First International Workshop on Mobile Cloud Computing Systems, Management, and Security (MCSMS-2015)

Issues in adopting Agile Development Principles for Mobile Cloud Computing Applications

Fahad Almudarra*, Basit Qureshi

College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia

Abstract

Mobile Cloud Computing has been a recent research and development area. Various framework and middleware have been developed recently providing solutions for utilizing cloud computing resources on resource constrained mobile devices. A very few number of works present sound engineering mechanism for developing such software. This paper discusses issues in developing mobile cloud applications using agile software development methodologies. Sha-Mo-Cloud, a hybrid cloud framework was developed leveraging public cloud services for storing data including images and videos captured from user’s mobile devices. Users can upload, access, manage and share personal content using web interface as well as a mobile application. To preserve user privacy and personalized data, part of the framework was developed in a private cloud environment. This work also reports the engineering aspects of development of ShaMoCloud. The software construction team previously developed similar applications using Team Software Process. The work presented here, compares and contrasts the experiences of the ShaMoCloud development team with four similar projects carried out recently.

© 2015 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of the Conference Program Chairs

Keywords: Mobile Cloud Computing; Mobile Social Networks; Content Management; Privacy; Agile development methods;

1. INTRODUCTION

In recent years there is a trend in migrating popular applications to the cloud with social networking applications such as Facebook, twitter etc. amongst many others. Many software giants have been offering their software and services in the cloud such as Google-docs\(^2\), Gmail\(^3\), MS-Office 360\(^4\) etc. When a Cloud is made available as a pay-as-you-go manner in the public domain, it is called a Public Cloud; the service being sold is Utility Computing\(^1\). Microsoft

* Corresponding author. Email address: fahad.almudarra@gmail.com
Azure⁵, Google App Engine⁶, Amazon EC2⁷ and recently IBM CloudOne⁸ have been providing Software-as-a-Service (SaaS) and Platform-as-a-service (PaaS) for developers to build applications using corporate infrastructure.

From a software engineering perspective, cloud computing is a potential paradigm shift following the major paradigm shift from mainframe based computing to client-server computing in the early 1980s. When cloud computing becomes mobile, it realizes the mobile device communication with a data center to access a shared computing resource or storage on demand. Where research in cloud based services have been extensive in recent years, very little effort is done in planning, designing, developing and testing applications for mobile devices and their integration with the cloud.

Mobile cloud computing utilizes the ever-increasing pool of computing resources available in the cloud. In mobile cloud computing, mobile devices can rely on cloud computing and information storage resource to perform computationally intensive operations such as searching, data mining, and multimedia processing²¹. Various frameworks, middleware and applications have been developed in recent years providing storage and computation as cloud services on mobile devices⁹, ¹³-¹⁴. Authors¹¹-¹² provide a survey of recent advances in Mobile Cloud computing. Despite all recent works, trust, security and privacy is an ongoing issue²³. In spite of various efforts to overcome these, there are a number of loopholes and challenges that still exist in the security policies of mobile cloud computing.

This research explores two aspects of mobile cloud application development:
1) Design and implementation of a hybrid cloud environment to satisfy the requirements of a mobile cloud content management and sharing framework, preserving the privacy requirements.
2) Study the various aspects of developing a mobile cloud application using extreme programming as an agile development methodology. Integrate usage of cloud technologies in software team development, integration and testing.

The ShaMoCloud framework is developed as a case study to address the above mentioned two aspects. The framework is composed of a Cloud Service integrating public and private cloud. A web interface is developed to interact with the cloud service. Additionally, a mobile application is also developed that connects with the cloud service, providing support to the user on the go. The framework is developed in an academic setting with a team of Software Engineering Students. Various aspects of software development are compared with results and recommendations in four recently completed projects, DSDM²⁸, MADONA²⁹, CMPlayer³⁰ and work done by Manuja et. al.¹⁹.

The remainder of the paper is organized as follows. Section 2 provides recent work in Mobile Cloud Computing and Mobile Cloud Software Development processes. Section 3, provides detailed architecture of the ShaMoCloud framework. Section 4 presents analysis of development aspects, gains and issues encountered in comparison with similar works. Section 5 presents the conclusions.

