QoS-Based Web Service Selection Mechanism for Ad-hoc Mobile Cloud Computing

By

Ayotuyi Tosin, **AKINOLA**

(201426777)

A dissertation submitted in fulfillment of the requirements for the degree of

Master of Science in Computer Science

Faculty of Science and Agriculture

Department of Computer Science

University of Zululand

KwaDlangezwa

RSA

Supervisor: Prof M.O Adigun

Co- Supervisor: Prof S.S Xulu

2017

DECLARATION

I, Ayotuyi T, Akinola declare that, the work contained in this dissertation has not been previously submitted in whole or in part, to meet requirements for any other degree or professional qualification at this or any other higher education institution. To the best of my knowledge and belief, the dissertation contains no material previously published or written by another person except where due reference is made.

Ayotuyi Tosin, **Akinola**

Signature: ___________________________

Date: ________________________________

DEDICATION

I dedicate this work to the Lord almighty, who gave me strength and great patience throughout the process of this study. He has been my eternal rock and a constant source of refuge. Also, to my mum, Mrs Comfort Olushola AKINOLA for her constant supports with prayers.

ACKNOWLEDGEMENTS

I would like to sincerely thank Prof. M.O Adigun for his unwavering support, advice, guidance and for providing me with the opportunity to make this work a reality. The success of this dissertation is attributed to his extensive support and refinement. I also appreciated my Co-Supervisor Prof S.S Xulu for his timely advice and encouragement throughout this study.

I am indeed indebted to my parents and my siblings for their support both morally and financially, coupled with their constant prayers without which this research work would not have been realistic. I am also grateful to Mr Olukayode Oki for his assistance in processing my SAQA (South African Qualifications Authority) certificate and ensuring my landing here was homely. Moreover, I am also indebted to Dr Akingbesote for his fatherly role during the course of my study as well as the assistance from Dr Ayoola. Every time I remembered both of you, I pray to God to make all your wishes come true. To my colleagues in the department: S C Fakude, D. Egbe, N. Ijeoma, and C. Kudzai. I appreciate you all for your support and making my research interesting during the stressing times. You are all too much!!! I also appreciate the assistance of Mr Pragasen Mudali during the proof reading of this dissertation. Thanks also go to Miss Thabile, Mrs Deli and Mr Paul Tarwireyi for their assistance any time I need one.

I am also grateful to my amiable sister Ibukunoluwa O. Deji-Alamu for her loving prayers and kindly devotion toward the success of this degree. Through thin and thick, the encouragement is never wanted. I appreciate you a million. I will also not forget my loving friend Oguntoye Jonathan for his support and prayers. Also, I appreciate all members of the Deeper Life Campus Fellowship (DLCF UNIZULU Chapter) for their support and generous prayers. May the Lord be with you all. Finally, To Telkom my sponsor, I cannot thank you enough for your support.

ABSTRACT

The Ad-hoc Mobile Cloud Computing (AMC) paradigm came into existence to settle one of the major challenges of Mobile Cloud computing especially low internet connectivity. The AMC is formed when mobile devices in the same proximity connect through a wireless connection or any other means; enabling the devices to request web services from one another within the Mobile Cloud. However, the user dissatisfaction experienced in the course of requesting for a Web Service is a challenge that needs to be addressed. This is because existing service selection approaches emphasized functional rather than non-functional qualities of matched services. Moreover, effective selection approaches must avoid quality computation during service selection that produces similar or redundant results at runtime.

In an attempt to address this service selection challenge, a service selection mechanism for Ad-hoc Mobile Cloud was developed in this work. This mechanism was synthesized from existing service selection approaches. The mechanism was evaluated using the experimental research method. The evaluation of the mechanism accesses the suitability of the selected web services for all requesting users. The quantitative evaluation aspect entails the use of execution time, throughput and service availability to analyse the performance of the selection approach. The approach employs the use of selected Quality of Service (QoS) properties and user feedback to determine the most appropriate service to be selected for any request. Experiments affirm a continuous updated and unlimited range of users' Web Service assessment (Feedback) as part of QoS properties enhances more optimal service selection within AMC computing paradigm. The is because non- feedback enabled attains one optimal selection out of seven thus guaranteeing 0.13 probability of optimal selection against a probability of 1 depicted in feedback-based selection mechanism.

TABLE OF CONTENTS

LIST OF ACRONYMS

ABBREVIATIONS	MEANING
SOC	Service Oriented Computing
SOA	Service Oriented Architecture
MCC	Mobile Cloud Computing
SMMEs	Small, Medium, and Micro Enterprises
GUISET	Grid-based Utility Infrastructure for SMME Enabling Technologies
QoS	Quality of Service
HTTP	Hypertext Transfer Protocol
RPC	Remote Procedure Call
XML	Extensible Markup Language
SOAP	Simple Object Access Protocol
WSDL	Web Service Description Language
UDDI	Universal Description, Discovery, and Integration
OWL-S	OWL Web Ontology Language for Services
IR	Information Retrieval
SAWSDL	Semantic Annotation for Web Service Description Language
SOP	Service Oriented Peers
LBS	Location Based Services
SAFE	Service Advisors For E-business
PC	Personal Computer
AMC	Ad-hoc Mobile Cloud
MCC	Mobile Cloud Computing
CC	Cloud Computing
GUI	Graphic User Interface
SME	Selection Middleware Engine

LIST OF FIGURES

LIST OF TABLES

LIST OF PUBLICATIONS

Parts of the work presented in this dissertation have been published in the following conferences and journal paper(s):

1. Akinola, A.T., Adigun, M.O., Mudali, P. A Federated Web Service Selection Approach for Ad-hoc Mobile Cloud Computing *in 19th Southern Africa Telecommunication Networks and Applications Conference* (SATNAC). Sept. 4th-7th, 2016, pp 368-373, SA.
2. Akinola, A.T., Adigun, M.O., Akingbesote, A.O., Mba I.N., 2015. Optimal Service Selection in Ad-hoc Mobile Market Based on Multi-Dynamic Decision Algorithms. In *proceedings of 2nd World Symposium on Computer Networks and Information Security IEEE, El Mouradi Hammamet 5, Tunisia*. Pp 1-6. DOI: 10.1109/WSCNIS.2015.7368303
3. Akinola, A.T., Adigun, M.O., Akingbesote, A.O., 2015. QoS-Aware Single Service Selection Mechanism for Ad-hoc Mobile Cloud Computing. In *proceedings of International Conference on Computing, Communication and Security IEEE, Mauritius* pp 1-6. DOI: 10.1109/CCCS.2015.7374164
4. Akinola, A.T., Adigun, M.O., Akingbesote, A.O., Mba I.N., 2015. Optimal Route Service Selection in Ad-hoc Mobile E-marketplaces with Dynamic Programming Algorithm using TSP Approach. In *proceedings of International Conference on E-Learning Engineering and Computer Softwares, kualar Lampur, Malaysia*. pp. 74–81
5. Akinola, A.T., Adigun, M.O., Akingbesote, A.O., 2015. Optimal QoS Service Selection in Ad-hoc Mobile Market Using Analytical Hierarchy Model. In *proceedings of World Congress on Sustainable Technologies London, United Kingdom, IEEE* pp 84-88 DOI: 10.1109/WCST.2015.7415124.

6. Akinola, A.T. and Adigun, M.O, 2016. Feedback-Based Service Selection in Ad-Hoc Mobile Cloud Computing in 3rd International Conference on Advances in Computing and Communication Engineering (ICACCE-16). Durban, Nov. 28-29, 2016. Pp 72-77 , SA.

CHAPTER 1

INTRODUCTION

1.1 Overview

The explosive growth of the mobile applications and emergence of cloud computing have drawn more attention to the scientific research world (Voas & Zhang, 2009). The bulkiness of mainframe computers has been drastically substituted with the privacy and mobility enhanced by the personal computers (Jarir, Boumhamdi, & Quafafou, 2009). These personal computers have made life highly convenient for its users through the help of the Internet which enhances connections among these computers. The Internet is a global system of interconnected computer networks that uses the standard Internet Protocol suite to link billion of devices worldwide. To achieve flexibility and agility between clients-providers interaction, computing processes are adopting Service Oriented Computing (SOC) which views business functions as modular software services which run on the Service Oriented Architecture (SOA) (Sathya, Swarnamugi, Dhavachelvan, & Sureshkumar, 2012).

