014
- [140] W3C “File API: Directories and System” 07.March.2012  
<http://dev.w3.org/2009/dap/file-system/file-dir-sys.html> , [Online] Accessed: Oct, 2014
- [141] W3C “Media elements” 14.Dec.2012,  
<http://www.w3.org/html/wg/drafts/html/master/embedded-content-0.html#media-elements> , [Online] Accessed: Oct, 2014
- [142] W3C “Application caches “<http://www.w3.org/TR/2011/WD-html5-20110525/offline.html>,[Online] [Online] Accessed: Oct, 2014
- [143] W3C “Page Visibility”, 26.July.2012, <http://www.w3.org/TR/page-visibility/#document-visibilitystate>, [Online] Accessed: July, 2014
- [144] Amazon “Kindle Format 8”,  
<http://www.amazon.com/gp/feature.html?ie=UTF8&docId=1000729511> [Online] Accessed: July, 2014

- [145] Guardian Apps Blog “Financial Times passes 2m users for its HTML5 web app”  
<http://www.guardian.co.uk/media/appsblog/2012/apr/24/financial-times-web-app-2m>  
 [Online] Accessed: July, 2014
- [146] Julie Larson-Green, “Previewing Windows 8”, Microsoft News Center, 01. June.2011,  
<http://www.microsoft.com/en-us/news/features/2011/jun11/06-01corpnews.aspx>,  
 [Online] Accessed: Oct, 2014
- [147] Matt Rosoff “Facebook Buys a Company to Help It Master Mobile Applications”,  
 Business Insider, 8.Nov.2011, <http://www.businessinsider.com/facebook-snags-new-talent-to-help-with-mobile-push-2011-11>, [Online] Accessed: July, 2014
- [148] PhoneGap “Getting Started Guides”, [http://docs.phonegap.com/en/2.2.0/guide\\_getting-started\\_index.md.html](http://docs.phonegap.com/en/2.2.0/guide_getting-started_index.md.html), [Online] Accessed: July, 2014
- [149] PhoneGap API Reference “Storage”,  
[http://docs.phonegap.com/en/2.2.0/cordova\\_storage\\_storage.md.html#Storage](http://docs.phonegap.com/en/2.2.0/cordova_storage_storage.md.html#Storage) [Online]  
 Accessed: July, 2014
- [150] Adobe phonegap build, Package mobile apps in the cloud, <https://build.phonegap.com>  
 Storage [Online] Accessed: Sep, 2014
- [151] Packaged Web Apps, <http://www.w3.org/TR/widgets> [Online] Accessed: Sep, 2014
- [152] Adobe® PhoneGap™ Build!, <https://build.phonegap.com/apps> [Online], Accessed:  
 Sep, 2014
- [153] API v1 Overview, <https://build.phonegap.com/docs/api> [Online], Accessed: July, 2014
- [154] Hydration, <https://build.phonegap.com/docs/hydration>, [Online], Accessed: July, 2014
- [155] Remote Collaboration and Testing, [http://docs.phonegap.com/en/edge/guide\\_phonegap-build\\_test\\_index.md.html#Remote%20Collaboration%20and%20Testing](http://docs.phonegap.com/en/edge/guide_phonegap-build_test_index.md.html#Remote%20Collaboration%20and%20Testing) [Online],  
 Accessed: Oct, 2014
- [156] Rhomobile Suite, <http://www.motorola.com/Business/US-EN/Business+Product+and+Services/Software+and+Applications/RhoMobile+Suite>, [Online], Accessed: Oct, 2014

- [157] Build Native Apps in the Cloud, <https://rms.rhomobile.com/>, [Online], Accessed: Oct, 2014
- [158] Ibid,
- [159] MoSync Reload, <http://www.mosync.com/reload>, [Online], Accessed: Sep, 2014
- [160] Stepfano, January 13, 2013, cloudbase.io MoSync helper class,<http://cloudbase.io/blog/2013/01/cloudbase-mosync-helper-class/>, [Online], Accessed: Sep, 2014
- [161] Your cloud functions, <http://cloudbase.io/why/cloud-functions> [Online] Accessed: Sep, 2014
- [162] Download the Android helper class, Cloudbase.io, <http://cloudbase.io/documentation/android/get-started>, [Online] Accessed: Sep, 2014
- [163] Build apps in C# for iOS, Android and Windows Phone, <http://xamarin.com/platform> (accessed 11.7.2014)
- [164] Xamarin, Key Strategies for Mobile excellence, [http://cdn1.xamarin.com/webimages/assets/Xamarin\\_Whitepaper-Key\\_Strategies\\_for\\_Mobile\\_Excellence.pdf](http://cdn1.xamarin.com/webimages/assets/Xamarin_Whitepaper-Key_Strategies_for_Mobile_Excellence.pdf), [Online] Accessed: Sep, 2014
- [165] Power, Speed and Quality, Key Strategies for Mobile Excellence, p.8, [http://cdn1.xamarin.com/webimages/assets/Xamarin\\_Whitepaper-Key\\_Strategies\\_for\\_Mobile\\_Excellence.pdf](http://cdn1.xamarin.com/webimages/assets/Xamarin_Whitepaper-Key_Strategies_for_Mobile_Excellence.pdf) , [online] Accessed: July 2014
- [166] Ibid.
- [167] Introduction to Xamarin test cloud, <http://developer.xamarin.com/guides/testcloud/introduction-to-test-cloud/>, [Online], Accessed: July 2014
- [168] Xamarin test cloud devices, <https://testcloud.xamarin.com/devices>, [Online], Accessed: July 2014