2. LITERATURE REVIEW

2.1 Cloud Computing

“Cloud computing refers to both the applications delivered as services over the Internet and the hardware and systems software in the datacenters that provide those services”¹¹. With regards to cloud services, the cloud computing types can be divided into two groups:

Service Models: Service model refers to the services type that provided by cloud providers such as Amazon EC2 and Microsoft Azure etc. Software as a Service (SaaS) refers to delivery of software applications or services that run in cloud provider infrastructure. The consumer does not control or manage the resources like servers, operating system and storages. Platform as a Service (PaaS) refers to delivery of deployment services onto cloud provider infrastructure whereas the consumer can deploy certain applications with limitations on control on management of the resources. Infrastructure as a Service (IaaS) refers to delivery of infrastructure provisioning services to the consumer where the consumer can provision servers, operating system, storage and others fundamental resources¹¹-¹².

Deployment Models: Deployment Model refers to how cloud infrastructure owner host and offer the cloud computing services. These are public, private and hybrid cloud models¹⁰. Public Cloud model refers to consumer usage of an organization’s cloud infrastructure on a pay per use basis. Private Cloud model refers to internal deployment,
management and operation of an organization’s own cloud infrastructure. Hybrid Cloud: model refers to an integration on both public and private cloud models within an organization.

2.2 Mobile Cloud Computing

Integrating mobile applications with cloud environments have recently become a popular approach in the industry. The term mobile cloud computing means to run an application such as Google’s Gmail for Mobile on a remote resource rich server (in this case, Google servers), while the mobile device acts like a thin client connecting over to the remote server through 3G11.

Mobile devices are typically challenged in terms of processing power, storage capabilities and communication issues. Although mobile computing provides users with mobility that allows them to accomplish various tasks on the go, mobile cloud computing provides the benefits of both mobile device and cloud computing. In a mobile cloud-computing environment, mobile device’s data storage and data processing are outsourced to the cloud. This increases battery life, computation processing, sharing of information as well as running computation intensive applications on a resource constraint mobile device12. Papers13-14 present examples of various applications developed in Mobile Cloud Computing environment.

Several approaches and techniques have been proposed to augment mobile devices by leveraging cloud computing. However, trust and privacy is still a major issue in MCC that hinder its vision. SAMI24 is an arbitrated multi-tier infrastructure model for mobile cloud computing leveraging Service Oriented Architecture. Authors25 present FocusDrive, a new mobile cloud data processing framework through trust management and private data isolation. Authors26 provide a trust model for offloading computation from mobile devices to a public cloud network. The main precursor for this research is improving mobile device energy efficiency. Zhang et. al.27 provide the concept of personal cloud network. They propose a novel approach of using hybrid cloud to distribute the computing among mobile devices, private cloud and public cloud. User data is stored within the personal cloud framework where users have full, physical control. User can authorize public cloud services to access user-approved data.

3. SHAMOCLOUD SERVICE ARCHITECTURE

This section provides requirement analysis and architecture for ShaMoCloud Service followed by implementation details.

3.1 Requirement analysis of the ShaMoCloud Service

ShaMoCloud Service is a cloud-based framework that provides content management of media files including images and videos. Users can access the service from a web interface or a mobile application. Content accessibility and privacy is of utmost importance. ShaMoCloud is implemented in a hybrid cloud environment for maximum accessibility, scalability and user privacy.

The framework is designed and developed for open accessibility from various types of client devices. A web based interface will provide users access the service through any web browser. Additionally an Android based application is also developed that connects to the service. Users would have to register followed by setting up an account with a two-step authentication process. Users can upload media content including images or videos to their accounts. The media content can be shared with friends of the user. Any user can define friends who can benefit from sharing and managing the uploaded media content. Friends could be invited to register the service and in turn share their media. Sharing permissions could also be revoked by the user. The service allows scalability of resources by allowing users to increase the repository size as well as integrate with other services in the future. We classify ShaMoCloud Services in two sets from requirement analysis perspective.

Video Services: Set of classes developed to cater to all services related to media management. These services provide video uploading, playing, deleting, editing and streaming. Additionally management of media content and user provisioning for all content is provided.