Cloud computing is an Information Technology Service Model where computing services are delivered on-demand to customers over a network in a self-service fashion, independent of device and location (Marston, Li, Bandyopadhyay, Zhang, & Ghalsasi, 2011). The resources required to provide the requisite Quality–of-Service levels are shared, dynamically scalable, rapidly provisioned, virtualised and released with minimal service provider interaction. Users pay for the service as an operating expense without incurring any significant initial capital expenditure, with the cloud services employing a metering system that divides the computing

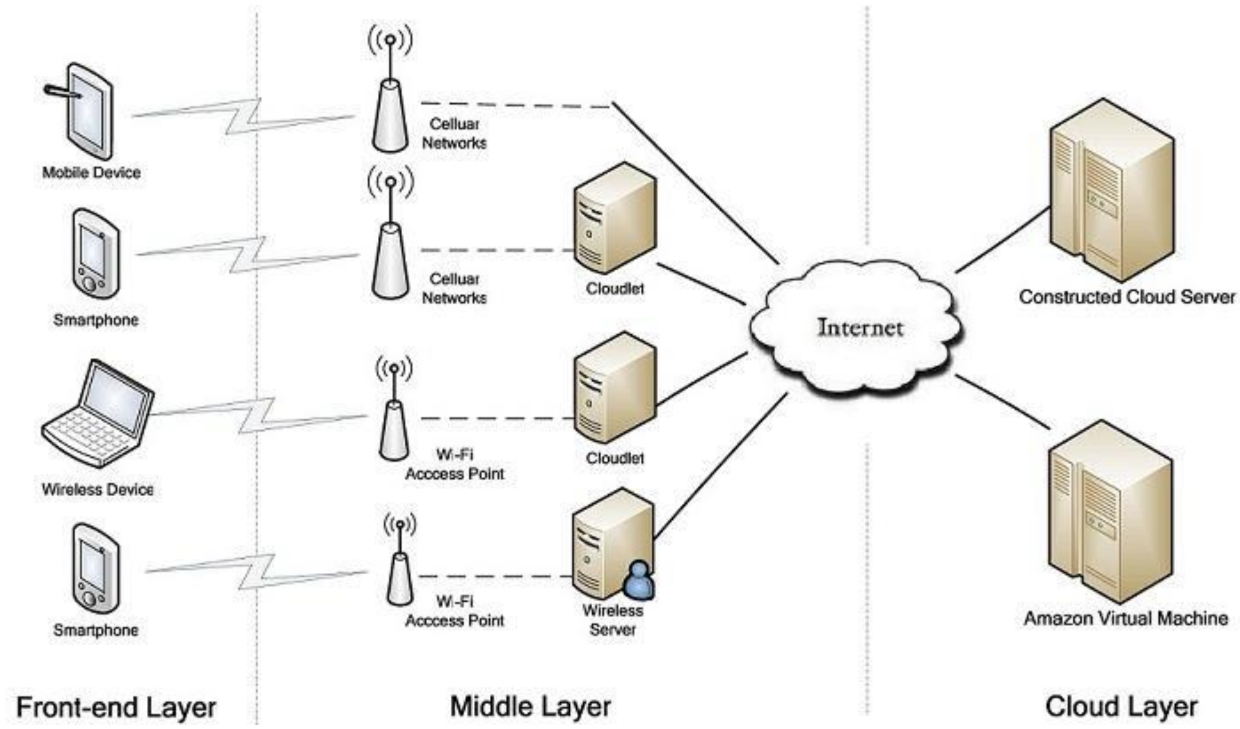

Figure 1.1: Mobile Cloud Computing architecture **(Kaushik & Kumar, 2014).**

resource into appropriate blocks. This cloud computing, in reality, is an on-demand grid computing. A closely related computing platform that combines mobile devices with cloud computing architecture is the *Mobile Cloud Computing* (MCC) (Marinelli, 2009). This is shown in Figure 1.1.

1.2 Preliminary Background

In the Mobile Cloud, there are three main participants, the cloud providers (Cloud Layer), the network providers (Middle Layer) and the consumers (Front-end Layer) as shown in Figure 1.1. It should, however, be noted that the middle layers in this paradigm often introduce a number of challenges. One of the challenges is inaccessibility to the central cloud due to weak or absence of Internet connection (Dinh, Lee, Niyato, & Wang, 2013). Also, the wireless connection at the middle layer often causes high consumption of a battery in the front-end layer (Chen, Won, Stoleru, & Xie, 2014). Additionally, request latency occurs as a result of many hop-count connection distances that may occur within the middle layer as seen in Figure 1.1. These and other challenges lead to the evolution of an infrastructure-less mobile platform called *Ad-hoc Mobile Cloud computing* (AMC). AMC represents a group of mobile devices that serve as a cloud computing provider by exposing their computing resources to other mobile devices (Dev & Baishnab, 2014). The Ad-hoc mobile system allows devices to participate at will by joining, leaving or being available for service provisioning (Huerta-Canepa & Lee, 2010). It is envisaged to be a temporary backup for the main cloud server when the mobile devices could no longer access the cloud server. AMC computing gains considerable attention in situations where there is usual occurrence of spontaneous disconnection to the cloud server. For example, a student in need of an application to resolve facial analysis (e.g facial recognition) would be able to access it from other mobile devices because it is a common tool being used by all the students in the school. This is because one or two devices would have likely made use of the application since there was connectivity earlier.

Therefore, implementing the AMC system described above into a service provisioning platform will require that requests are optimally matched with available Web Services provided by service providers. Generally, users' interest is a key factor that influence users' satisfaction in any service oriented computing. Thus, *Quality of Service* (QoS) is an important tool that the providers use to specify what they are offering and also help the users to specify what they really want.

These Web Services are software components which are built up to enhance the implementation of the SOA paradigm (Badidi, 2011). The SOA is an architectural paradigm that has become the preferred choice for designing and developing systems which utilise Web Services to perform specified function or task (Sathya et al., 2012). For example, one of the aims of the Small, Medium, and Micro Enterprises (SMMEs) is to be able to access the low-cost ICT infrastructures and services through utilising the commonly available mobile devices to consume such Web Services provided from the cloud. Hence, the Centre of Excellence of the University of Zululand is currently undertaking a SOA related project called Grid-based Utility Infrastructure for SMME Enabling Technologies (GUISET). Thus, all challenges related to that SOA implementation are also that of GUISET challenges as well. Invariably, GUISET as a middleware platform is looking forward to providing an enabling technology which would not only reduce the challenges involved in providing Web Services to mobile customers (subscribers), but also addressing the deployment of modern technical solution to avoid total idleness of mobile users during low connectivity. Therefore, GUISET adopts a kind of utility-based business model wherein service providers consider making infrastructure management, infrastructure-less management and resource services available on-demand at little or no cost, mostly to –SMMEs (Buthelezi, Adigun, Ekabua, & Iyilade, 2008). The research work in this dissertation comes under the infrastructure-less management of mobile devices that make use of Web Services on the GUISET middleware platform.

The advancement in mobile software and hardware technology development has drawn the attention of the research world to the use of mobile devices in order to improve and complement cloud computing effort. This has enabled mobile devices' ability to perform a higher task than they could perform previously. Therefore, GUISET looks towards utilising the power of the technological advancement embedded in the mobile devices to fulfil its goals in terms of cost minimisation and easy access to Web Services through peer mobile devices. For example, the m-Learning application is one of the GUISET use cases that is hoped to be implemented in a MC, where teachers and students are expected to form an AMC, accessing, discovering and selecting educational materials.

From the foregoing discussion, it is evident that both the MCC and its evolving AMC paradigm aim at making available relevant Web Services to mobile clients. Literature shows that many competing Web Services may exist with similar functionality but dissimilar Non-functional qualities (Iordache & Moldoveanu, 2014; Wagh & Thool, 2014). Usually, Web Service provisioning process includes discovery and selection process together. The discovery process involves searching for an appropriate Web Service based on its functional requirement. On the selection process, the best fitting service is selected from the services discovered earlier by taking into consideration the client's preferred non-functional requirements which describe the desired Web Service. Hence, it is very important to understand the qualities that explain the properties of Web Service in relation to Web Service selection in the context of AMC.

There exist many solutions in the literature that researched into selecting appropriate Web Service, for example in grid, cloud computing, and MCC (S. Wang, Zibin, Qibo, Hua, & Fangchun, 2011). However, the issue of how selection of Web Service in the context of AMC with similar functionalities has not been fully addressed. In addition, during selection process, the situation where the computation or aggregate scores have the same score (Ties) during run time is also a challenge. This research addresses these challenges by opening up the use of extensive QoS such as response time, reliability, service cost and more importantly user feedback to get the optimal services among various competitive ones at each instance of client's request.

1.3 Statement of the Problem

It is quite obvious that personal mobile devices have gained immense popularity in the recent years. Apart from not spending high cost to purchase stationary infrastructural resources, the great advantage of using it anywhere and anytime cannot be overemphasised. Such advantages include improved alternative for data and resource storage, advanced processing power, reduced overall level of power consumption, and ubiquitous computing (Kirby, Dearle, & Macdonald, 2011). Thus, GUISET, as explained in Section 1.2 is a project that involves the use of these mobile devices.

GUISET as a middleware platform is looking forward to providing an enabling technology which would not only help reduce the cost of providing services to mobile customers but also research into increased mobile device performance through solving the challenge of selecting the best fitting Web Service to mobile clients especially during Web Service selection ties. Some systems select arbitrary Web Service during such ties while others iteratively compare some selected qualities for relevant service and consuming a lot of time in the process. Therefore, this study aims to provide the support of spontaneous interaction networking with selection of optimal Web Service. The research's approach is to investigate the relevant mechanism for selecting Web Services based on the QoS or Non-functional properties associated with them. Moreover, from the existing literature, there is non-availability of a feedback-based mechanism in existing systems for rating services as a criteria for service selection in the context of AMC. Therefore, it is envisaged in this research that the use of extensive QoS parameter(s) (i.e continuous feedback updates) in the formulated service selection model will enhance more optimal service selection in an AMC.