- [169] Introduction to Xamarin Test Cloud,  
<http://developer.xamarin.com/guides/testcloud/introduction-to-test-cloud/> [Online]  
 Accessed: July 2014
- [170] Accelerate Mobile Innovation in the Enterprise. <http://appery.io/appbuilder/>  
 [Online] Accessed: Sep 2014
- [171] Creating mobile UI, <http://devcenter.appery.io/documentation/building-mobile-ui/>  
 [Online] Accessed: Sep 2014
- [172] Using the Twillio API and Sample App plugin,  
<http://devcenter.appery.io/documentation/plugins/using-twillio-api-and-sample-app-plugin-in-to-interact-with-voice-and-sms-data/>, [Online] Accessed: Sep 2014
- [173] Write once in JavaScript, run native on every major platform,  
<http://www.appcelerator.com/platform/apps/> [Online] , Accessed: Sep 2014
- [174] You can't manage what you can't measure,  
<http://www.appcelerator.com/platform/analytics/> [Online] , Accessed: Sep 2014
- [175] Great apps need great data, <http://www.appcelerator.com/platform/apis/>, [Online],  
 Accessed: Sep 2014
- [176] Native vs. HTML5 mobile application development, which option is best,  
<http://www.appcelerator.com.s3.amazonaws.com/pdf/appcelerator-whitepaper-native-html5.pdf>, p. 11 [online], Accessed: Sep, 2014
- [177] Modules.Facebook, <http://docs.appcelerator.com/titanium/3.0/#!/api/Modules.Facebook>,  
 [Online], Accessed: Sep 2014
- [178] The Appcelerator Salesforce Mobile Pack Connector ,  
<http://www.appcelerator.com/salesforce/>, [Online], Accessed: Oct, 2014
- [179] SAP and Appcelerator Integration, <http://www.appcelerator.com/sap/>, [Online],  
 Accessed: Sep 2014

- [180] Appcelerator Platform Services API for Android, <http://docs.appcelerator.com/aps-sdk-apidoc/latest/android/>, [Online], Accessed: Sep 2014
- [181] Architecture, <http://www.appcelerator.com/platform/alloy/>, [Online], Accessed: July, 2014
- [182] Appcelerator Cloud Services SDKs,  
<http://docs.appcelerator.com/cloud/latest/#!/guide/sdks>, [Online], Accessed: Oct, 2014
- [183] Appcelerator Platform Services API for Android, <http://docs.appcelerator.com/aps-sdk-apidoc/latest/android/>, [Online], Accessed: Sep, 2014
- [184] The Telerec Platform, <http://www.telerik.com/platform>, [Online], Accessed: Sep, 2014
- [185] Analytics, Solutions for Mobile Application Development,  
<http://www.telerik.com/cloud-development>, [Online], Accessed: Sep, 2014
- [186] Scale With Unified Cloud Infrastructure.<http://www.telerik.com/platform#scale-with-unified-cloud-infrastructure>, [online], Accessed: Sep, 2014
- [187] Supported Frameworks,  
<http://docs.telerik.com/platform/appbuilder/introduction/supported-frameworks/working-with-frameworks>, [online], Accessed: Sep, 2014
- [188] AppFeedback, The easiest way to get actionable feedback from your users about their mobile app experience, <http://www.telerik.com/appfeedback>, [online], Accessed: Sep, 2014
- [189] Kony Studio, Developer tools and features,  
<http://www.kony.com/products/development/features>, [online], Accessed: Oct, 2014
- [190] Ibid.
- [191] Kony MobileFabric, Open, standards-based development,  
<http://www.kony.com/products/mbaas/features>, [online], Accessed: Oct, 2014

- [192] Kony Visualizer, Sharing and collaboration features,  
<http://www.kony.com/products/visualization/collaboration>, [online], Accessed: Oct, 2014
- [193] Kony Multi-Channel Experience Cloud, Kony,  
<http://forms.kony.com/rs/konysolutions/images/DS-Kony-Experience-Cloud-Datasheet.pdf>, [online], Accessed: Oct, 2014
- [194] The open platform for user-first, multi-channel experiences.  
<http://www.kony.com/products/experience-platform> [online], Accessed: July 2014
- [195] Kony Cloud, Architecture and Security Overview.  
<http://forms.kony.com/rs/konysolutions/images/Kony-DS-Architecture-and-Security-Overview.pdf>, [Online]. Accessed: July 2014
- [196] Kony Studio, Messaging and notification features,  
<http://www.kony.com/products/development/messaging>, [Online]. Accessed: July 2014
- [197] Ibid.
- [198] Kony MBaaS (Mobile Backend-As-A-Service), Kony,  
<http://forms.kony.com/rs/konysolutions/images/DS-Kony-Discover-Enterprise-Grade-Cloud-Services.pdf>, [Online]. Accessed: Sep 2014
- [199] Flexible deployment options, Kony, <http://www.kony.com/products/deployment-options>, [Online]. Accessed: Sep 2014
- [200] Kony Cloud, Architecture and Security Overview,  
<http://forms.kony.com/rs/konysolutions/images/Kony-DS-Architecture-and-Security-Overview.pdf>, [Online]. Accessed: Sep 2014
- [201] Start Building, <https://www.parse.com/docs>, [Online]. Accessed: Sep 2014
- [202] Parse. <https://parse.com/products/core>. [Online]. Accessed: Oct, 2014
- [203] Push Notification Guide, [https://parse.com/docs/push\\_guide](https://parse.com/docs/push_guide), [Online]. Accessed: Oct, 2014

- [204] Parse Analytics, <https://parse.com/products/analytics>, [Online]. Accessed: Oct, 2014