User Services: These services provide the user with account management, accessibility, access control, friend requests, and friend’s management.
3.2 Detailed Architecture:

ShaMoCloud essentially is designed in 3-tier architecture. Figure 1 illustrates an overview of the architecture. The three tiers are named based on their utility. User Services provides interface to the user on web or on a mobile device. The middle tier named Cloud Services implements classes for communication, database for storage, media sharing, content management etc. The third tier provides authentication for data deemed private by the user. Following presents details of these three layers.

A. User Services Layer: At the user level, the cloud service is accessible over the web as well as an application on a smartphone. The devices connect to the second tier through web services. Native code on the Android hardware is written to communicate with the middle tier. Android mobile application was developed to provide the ShaMoCloud services for sharing media and content management. Figure 2 shows the architecture of the android application with alignment of WCF services with the middle tier. Following lists components used for development of user applications.

- New classes were written to implement Restful Web services in JSON format to provide for Android as well as web accessibility. The purpose was to handle JSON request and response of the web service before/after call to the web service. Within this class, http libraries called to handle the various service requests.
- The AsyncTask Class enables proper and easy use of the User Interface (UI) thread in android. It allows performing background operations and publishing results on the UI thread without having to manipulate threads and/or handlers. The AsyncTask class was modified for access by web service calls to manage the UI thread.
- Xml was used for common functionality and to provide interface to the android native code.

B. Cloud Services Layer: The second tier implements the cloud services and web services in the public cloud environment. User’s selected media content is made available in the ShaMoCloud Database. Additionally most classes implementing the video services and user service run as part of the cloud and web services. The framework is a cloud-based solution since it is hosted in a public cloud service provider. ShaMoCloud hosted at Amazon EC2, part of the Amazon Web Service (AWS) utilizes in addition to hardware resources, a web server and SQL Database. Following lists the major components of this tier:

i) Web Interface: This component is developed in asp.net to provide web accessibility to the cloud service. It provides the video content management and sharing functionalities for the user.

ii) WCF Services: This component manages all Windows Communication Foundation (WCF) services. It serves as an intermediate service for smartphone application. It implements:
   a. A SOAP protocol supporting Restful Web services.
   b. Interface for format compatibility with XML and JSON.
   c. Exchange of complex messages including streams of binary data.

In light of the application requirements, two major WCF services were defined in the solution. Each of these services may contain multiple components defined as functions based on the architecture:
- User services: User service contains all functions related to the user as an entity. These include login, sign up, defining and requesting friend messages, and managing friends. Additionally
- Video Services: Video service contains all functions related to the media posted by the user. This includes listing of media (videos / images), uploading media, tagging / annotating media, sharing, deleting and editing media content.

Additionally the public ShaMoCloud DB hosted on SQL Server within Windows Server instance holds all media content tagged to be public. Any content that is tagged private is moved to the repositary in the private cloud.

C. Privacy Services Layer: The third tier composes of a local database for privacy in a trustworthy server. Media content deemed private along with user management settings and authentication details are kept. Communications between second and third tiers are encrypted. We use a centralized approach for user privacy. All media content classified as private is kept in user’s personal and trustworthy machine.

This layer provides accessibility to media content for creating, uploading, deleting all kinds of media. User can tag and annotates media content and classify it as public or private. All media classified as private is moved to the Database
on the private cloud machine. Consequently all publicly tagged media stays on the ShaMoCloud database in the public domain. All information related to authentications and authorization is also managed at this layer. The Private Services layer maintains accessibility information such as start-end time stamps, upload time, location and owner of media contents, accessibility details for devices and media content itself in a SQL database. This service can also be viewed as a secure indexed file storage system.