1.4 Rationale of the Study

AMC has come up as one of the solutions to low computing storage, limited bandwidth and intermittent connectivity of mobile devices with IT infrastructures (Jung & Wang, 2010; Huerta-

Canepa & Lee, 2010). However, the possibility of mobile devices to host Web Service has opened up the opportunity to share Web Services among themselves within the AMC environment (Srirama & Paniagua, 2013; Wagh & Thool, 2013; Wagh & Thool, 2014). The sharing of Web Services within peer devices has opened up the challenge of meeting various user requests within the AMC system. Among the major challenges is the competition within Web Services. This is because of Web Services that have different Non-functional properties with the same functionality. In addition, Web Services sometimes attain the same QoS computation at run time during service provisioning. To resolve this, most research approach this by selecting any one at random among those that attain the similar QoS computation. Whereas, the use of one or more other generic parameters could have ensured that an optimal solution is achieved among the selection ties. Literature shows that the QoS properties are viable tools in expressing and requesting of interest-driven Web Services (Sathya et al., 2012). In addition, selection of the best or optimal service in AMC is a function of the selection mechanism put in place. Thus, the need for mobile users to have access to an optimal Web Service during service selection motivated this research. This study envisaged that with the proper service selection mechanism in place, mobile devices within the AMC may take full advantage of other specific mobile peers that could provide them with the most appropriate service available within the AMC.

1.5 Research Questions

How can QoS-based Web Service selection mechanism be implemented in an infrastructure-less domain such as Ad-hoc Mobile Cloud environment?

Sub-Questions:

1. Why are the current QoS-based Web Service selection mechanisms not adequate for the AMC scenario?
2. How can QoS information be used to enhance optimal Web Service selection process in Ad-hoc mobile environments?
3. How can QoS-based Web Service selection be evaluated in typical use case scenario?

1.6 Research Goal and Objectives

1.6.1 Research Goal

The goal of this research work is to develop a QoS-based Web Service selection mechanism for AMC computing.

1.6.2 Research Objectives

In achieving the goal of this research work, the following objectives were accomplished:

1) To investigate existing approaches to QoS-based Web Service selection.
2) To implement a service selection mechanism aided by QoS-based parameters on a use-case scenario from existing practices.
3) To evaluate the performance of the proposed mechanism in terms of execution time, throughput, service availability and feedback effect.

1.7 Research Methodology

This research work took a quantitative approach. The research objectives were accomplished by using the following methods, namely Literature survey, Mechanism formulation and Proof of concept. The research evaluation was based on well-constructed experiments using Java toolkits and plugins. The following were carried out to achieve the desired goal of this research:

1.7.1 Literature Survey

The aim of this research method was to investigate existing research regarding Web Service selection for various paradigms like grid, cloud and MCC for service selection. The survey would have been incomplete without reviewing the modes of selection in Ad-hoc networks and likewise sensor networks. The survey focused on how existing scholars formulated models and architectures for service selection, and how algorithms were implemented for optimal selection of service providers. Assessment of these research areas was carried out in relation to the need of AMC for service selection. The survey also identified the vital parameters of importance in any Ad-hoc environment for the optimal selection of services.

It is also crucial to understand the scope, purpose, and limitation of the currently existing selection mechanisms which often encounter service ties (service providers with similar QoS aggregates). These ties introduced additional latency into the service response time thereby affecting the total time taken in service execution. This and other important issues were fully discussed in chapter 3 and the output helped to inform the study formulation of an Ad-hoc mobile selection mechanism that enables the optimal selection of Web Service. Research objective 1 was accomplished by using the study from the Literature survey conducted. The existing approach were seen in the literature alongside with the gap to be filled.

1.7.2 Formulation of the web service selection mechanism

The shortcomings of existing approaches to service selection were addressed by formulating a user feedback aided selection mechanism based on the QoS of Web Services. Using the knowledge that was gained from the literature, the design metrics and the selection mechanism were developed. Using a scenario, the proposed solution is explored and demonstrated through a proof of concept. The research objective 2 was achieved by using mechanism formulation and proof of concept that were carried out here.

1.7.3 Proof of concept

The validation of the result to ensure its usefulness and practical utilisation are important aspect of the design of this study. This study developed a prototype of the service selection mechanism and evaluates it as a proof of concept to show the applicability and usability of the proposed mechanism. Experiments were then conducted to evaluate the performance of the prototype using the response time, service availability, throughput of service requests as the metrics. The effect of user feedback during selection ties is also examined. Research evaluation for the objective 3 was based on well-constructed experiments using Java toolkits and plugs which was part of the build-up of the proof of concept.

1.8 Dissertation Structure

The rest of this dissertation is structured as follows:

Chapter 2 discusses the need for service selection in service provisioning paradigms. This discussion elaborates the use of Quality of Services as a tool for service selection in different service oriented systems and, in conclusion, discusses the various approaches to service selection in the AMC system.

Chapter 3 presents literature survey in service selection approaches, using the functional, Non-functional and the user feedback as the selection approaches. Existing approaches were analysed and compared to identify their strengths and shortcomings as well as the reuse possibilities of combining approaches for the solution.

Chapter 4 presents the basic structural definition of service selection mechanism and its elements as well as the design requirements which forms the basis of the mechanism. It also explains the functions of each of the components used in developing the Ad-hoc selection mechanism.

Chapter 5 presents the major aspect of this research through defining the implementation of the selection mechanism. It discusses the experimental set up for the proposed selection mechanism and how each of the components was derived.

Chapter 6 presents the performance evaluation aspect of this dissertation. The chapter discusses some experiments performed which confirmed the suitability of the proposed mechanism to solving the highlighted service selection challenges.

Chapter 7 concludes the dissertation with an emphasis on its contributions in the domain of Web Service selection in AMC computing environment. The researcher also discussed how the research questions were addressed in each case. The limitations of this thesis are reviewed here to drive future research directions that may be tackled by interested readers.

CHAPTER 2

BACKGROUND

2.1 Introduction

This chapter discusses the various technologies underlying the AMC computing paradigm and how they are interconnected to the paradigm. These include the Distributed Computing, Service Oriented Architecture, Grid Computing, Cloud Computing and Web Services. Distributed Systems explain a software system through which components located on interconnected computers pass messages from one to another via a communication medium such as Hypertext Transfer Protocol (HTTP) and Remote Procedure Call (RPC). The component devices within the system interact with one another to achieve a common goal. These components are characterised by independent failure, lack of global clock for synchronisation and concurrency of the components (Coulouris, Dollimore, & Kingberg, 2001; Papazoglou, 2008).

The SOA ensures the combination of modular software components (Web Services) to be orchestrated together to perform a specific task over the networked computer devices as depicted in Grid and Cloud Computing without human interaction and the need to change the underlying running program. Grid Computing is one of the distributed paradigms that allow consumers to make use of grid resources or services without necessarily owning any of the used resources or services. The need for on-demand service request served as the inspiration for the Cloud Computing paradigm.

Cloud Computing offers services over the Internet that lower IT capital expenditure and reduce business operating costs. These services offer on-demand capacity with self-service provisioning on a pay-per-use basis for greater flexibility and agility (Sabahi, 2011; Shaikh & Haider, 2011). Cloud resources can be quickly deployed and effortlessly scaled with all procedures and applications. As a result, cloud computing helps organisations enhance service delivery, streamline IT management and adjust IT services to implement business necessities. Cloud computing can also simultaneously support core business functions and provide capacity for new and innovative services (IBM, 2010). The AMC is typically composed of interconnected mobile devices which act in a similar way as connected computer systems to share resources and services amongst themselves. These processes of sharing resources and Web Services have necessitated the need for selection of appropriate Web Services within the AMC that optimally meet the user requirements.

Section 2.2 discusses the importance of Web Services and the usefulness of QoS in service description. Section 2.3 describes the categories of qualities of services while section 2.4 explains the QoS control mechanisms that is typically used in service selection. The need for Web Service selection in AMC was elaborated in section 2.5 and finally section 2.6 discusses the approaches in AMC service selection. Section 2.7 summarises the chapter.

2.2 Web Services (WSs)

Web Services are the building blocks used to expose the functionalities provided by the service providers for heterogeneous consumers in a decentralised and distributed systems (Marwaha, Banati, & Bedi, 2013). WSs allow for the creation of distributed applications in that they can be published to and accessed over the Internet and corporate Intranets. This is ensured by using open standards

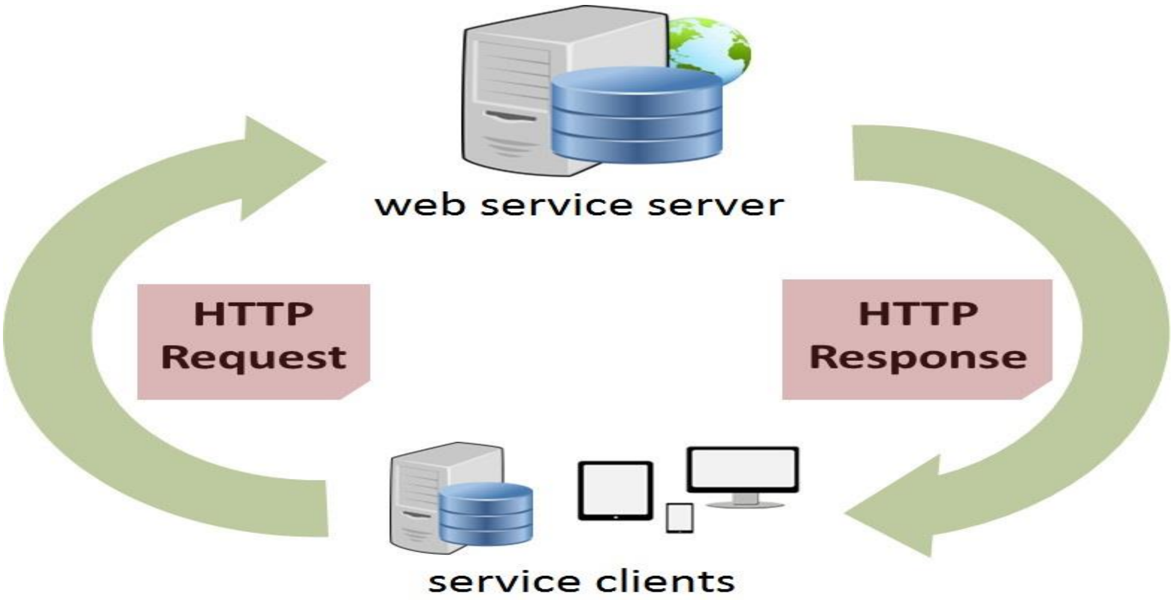

Figure 2.1 Service clients consuming Web Service via HTTP protocol **(Hamad et al., 2010).**

such as (HTTP), Extensible Markup Language (XML) and Simple Object Access Protocol (SOAP). These standards are used for sending requests to and retrieving responses from the compatible mobile devices. The Web Service Description Language (WSDL) (D'Ambrogio, 2006) contains all the information about service capabilities and the invocation mechanisms. The service clients issue a HTTP request to consume a service from a server and in reverse receive a HTTP response as shown in Figure 2.1.

The Universal Description, Discovery, and Integration (UDDI) is a service registry that allows for the dynamic discovery of Web Services (Marwaha et al., 2013). One major goal of the various protocols discussed above was to enhance the development of Web Services through varying open standards to enable the interoperability of varying applications developed from different languages.

Different applications may have been built on a variety of different operating systems (e.g. Windows and Linux) with different programming languages, using different middleware and different data stores. The main goal of Web Services are to enable these various components to interact by providing a standard set of technologies for service discovery (UDDI) and service invocation (SOAP). The design of Web Services is to support interoperable machine to machine interaction over a network. The Web Service supports a more general trend towards SOC, thereby promising to transform enterprise software systems to an orchestral of loosely coupled reusable service components. The reusable qualities provide a better connectivity among various business partners, Web Services support not only accessibility by laptops and desktop computers but also via multitouch tablet computers and small mobile devices to perform all forms of business operations. This approach would inculcate various round trips between mobile user's wireless devices and Service Hosting Stations e.g. grids and clouds. Such services are called M-services such are M-Learning services, M-health services and M-commerce services (Yang & Bouguettaya, 2006).

These frequently accessed mobile services normally portray the following characteristics: (*i*) they are specific to a particular location such as a city or a community, (*ii*) they provide real-time services such as weather report, exchange reports, (*iii*) they do not need user interaction and, (*iv*) they are always accessed by many clients. Thus, Web Services are self-contained and self-describing computational Web components designed to support machine-to-machine interaction by remote invocations (Zhu, Kang, Zheng, & Lyu, 2012). The various functionality of Web Service can be combined to offer composite services to human users or software applications. As an emerging technology, which enhances the application to application remote interactions, Web Services standardisation has continued to play important roles in the integration of heterogeneous software towards achieving a general-purpose service infrastructure that could satisfy the enormous need of the larger society.

Currently, Web Service Technology is being used by various enterprises and industry to implement their software applications as Web Services for different needs. The Web Service can be built based on varying types of application modules e.g. a Web Service could be (Papazoglou, 2008):

- a fully-fledged business process, such as the automated purchasing of office supplies; or
- a service-enabled resource, such as computation resources, storage resources and network resources;
- a self-contained business task, such as flight booking or hotel reservation service.

The recent Web Service-based software application portrays varying advantages over the common traditional applications (Elgazzar, Martin, & Hassanein, 2014a). These are explained below:

1. Decoupling and just-in-time integration: Web Services allow for building an application with two or more simple services available through service orchestration. However, individual services remain independent despite the service orchestration.
2. Easy and fast deployment: this enhances the development of entirely new Web Services as a result of service reusability and the combination of existing services.
3. Interoperability: the various set of Web Service standards allows any Web Service to interact with one another. This means that the collaboration between different Web Services is highly independent of the implementation language and platform on which it runs. Various applications can thus be deployed as a Web Service and inter-operate with one another.
4. Reduced complexity: the complexity of Web Services has been hidden through technical encapsulations. The service requestors do not care about the implementation

aspect of a service but were only concerned about the features and benefits of a particular service.

5. Robustness for efficient performance: Various service providers compete for consumers thereby improving the services released into the service marketplace. This gives room for users or clients to select, effectiveness and other parameters.

The enumerated advantages influence the expectations and choices of service client when service request are being issued in various service-oriented platforms. This actually necessitates the providers' need to study these properties of the Web Service in order to ensure adequate and updated services are available to the users. The QoS has been a pivotal property that drives users' continual request for service. The next section elaborates more on the nature of QoS in Web Services.

2.3 QoS in Web Services

Every application built on Web Service Technology provides certain functionality. These applications also possess some Non-functional properties. These properties are referred to as the QoS. For example, an application might produce a certain reliability value for every time the service users' issue a request, and this may be based on a particular price for using such service. It should be explicitly stated that reliability and cost constitute a QoS property. The reliability may probably be expressed in percentage while the cost is expressed in dollars. These properties contribute towards determining the level of satisfaction that a user derived from using a service. This satisfaction thus has a relative effect on the success of a particular application which is based on Web Services. Therefore, these properties are important when providing services to clients as the QoS properties go a long way to influence their choices. Various service providers also ensure that the standard of service qualities are strictly adhered to and constantly updated so that the Web Services that they offered ranked among the best that are dynamically selected at runtime (Yu & Lin, 2005).

2.3.1 QoS Models of Web Services

One important issue to be considered in the management of QoS for Web Service-based systems is to identify the QoS models of Web Services. The concept of QoS for Web Services is inspired by early computer application fields such as QoS in Networking (Zhang, Ding, Wan, Gu, & Li, 2010), real-time applications (Schafer, Frankowski, Herlocker, & Sen, 2007) and middleware systems (Zulkernine, Martin, Craddock, & Wilson, 2009). Various scholars have grouped different QoS properties based on their discovery in line with different research conducted. The three commonly used QoS models were presented in Zeng's QoS Model, Ran's QoS Model and UML QoS-Profile. These models are presented in Table 2.1

In the past years, modelling QoS for Web Service has been a challenge for most scholars' (Aagedal & Ecklund, 2002; Gouscos, Kalikakis, & Georgiadis, 2004; Ran, 2003). Though each of the models in the table provided a comprehensive description of the QoS properties for Web Service, only Zeng's model expatiate how various other techniques had been used alongside the use of the latest normalisation techniques when carrying out QoS computation. The use of Zeng's QoS model attests to its general deployment in QoS service computation in many articles and conducted experiments (Liu, Guo, Chen, & Lan, 2009). Attention is paid to Zeng's model because of its suitability in relation to the AMC service parameters. Moreover, the model's broad adaptability to the context of the environment in use and its ease of implementation has been well verified in the research environment (Liu et al., 2009).

Table 2.1 Summary of QoS Models

Zeng's QoS Model (Zeng et al., 2004)	**Ran's QoS Model** (Ran, 2003)	**UML QoS-Profile** (Aagedal & Earl F. Ecklund, 2002)
Throughput	Throughput	Throughput
Response time	Response time	Latency
Cost	Cost	Efficiency
Availability	Availability	Availability
Reliability	Reliability	Reliability
Reputation	Security	Security

Various QoS properties related to Web Service as seen in Table 2.1 are as explained as follows:

- *Throughput*: This represents the number of Web Service requests served during a particular period of time.
- *Response Time*: This measures the expected delay in seconds between the moment when a service user sent a request and the time at which the response is received.
- *Service Cost*: This represents the amount in for of money that the service user would pay for using a particular service.
- *Availability*: This is the probability that a service is accessible for use.
- *Reliability*: This represents the ability of a service to tolerate system or network failures and yet actualises its functionality correctly and consistently.
- *Reputation*: This is measure based on the user experience of using a particular service. It is the measure of services' trust.

These qualities are specified within the XML-based specifications of Web Services. They are arranged in an orderly manner such that each Web Service differs from another based on the different values associated with the varying QoS properties. The arrangements of the QoS properties within the WSDL file are outlined in the next section.

2.3.2 The Web Service Configuration

Each Web Service consists of an associated XML-based document termed WSDL. The WSDL document outlines the Web Service functionality and the interface information. The Web Service implementation definition explains how a service interface is implemented by a given provider. The service interface definition consists of the interface specifications and the business category information that are registered against them in the service registries. Each of the Web Services carries out a set of operations. There are three types of metadata that show the combination of service information in WSDL (Bandyopadhyay, Si, Mondal, & Mallik, 2012). These are:

- **Name and text description**: The registry houses the information that describes the Web Services such as a name, a text description as well as the interface description.
- **Operation descriptions**: The WSDL file carries the operation of each service which is described by its name and text description.
- **Output/Input descriptions:** Each output and input of an operation consist of some parameters that include the name and data type. These parameters could be arranged using complex types in hierarchical order.

The information carried by Web Service need a control mechanism that will ensure that the metadata highlighted above are guaranteed when services are offered by a service provider. The next section describes common control mechanisms used in QoS descriptions.

2.4 QoS Control Mechanisms

There are various QoS control mechanisms among which include load balancing, content adaptation, caching, admission control, request scheduling and end-to-end QoS control (Guitart, Torres, & Ayguade, 2010). The end-to-end QoS-control mechanism is the relevant one in the context of single device-to-device connection within AMC. The end-to-end mechanism focuses on using the dynamic service selection approach along with admission control to regulate the user's satisfaction based on the QoS requirements specified when a service request is issued (Lacuesta, Lloret, Sendra, & Peñalver, 2014). Admission control and Dynamic Service Selection are considered in the next subsection.