4. ANALYSIS OF DEVELOPMENT ASPECTS AND COMPARISON WITH OTHER WORKS

This work bridges the gap existing in the software engineering, cloud computing and mobile application development areas. We focus on software engineering aspects of developing mobile cloud applications. To this end, we compare software development practices in four recent research publications.
4.1 Issues for mobile cloud development in Recent works

Kalem et. al.\textsuperscript{28} describe the relation between Agile development methods and Cloud Computing. They first developed a warehouse management system called Dynamic Systems Development Methods (DSDM). Later the same application was developed and deployed as a cloud service in Google App Engine platform. The researchers discussed and contrasted both development approaches and concluded that Cloud based agile development reduces the cost in terms of development time. The development environment was completely changed when migrating from one project to another. It’s not clear what effect would development environment have on the suggested approach. Another consideration could be on the learning curve for development. The same team that built the earlier application developed the cloud based application, having already experienced the pitfalls in providing the solutions. Additionally the web application was tested and deployed on web browsers that could be accessed from smartphones but not specifically for a mobile application executing native code.

Authors\textsuperscript{29} provide a Methodology for Automatic Development of cloud-based business Application (MADONA), accessible to non-IT professionals. The primary goal of MADONA is to allow business stakeholders to develop automatically a cloud-business application simply by describing business requirements via a web form. MADONA determines the business needs and the service ranking is calculated based on the coefficients associated to QoS attributes as assigned by the user based on his priorities. The researchers present the benefits of using agile approach in developing MADONA. The approach used for selecting user requirements is agile but essentially the development approach follows that traditional Waterfall process.

Yu and Boskovic in their work\textsuperscript{30}, discuss integration of cloud computing environment with mobile software engineering. They built a cloud service for streaming media content including videos which would be played on CMPlayer, a mobile cloud video playing application. The Cloud service was built on Google App Engine framework. The authors conclude by providing recommendations and changes to traditional software development processes. The authors faced challenges researching for development solutions. A considerable time management was required for finding resources or developing additional resources for the mobile application.

Manuja and Manisha\textsuperscript{19}, discuss how to move from Agile to Agile on Cloud and present the challenges by developing a real life application. They propose a modified Scrum Framework aligned with Cloud Computing in an ideal way by using Amazon Web Services. They focused on addressing the Cost-Time-Quality (CTQ) challenges by implementing Agile on Cloud model. The results compare traditional scrum against the proposed Agile on Cloud model, which shows indications of improvements with regards to CTQ. Agile on Cloud model shows positive effects on cost and time dimension whereas there is no effect on quality dimension due security and privacy issues. Furthermore, the model suggested was not tested for building mobile applications. It was assumed that all devices would access a web interface. No consideration was given to limitations of User Interface when developing for mobile devices.

4.2 Gains in developing ShaMoCloud Service

The team that developed ShaMocloud had prior experience of developing using Team Software Process(TSP)\textsuperscript{17}. The TSP, based on waterfall model, is a quality focused process where defects are identified and prevented at an early stage of development. The stark contrast in software engineering practices when migrating from TSP development approach to extreme programming was clear. It can be argued that TSP is not suitable for a project with fast changing requirements such as development projects in Mobile Cloud computing domain. Following lists the pros and cons of developing ShaMoCloud using extreme programming approach

- Public Cloud domain was used for common storage area. Scalability and elasticity issues were easily managed. A configuration management system supported by the common cloud based storage space was used by distributed teams to share the code and project updates. Additionally web server and MS SQL database were utilized that were part of the AWS EC2 instances.
- Version control for various code was easily managed. Cloud computing facilitates solutions for managing teams such as Trello\textsuperscript{13} and for source code versions control such as Bitbucket\textsuperscript{18} and Github\textsuperscript{22}. Project setup was much speedier compared to TSP implementation. Shared development environment facilitated sharing and updating code for various classes and libraries online.
- On the public cloud environment, there was no need to manage the network infrastructure. The private cloud was
implemented using a server at the client’s premises. There were no problems in establishing hybrid cloud with the two services.

- There were no issues in testing and integration. All reports were shared by team members in the shared space. The customer had access to all reports and provided input from time to time on various issues.
- Trust and privacy concerns are a big issue in the public cloud which were addressed by ShaMoCloud. The proposed 3-tier architecture allows users to manage privacy of their media content. Images and videos could be classified as private and moved to the private cloud for user satisfaction. Additionally user can also restrict access to the private media content.
- Data Ownership in the public cloud was an issue that was addressed by relying on identity and access management by the user. Users may willingly choose to keep public data on Amazon servers whereas sensitive data may be offloaded to the private cloud.
- The overall cost of development in terms of time and quality was much less compared to team’s earlier data using TSP. The team also benefited from shared environment to reduce the number of defects thus improving the software quality.