2.4.1 Admission Control

This is a validation process in communication systems where a check is carried out before a connection is established to see if the current resources are sufficient for the proposed request to be fulfilled. The performance of a Web Service can be affected by many factors which include network failure, server failure, sudden execution of background jobs by the server etc. Admission control manages the influx of multiple service requests. The issue of network failure has already been taking care of AMC which is a solution to the intermittent connections to the cloud server (Huerta-Canepa & Lee, 2010). The admission control prevents the deterioration of the whole system as the number of service requests increases to ensure that service responses and requests are been regulated.

A good admission control system helps improve the service performance during request over-loading by only allowing a regulated number of client requests at a particular time for a specific service. The two types of admission control schemes used in Web Services are: session-based and request-based. The session-based scheme ensures only the specified numbers of requests are accepted and once they are accepted, it is guaranteed that the request is attended to during the session.

2.4.2 Dynamic Service Selection

QoS-aware service selection refers to the selection that is carried out with the aim of satisfying both functional and Non-functional properties of a client's request. There is an alternative to the dynamic service selection and this is called *Static Service Selection* (Khan, Bashir, Javed, Khan, & Khiyal, 2010). However, the static service selection is time-consuming and is irrelevant in the context of Ad-hoc Mobile Cloud. The dynamic service selection sometimes utilises the help of agents to select services. Such selection could search for a single service that executes a unit operation or combine one or more simple services to carry out a whole single request from the client. For example, a single flight booking service might be requested all alone. Another request might include the aggregation of simple services e.g. hotel reservation, car rental service and flight service. However, the AMC is a model that implements mobile to the mobile service requests and this study ensures that the control system enables end-to-end service request whilst at the same time allows dynamic service selection.

2.5 Ad-hoc Mobile Cloud Computing

Mobile devices which are otherwise called smartphones are becoming a more powerful tool for personal information processing. Mobile devices are ubiquitous thereby making information processing possible on transit. The Mobile Cloud computing architecture assumes persistent connectivity to the cloud server through a wireless connection or mobile cellular network. However, this is not always the case at all times as mobile devices experience intermittent connections. Overcoming this challenge means that mobile devices must have a stable and reliable fall back connectivity in such situations when a disconnection occurs.

Also, this stable and reliable fall back could prove very vital in environments where there were no connections at all. Such environments include a local village where schools could use the

AMC system to share resources. This could also be applied to local health care system where database information could be shared among numerous mobile devices within a health care system to update patient information and records. The local markets consumers could also use the Ad-hoc mobile system to update their goods and services record and also do some uploading to invite customers for shopping (Kovachev, Cao, & Klamma, 2011).

The AMC has become a vital platform for mobile users to make an optimal use of their devices in terms of meeting and bridging the gap between mobile devices and the versatile computer systems. With this improved uses of the AMC, proper selection mechanism is vital to control and structure the simultaneous influx of request from various mobile devices (Akinola, Adigun, & Mudali, 2016). The next section explains the various approaches used for service selection in Mobile Cloud computing.

2.6 Ad-hoc Mobile Cloud Service Selection Approaches

The selections are carried out with a view to improving user satisfaction by invoking the most relevant service. However, service selection in a mobile environment is often preceded by service discovery. Mobile nodes are needed to be discovered before selection could be made. The discussion in this dissertation focuses on the service selection aspect in a mobile environment and no discussion was made regarding service discovery mechanisms. Therefore, it is assumed that the system is aware of all the available service providers of certain service and routes to those nodes as well. Moreover, the system also has full knowledge about the useful information about the service providers as well as the QoS rendered by them. In the same vein, the chosen Web Service selection approach does not consider the network parameters of the mobile nodes in the sense that AMC is actively serving as a back-up for network interruption in MCC. Thus, it is assumed that a stable WiFi connection existed within the cloud among the

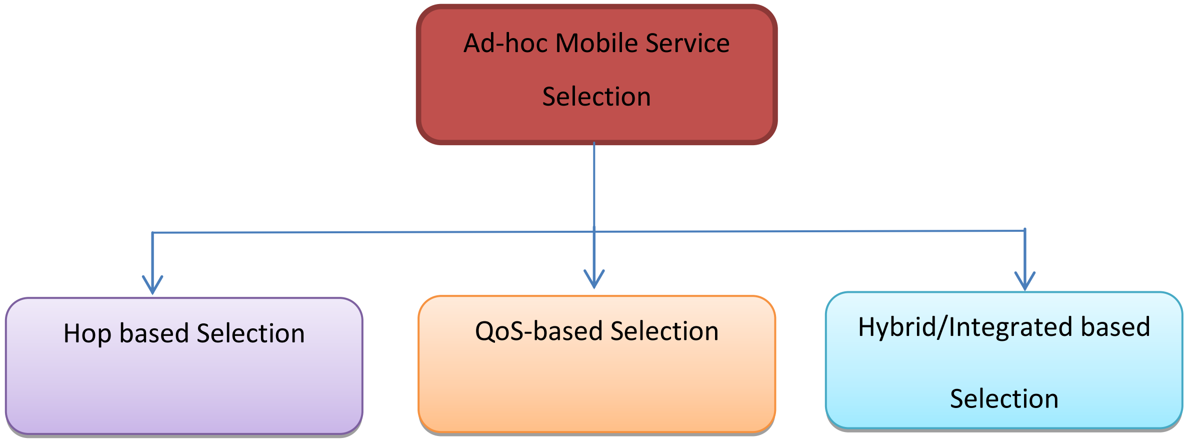

Figure 2.2 Types of Ad-hoc Mobile Service Selection **(Z. Zhang et al., 2008).**

mobile peers. Since the focus of this study is not on mobile service connection, therefore, network connectivity is not addressed.

The nature of mobile environment has a great influence on the classification of the mobile service selection approaches. There are three major Ad-hoc mobile service selection approaches which are Hop count-based selection, QoS-based selection and Hybrid/Integrated based selection as shown in Figure 2.2 (Zhang et al., 2008). The Ad-hoc system considers the positioning of the mobile devices in the cloud system as an important factor whose effect affects the selection process. The behaviour of the mobile service in terms of its quality is also very important especially in a mobile environment where mobile devices change their position often, based on the mobile user.

2.6.1 Hop Count Based Selection

Several approaches have been used in the literature to address prompt service selection in the mobile environment especially when response time is very pivotal. Hop count has been used for time constrained mobile service selection in wireless mesh networks and sensor networks etc. (Korkmaz & Krunz, 2001; Lacuesta et al., 2014; Varshavsky, Reid, & Lara, 2005). This selection approach is often deployed in a time critical situation where services are needed within the shortest time possible for example, a life-saving situation.

This system considers time as a very important issue and therefore looks or searches for closest node to the requestor. This has the advantages of *i.*) lower risk of downlink with the least distances apart; *ii.*) lower rate of battery consumption with the reduced distances between the nodes and *iii.*) lesser possibilities for the two nodes, to be out of reach of each other (Yang, Galis, & Chen, 2010).

2.6.2 QoS-Based Selection

The World Wide Web Consortium (W3C) working group in 2003 explained various QoS properties for Web Services in their 25th November 2003 publication of W3C. It comprises of a number of generic items for cross-reference between the possible needs of service users and various functions supported by Web Services. These include availability, accessibility, reliability, integrity, capacity, scalability, interoperability, accuracy, exception handling etc. The International Organization for standardization, ISO 8402 describes quality as **"*the totality of features and characteristics of a product or service that bear on its ability to satisfy stated or implied needs*"**. According to Deora et al., (2003), what defines quality is uncertain, and different views arise in different studies and from varying perspectives. The most common three views of QoS are discussed thus as:

1. **Reputation**. Reputation as a function of quality is ascribed to users' experience and expectation from a WS. This value is built collectively over the time of service's existence from users' feedback (Averbakh, Krause, & Skoutas, 2009). For example, if a WS has consistently delivered a specific functionality with specific performance levels at all time of its operation, then it provides a good QoS.

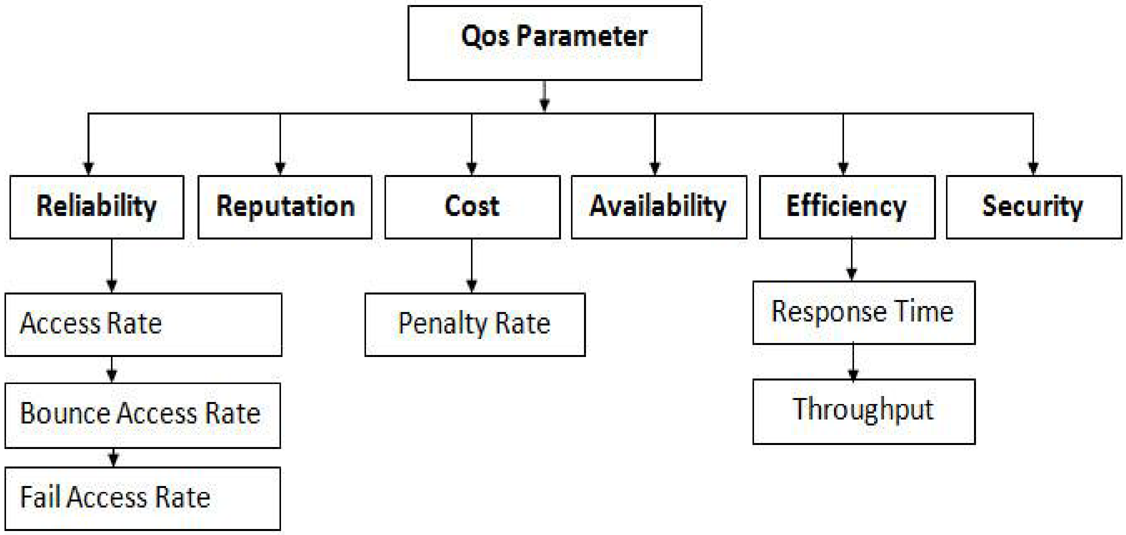

Figure 2.3 Non-functional property classification **(Kayastha & Baria, 2015).**

2. **Conformance**. The conformance expresses quality as a means of attaining some specifications. For example, if a Service Provider A has specified that his WSs will operate at the accessibility of 0.9999 over a period of time and this was shown to be true, then the Service Provider A is recognised as offering a good QoS.
3. **Functionality.** This considers the amount of functionality that a service can offer to its users. For example, if Service Provider A allows a user to check available products online as well as pay for the selected goods, and if this payment functionality is not realisable in Service Provider B or C, then Service Provider A is offering a better quality than B and C.