The ShaMoCloud project was completed in 4 releases. Table 1 shows the development effort in terms of days in all phases of ShaMoCloud development. The team determined that the DSDM Project 28 is closest in terms of requirements and size compared to ShaMoCloud. It can be seen that most phases of project development were comparable to the DSDM project.

### 4.3 Issues with ShaMoCloud Service

Quick paced development using extreme programming compared to TSP development. As anticipated, agile processes accelerate development. Following lists issues with ShaMoCloud Development project.

- Frequently updating mobile and cloud libraries: At the start of ShaMoCloud project, the Android API level was 19 but by the time the project was completed Android API level 21 was released. The latest update required changes to be made to the application that was built. These frequent changes provide new functionalities whereas deprecating libraries and calls consequently requiring frequent updates to the applications.
- Research Effort in developing: The team spent considerable time researching for open solutions. There are very few libraries available for development in mobile cloud environment. Most libraries and available as open source which typically require modification and integration with existing code.
- Data Privacy: Although ShaMoCloud addressed the issue of data privacy using private cloud for secure and data categorized by the user as private, the scalability issue on the private cloud layer would require investment. In order to solve this issue, the team is considering the concept of personal clouds presented in work 27 where users would access their personal data from the public cloud using a virtualized device working as an overlay on their personal hardware.
- Project Environment Safety: During development, the team used tools in public domain which may expose code, built data, configuration files etc. leading to huge risk in terms of security and project confidentiality.

Despite the above mentioned issues, the team is confident that the overall cost in terms of software size, time and quality was reduce by using cloud based extreme software development while working on the ShaMoCloud project.

### 5. CONCLUSIONS

This paper presented ShaMoCloud Service, a mobile - cloud based environment for content / media management. Design and implementation of a hybrid cloud environment was detailed with focus on integrating the three tiers of the framework while preserving data security and privacy. Various aspects of development using extreme programming as an agile development methodology were discussed. The work was compared to similar projects highlighting the gains and challenges in development. Our experience shows that Mobile Cloud application development can be integrated with agile development methodologies reducing the cost, time and improving software quality. Furthermore, accelerated pace of development using agile methods is suitable for fast changing pace of newer mobile and cloud technologies. Results show the positive indications to applying Extreme Programming (XP) for both mobile and web application integrated with hybrid cloud.
Table 1. Process Time Comparison in Development Phases between ShaMoCloud and DSDM28

| Phases                        | DSDM28 Time | ShaMoCloud Phases      | Time |
|-------------------------------|-------------|------------------------|------|
| Requirement Gathering         | 4 days      | Requirement and Planning| 2 days|
| Testing Design and development| 32 days     | Design, Coding and Development | 35 days|
| System Testing and deployment | 12 days     | Integration Testing and deployment | 7 days|
| Research Effort               |             |                        | 19 days|

Acknowledgements

This work is partially funded by Resarch grant number GP-CCIS-2013-11-18 at Prince Sultan University, Riyadh, Saudi Arabia. The authors would also like to thank all project participants for their valuable contributions.