The QoS of WSs is a widely embraced tool mostly used in querying services for invocation in a service provisioning environment. The Web Service with the highest QoS is chosen to execute the request. The response time of a Web Service is one of the qualities that determines the performance of a service. Though several other qualities of service metrics are available, yet the time it takes to receive a response to a service request by a user is of great importance. For example, considering an asthmatic patient with body monitoring device, the sudden development of the illness needs an urgent attention especially by the nearest clinic around. The monitoring device is expected not to only alert the patient of the re-occurrence of the symptoms earlier but also chose within the community where he/she resides, the nearest health care provider within the shortest time possible.

The Figure 2.3 shows how the QoS parameters are interrelated to one another. They all address “how” Web Services perform in carrying out a specific task. The way such tasks are affected has a direct relation to the user satisfaction thus the users tends to seek service with the best QoS aggregate scores that satisfy their specification.

QoS-based service selection performs a pivotal role in the process of service composition as well. QoS overcomes the problem faced in functional based service selection such as providing similar functional semantic properties, which might lead to the problem of differentiating available services. QoS-based service selection approaches according to Maximilien & Singh, (2005) can be sub-divided into two primary groups as *i*) Design-time approach in which an application designer uses service registries alongside with service descriptions to select and test service binding. OWL-S also, according to Sycara et al., (2003) is an example of a loaded service ontology used for semantic service discovery and selection. This is always supported by QoS as the Non-functional properties being considered during trial and error tests of the selected services. *ii*) The run-time approach allows the discovery, selection and binding of services to occur based on the Non-functional properties of services as the values of those properties were at the point of service invocation. This approach often requires the use of various QoS properties, metrics, models and the middleware.

2.6.3 Hybrid/Integrated-Based Selection

The integrated approach combines the properties from the route/hop count with the QoS approach to arrive at the optimal solution for a particular request. Due to the nature of the request that a service user specified, there might be a need to prefer particular properties more than the other. For example, time constraint might be very imperative for a user in need of urgent medical attention or user who needed to catch up quickly on a flight whereas for another requesting for an educationally related service might not be as more urgent. Mobile devices enable the use of various services on transit through a wireless connection either to the cloud server or within themselves. Thus, the integrated selection approach seeks to address users' satisfaction by ensuring optimal service is being selected at every request.

2.7 Chapter Summary

This chapter has presented the foundational conceptualisation for this research. This chapter has also discussed the gradual development of computing technologies towards the AMC computing paradigm. Fundamental computing paradigms, including Grid Computing, Cloud Computing, MCC as well as service provisioning challenges in MCC were introduced. Since service selection is a challenge peculiar to all computing paradigms as users encounter optimal satisfaction issues, the main focus in this research is based on ensuring optimal satisfaction by mobile users. Mobile Service selection analysis has been carried out in this chapter, and it was carefully noted that user satisfaction is the driving force behind service selection in which qualities of such services play a pivotal role. Finally, chapter three discusses more fully the various ways in which users express their interest via the QoS to influence the selection of service in existing paradigms.

CHAPTER 3

LITERATURE REVIEW

3.1 Introduction

The Mobile devices such as Personal Digital Assistance (PDA), WebPad, Tablet PC, and cell phones are the key innovations to ubiquitous computing. This is because they are carriers, as well as communication tools that are not bounded by time and place (Akinola, Adigun, Akingbesote, & Mba, 2015). Mobile devices function as tools through which wealth of rich experience is brought to mobile users, in that they allow the invocation of services on transit. Mobile devices have become an indispensable tool to various categories of users due to its tremendous advantage especially for invoking services from the cloud server. This paradigm that allows mobile invocation is termed MCC (AlShahwan & Faisal, 2014; Bahl, Han, Li, & Satyanarayanan, 2012; Wu, Wang, & Wolter, 2012). In spite of the increasing popularity of this paradigm, it has not been able to attain its full capacity due to its inherent challenges peculiar to mobile devices such as resource poverty, intermittent disconnection and latency issues (Fangming, Peng, Hai, Linjie, & Jie, 2013).

With the advent of the wireless connection and various Cellular Network Service providers, mobile device users enjoy the use of cloud services anywhere. Mobility constraint is no more a challenge when services are needed by the users. However, one of the challenges is that of disconnection from the source of services due to various reasons like the network providers, long WAN distances and low battery power (Fangming et al., 2013). In order to tackle this challenge, the AMC evolved. See Kovachev et al, (2012); Fangming et al, (2013); Kaushik & Kumar, (2014) and Lacuesta et al, (2014) for more details on AMC.

The AMC is a group of mobile devices that serve as a cloud computing provider by exposing their computing resources to other mobile devices (Fangming et al., 2013; Kaushik & Kumar, 2014; Kovachev et al., 2012; Lacuesta et al., 2014). However, the realisation of the full potential of any service provisioning platform requires some technological advances in areas of service interoperation, discovery, selection, composition and orchestration (Swarnamugi, 2013). Various approaches have been used to achieve the above task in various existing paradigms like grid and cloud. However, it is not clear if such approaches are directly applicable to mobile Web Service provisioning. The approaches may also be enhanced to fit AMC computing platform appropriately. Moreover, the peculiarity of independent mobile devices existing within an environment must notably address the issue of discovery and selection of peers. In this chapter, the researcher discusses the state of the art in the selection of services in any service oriented environments. The study aimed at analysing whether or not the existing selection methodologies will be suitable for service selection within the Ad-hoc mobile context.

On this chapter, section 3.2 discusses service selection requirements while section 3.3 critically analyses service selection methodologies in service-oriented computing. Other approaches to mobile service selection were discussed in section 3.4. Section 3.5 explains the selection in AMC environment and lastly, the summary of the discussions is contained in section 3.6.

3.2 Service Selection Requirements

It is important to consider the various service selection requirements before examining in detail the various approaches used in service selection. This is to enable the researcher to know what is required of each of the approaches in delivering optimal service selection within the AMC environment. For any service selection approaches, the basic requirements according to Swarnamugi, (2013) are given as Customer Service Requirement, Service Offerings and the Summation of Evaluation Results.

3.2.1 Customer Service Requirement

There are two main categories of customer service requirements which could either be complex or simple (single). The single requirement does not look out for composite services to meet up for a customer query. The single requirement checks only for a simple service that matches user's Non-functional service description. However, complex requirements often consider the service functional and Non-functional parameters to ensure each of the combining services meet up with the desired qualities. Mostly, complex customer requirements are usually an aggregate of single services hence, they are referred to as composite requirements.

3.2.2 Service Offerings

Service offerings are used to create refined levels of service for existing service integrations. They specify service provider offers by stipulating the functional and Non-functional aspects of the services through service domain ontology and Quality of Service (QoS) ontology respectively. The service domain ontology provides us with the parameters that classify the functional properties of service offerings while the QoS ontology provides the consumers with the Non-functional properties of service offers from the service providers. One major challenge under the service offering is that of matching consumer preferences in terms of QoS parameters with the service categorisation that is registered in the QoS ontology. Different paradigms use different mechanisms to achieve this task. The discussion about the methodologies is in section 3.3

3.2.3 Evaluation Results

The selection of services is based on the result from the QoS evaluation. The computation derived from the consumers' QoS request is compared with that of the offered provider services. When the service request is a complex requirement, it could look through the individual single services for their conformity with others for the desired output for example, in the case of Grid or Cloud Computing.

3.3 Service Selection Methodologies

Service-Oriented Computing (SOC) is a computing paradigm which uses services as underlying elements to enhance the development of application processes in a rapid, low cost and heterogeneous environment (Liu, Wang, Orgun, & Lim, 2010; Swarnamugi, 2013). This paradigm enhances services to be described, published, discovered and selected for service consumers in a heterogeneous service provisioning environment. Research from Swarnamugi, (2013) reveals that most works on Web Service and the semantic web are based on service discovery while the issue of service selection is yet to be fully addressed. Apart from the fact that the issue of service selection has not been fully addressed, the issue of selection of service with similar functions is another challenge (Iordache & Moldoveanu, 2014; Wang, Chao, & Lo, 2010). Service discovery deals with locating service implementations that meet a particular description. The service consumer specifies the kind of service to be consumed and the discovery mechanism locates which service descriptions are suitable to be considered in carrying out the task. In the same way, service selection entails choosing a service implementation, among those that are discovered for the given description.