References

1. Armbrust M, Fox A, Griffith R, et. al. Above the Clouds: A Berkeley View of Cloud Computing Michael. Technical Report UCB/EECS-2009-28, Berkeley, 2009.
2. Google Docs. Google, [Online]. Available: http://www.google.com/intl/en/docs/about/. [Accessed 12 January 2015].
3. Google Gmail. Google, [Online]. Available: https://www.gmail.com/intl/en_us/mail/help/about.html. [Accessed 12 January 2015].
4. Products Office. Microsoft, [Online]. Available: https://products.office.com/EN/. [Accessed 12 January 2015].
5. Microsoft Azure. Microsoft, [Online]. Available: http://azure.microsoft.com/en-us/. [Accessed 12 January 2015].
6. App Engine. Google Cloud Platform, [Online]. Available: https://cloud.google.com/appengine/. [Accessed 12 January 2015].
7. Amazon EC2. Amazon, [Online]. Available: http://aws.amazon.com/ec2/. [Accessed 12 January 2015].
8. CloudOne. CloudOne, [Online]. Available: http://oncloudone.com/. [Accessed 12 January 2015].
9. Sardis F, Mapp G, Loo J, Aiash M and Vinel A. On the Investigation of Cloud-Based Mobile Media Environments With Service-Populating and QoS-Aware Mechanisms. In IEEE Transactions on Multimedia 2013;15(4):769-77.
10. Mell P and Grance T. The NIST Definition of Cloud Computing. NIST Special Publication 800-145, 2011.
11. Fernando N, Loke S W and Rahayu W. Mobile cloud computing: A survey. Future Generation Computer Systems 2013;29(1): pp. 84-106.
12. Abolfazli S, Sanaei Z, Ahmed E, Gani A and Buyya R. Cloud-Based Augmentation for Mobile Devices: Motivation, Taxonomies, and Open Challenges. In IEEE Communications Surveys & Tutorials 2014;16(1): pp. 337-68.
13. Moustafa H, Maréchal N and Zeadally S. Mobile Multimedia Applications: Delivery Technologies. IT Professional 2012;14(5): pp. 12-21.
14. Wan J, Liu Z, Zhou K and Lu R. Mobile Cloud Computing: Application scenarios and service models. In 9th Intl Wireless Comm. and Mobile Comp. Conference (WCMC), Sardinia, 2013.
15. Trello. [Online]. Available: https://trello.com/. [Accessed 18 March 2015].
16. The AsyncTask Library, available online, http://developer.android.com/reference/android/os/AsyncTask.html [Accessed 12 January 2015].
17. Watts S. H. The Team Software Process (TSP). CMU/SEI-2000-TR-023, Pittsburgh, 2000.
18. Bitbucket. [Online]. Available: https://bitbucket.org/. [Accessed 18 March 2015].
19. Manuja M and M. Moving Agile based projects on Cloud. In IEEE International Advance Computing Conference (IACC), Gurgaon, 2014.
20. Carvalho F and Azevedo LG. Service Agile Development Using XP. In IEEE Seventh International Symposium on Service-Oriented System Engineering, Redwood City, 2013.
21. Dev D, Baisnab, KL. A Review and Research Towards Mobile Cloud Computing. In 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), pp.252-256, April 2014.
22. Github. [Online]. Available https://github.com/. [Accessed 18 March 2015].
23. Doelitzscher F. et.al. Private cloud for collaboration and e-Learning services: from IaaS to SaaS. Springer Computing 2011;91, pp. 23-42.
24. Sanaei Z, Abolfazli S, Gani A, Shiraz M. SAMI: Service-based arbitrated multi-tier infrastructure for Mobile Cloud Computing. In IEEE Intl. conf. Communications in China (ICCC), pp.14,19, 15-17 Aug. 2012
25. Dijiang H, Zhbin Z, Le Xu, Tianyi X, Yunji Z. Secure data processing framework for mobile cloud computing. INFOCOM WSHPS. pp.614-618, 10-15 April 2011.
26. Bhattacharjee S, Majumder S, De D. Trust model for femto-cloud based mobile network. In 5th International Conference on The Next Generation Information Technology Summit (Confluence), pp.53,58, 25-26 Sept. 2014.
27. Zhang JY, Pung Wu, Jiang Zhu, Hao Hu, Bonomi F. Privacy-Preserved Mobile Sensing through Hybrid Cloud Trust Framework. In Sixth IEEE International Conference on Cloud Computing (CLOUD), pp.952-953, June -July 2013.
28. Kalem S, Donko D and Boskovic D. Agile methods for cloud computing. In 36th International Convention in Information Communication Technology Electronics Microelectronics (MIPRO), Opatija, 2013.
29. Benfenatki H, Ferreira Da Silva C, Benharkat AN, Ghodous P. Cloud-Based Business Applications Development Methodology. In 23rd IEEE WETICE Conf. pp.275,280, June 2014.
30. Yu WD and Yuan H. An Approach to Explore Mobile Software Engineering Advances in Cloud Computing Environment. In 35th Annual IEEE Conference on Computer Software and Applications Workshops (COMPSACW), 2011.