The various methodologies used in the selection of services is shown in Figure 3.1. Each of the methodologies was discussed in the following subsection.

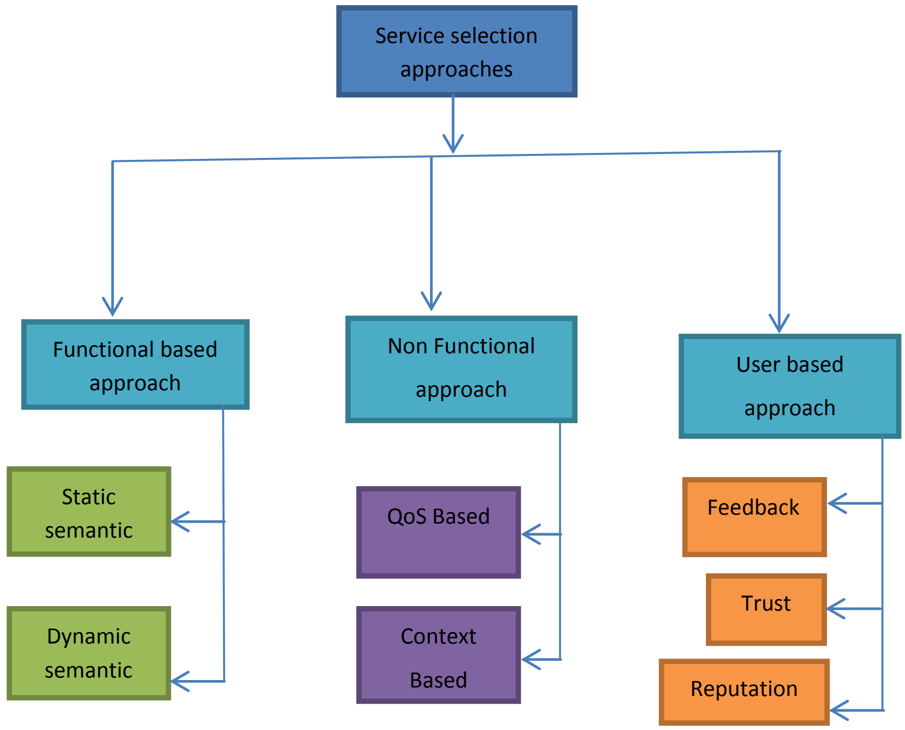

Figure 3.1 Web Service selection approaches **(Swarnamugi, 2013).**

3.3.1 Functional Based Approach

The selection of an appropriate service is based on the retrieval of a functional description of a service from the registries and then certifying that the description and requirement of the interfaces match each other. To fully realise the potential of Web Services, there is a need for advances to be made in the areas of service interoperation, discovery, selection and orchestration (Swarnamugi, 2013). Most of the web services are converted into the semantic web to enhance the easy description of Web Service functionality. The Semantic Web allows solving the integration problem between different systems and organisations. The semantic expressions in Web Services provide for the inclusion of Functional Semantics.

The Functional Semantics, therefore, takes into consideration the provision of satisfactory result that meets up with the customer interest. The functional property is also called the functional semantics of a service that describes what a service really does. The semantic Web Service selection was implemented in the work of Klusch and Kapahnke using the hybrid version of SAWSDL matchmaker called SAWSDL-MX (Klusch & Kapahnke, 2008). The architecture for the matchmaker is as shown in Figure 3.2. The selection uses crisp logic based matching and IR-based text retrieval strategies for service selection. The logic based matching uses subsumption reasoning to implement the service functionality expressions for service matching. The reasoning assumes the logic-based selection of appropriately regulated services for various clients. The algorithm for the logic reasoning considers the order of filters based on the degree of relaxation provided to sort out selections. For example, it would not be very appropriate to return a service providing information on laptops, if the user explicitly requested information on a very special product of a personal computer like Dell.

The IR-based strategies use token-based similarity measures to carry out syntactic matching. The syntactic similarity value is computed for every pair of service offer and the request operation which is used to rank operations with the same logic-based matching degree. The result of the weighted keyword vectors of input values and output values for every operation are generated by unfolding the reference concepts in the registries. Generally, the semantic functional approach for service selection is divided into two major sections as: *Static and Dynamic Semantic* approaches.

3.3.1.1 Static Semantic Approach

This presents the properties of the message and the operation semantic of a Web Service. The properties of messages include the various parameters passed during service selection such as the Data type, Language, Unit and the business role. The message types passed includes the serviceability, provider type, purpose, consumer type.

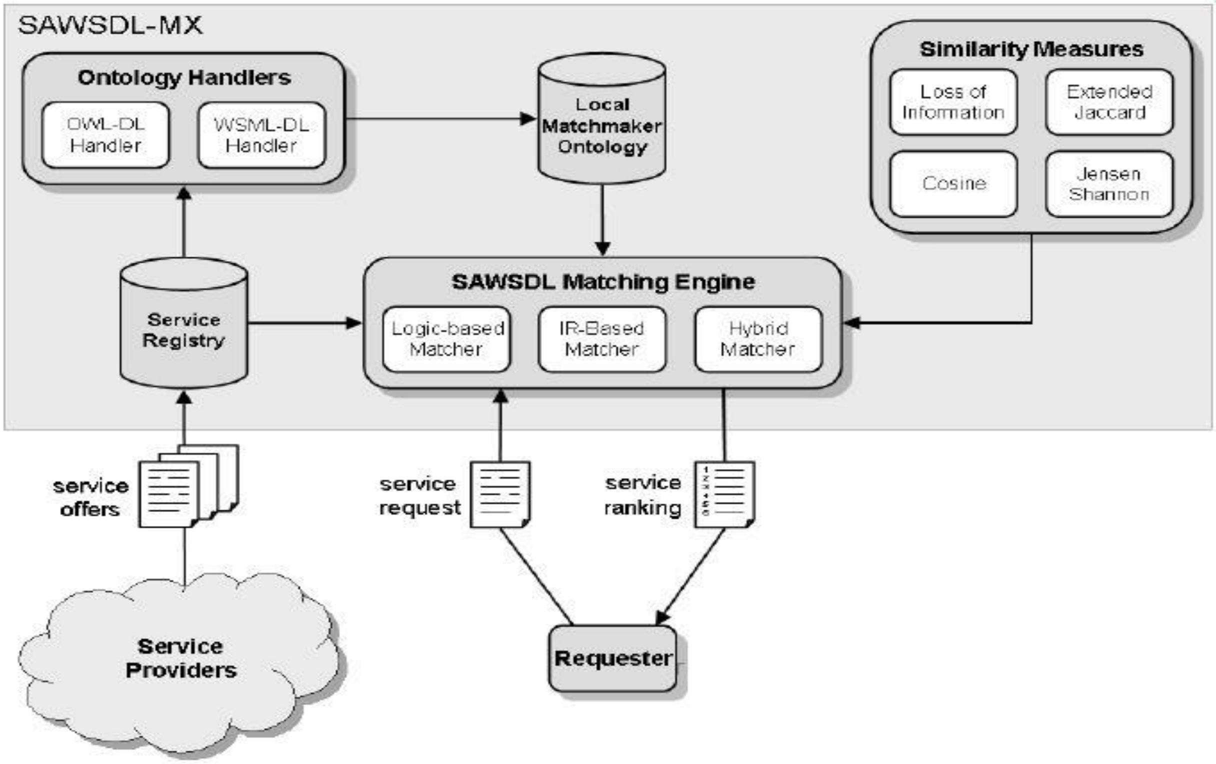

Figure 3.2 SAWSDL-MX Architecture **(Klusch & Kapahnke, 2008).**

3.3.1.2 Dynamic Semantic Approach

This presents the properties of the behaviour and the operation logic of Web Services. However, a dynamic semantic approach often contains more than one service provider when used for service selection (Swarnamugi, 2013). This situation leaves the service users with the only option to choose services that match other preference they preferred, which is referred to as the Non-functional properties (Kun, Hong, Liming, & Jian, 2010). The drawbacks of the functional service selection approach make the service provider seek for a more informative alternative that could differentiate their products from that of other in relation to its performance. The use of the Non-functional properties generated two major concerns such as how extra properties of Web Services could be described and how appropriate service could be selected based on their described properties (Wang, Vitvar, Kerrigan, & Toma, 2008). Moreover, considering the AMC environment, the use of highly logical algorithm and complex ontological data storage generate a challenge for memory restrained mobile devices therefore making the approach unsuitable for deployment.

3.3.2 Non-functional Based Approach

It is a common experience in a service provisioning environment for Web Services to provide similar functionalities with different Non-functional properties. Customers or service users are always left with the option to choose out of various services that meet their basic needs. Literature shows that there are Web Services which satisfy users more than the other depending on the specified QoS by the service consumer (Sivakumar & Magendiran, 2012). Therefore, to differentiate the various Web Services during service selection, their Non-functional properties need to be considered. The Non-functional properties are categorised into Quality of Service (QoS) and context-based services (Swarnamugi, 2013). The properties of the QoS of a service may be reliability, call cost, response time, security, availability. The context could also include properties like service context (provider's details, service descriptions), device context (location, reputation) and customer's context (intention, e-mail, application, customer's name, termination of software and hardware). These properties are very important to the users hence the need to be taken into consideration when selecting a Web Service.

3.3.2.1 QoS-Based Service Selection Approach

In Susila et al, (2012), the author proposed the architecture for a Web Service selection using the QoS based approach in a service provisioning environment where the service user searches the service registry (UDDI) for a list of all services that address the concerned request. Sometimes, the service consumer uses the help of a service broker to derive a better Web Service as well as differentiating the services from the registry. The proposed architecture is as shown in Figure 3.3

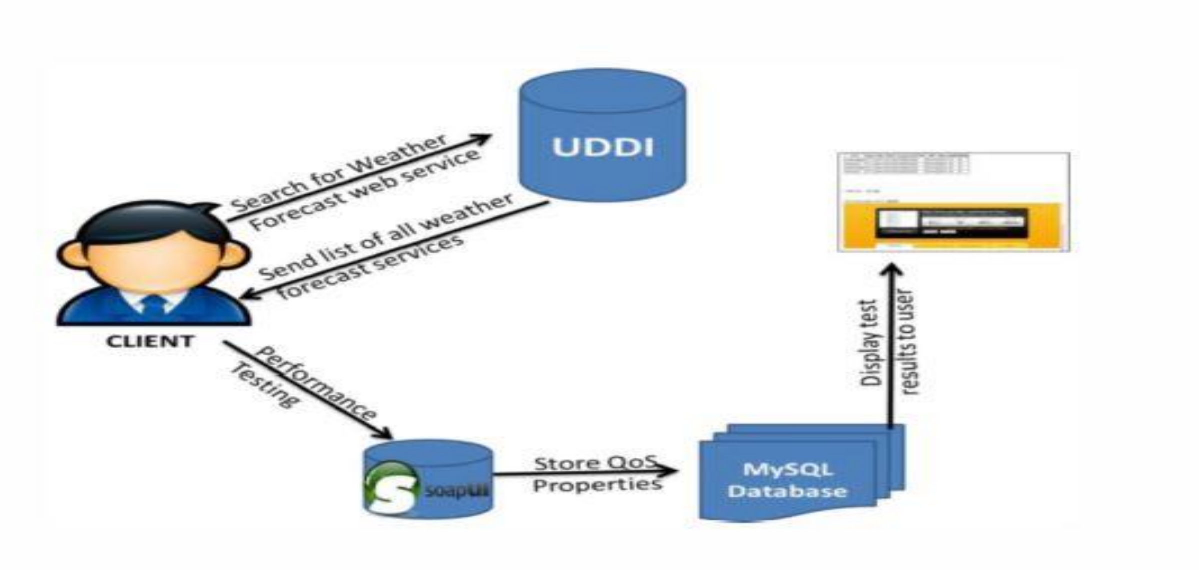

Figure 3.3 QoS based with Broker Architecture **(Susila et al., 2012) .**

The broker deploys the SOAPUI (an open-source Web Service testing application) as a tool to test the performance of services saved in the registry. This performance is expressed in terms of QoS values of each of the Web Services uploaded into the registry for clients' use. The QoS properties derived from the SOAPUI testing on the Web Service are shown to the service consumer to select the appropriate service that meets the requested requirements. However, this approach will be unsuitable for the Ad-hoc mobile environment where several competing Web Services are available. The registry option is also not suitable for mobile environment as devices have limited memory to contain such registries.

Several authors have worked on the use of the QoS-based approach. For example the author Xin et al, (2014) solves the challenge of location-based services selection. They proposed a framework for location-based selection which is shown in figure 3.4.

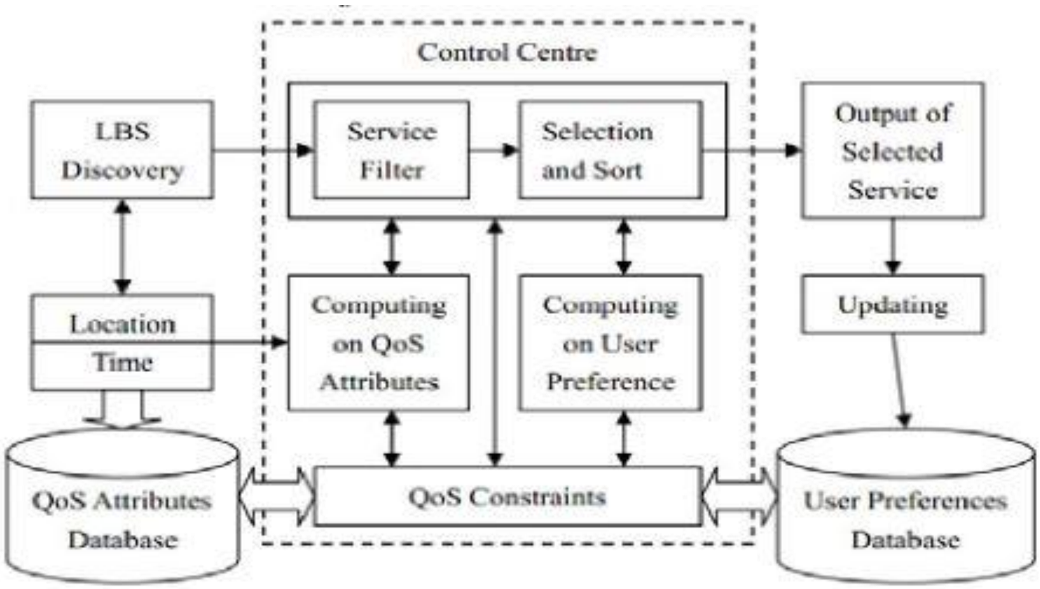

Figure 3.4 Framework for LBS selection **(Xin et al., 2014) .**

The framework enhances modelling and updating of group user preferences as well as mobile service selection. In this work, QoS attributes were combined with user preferences of a group of consumers to propose an algorithm and a mobile service selection model to solve the selection challenge. However, the selection inferences are based on a group of collective users which is not perfectly suitable for an AMC environment that is envisioned. Moreover, the test was carried out under a well-connected internet environment as against intermittently connected Ad-hoc setup.

Also, Amoretti et al., (2008) proposes reputation-based service selection framework which emphasises on intra-and inter-SOP (service oriented peers) module interaction. The framework contains a component called SAFE that ensures a mobile peer computes the reputation of a provider based on its previous experience. The SAFE component is assigned the task of "voting strategies" to ensure that proper record of the aggregated reputation influences selection decisions. However, this work was silent about the situation where Web Service attains similar computation aggregates thereby leaving behind for us a gap to fill.

Furthermore, the study conducted by Akingbesote et al., (2013) proposed a Quality of Service aware Multi-Level Ranking Model (MLRANK) for selecting an optimal Web Service in cloud computing. The study addressed the occurrence of ties within a number of services that are available in the UDDI. The study highlights the challenge of selecting an optimal Web Service when there are ties with the used criterion where performance alternatives have the same score. The proposed model is as shown in Figure 3.5. The study achieved optimal selection by comparing the service consumer's QoS preference with the Web Service QoS offerings. The provider offering that best fits the QoS preference is taken as the optimal Web Service. The study used non-deterministic QoS metrics and concentrate on various information services to test the performance of the proposed model. The result showed the model can satisfy service consumers' request based on Non-functional requirements. However, this study serially considers each of the selected qualities one after the other and check which one is higher than the other to make a selection. This approach is not reliable in the context of AMC. This is because decisions are expected to be as fast as possible. In addition, the work never considered the possibility of the consumers having relatively equal priority for the specified QoS properties which makes the selection yardstick less efficient.

3.3.2.2 Context-Based Service Selection Approach

A Web Service is termed to be context-aware if it is able to make use of the information in its environment to provide users with personalised and customised services. Keidl & Kemper, (2004) proposed a context processing framework to influence context-aware Web Service provisioning. This framework is as shown in Figure 3.6 containing three main parts, which are the main Web Services, context services and the context plugins. The major flow of process

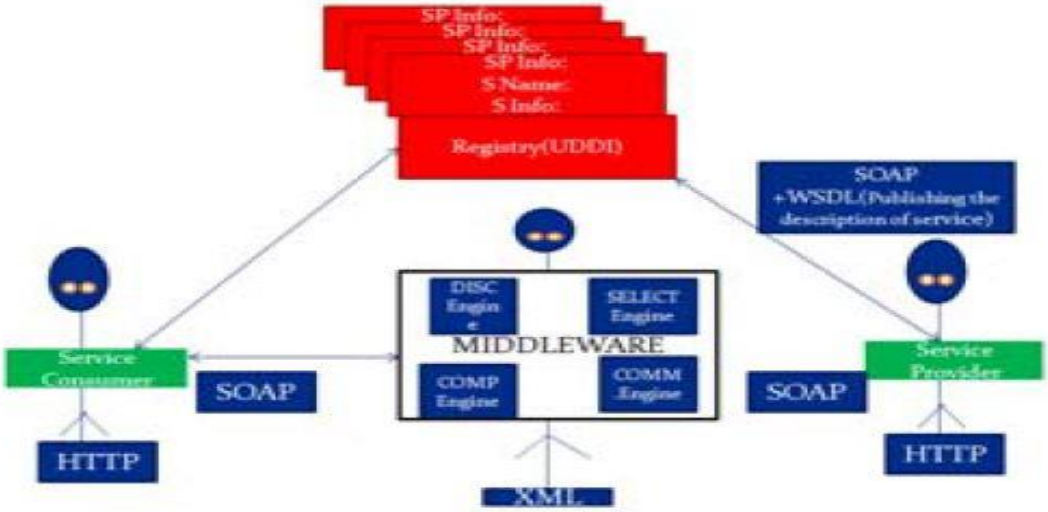

Figure 3.5 QoS-Aware Multi-Level Ranking Model **(Akingbesote et al., 2013).**

within the framework occurs through the context plugins and context pre- and post- process Web Service messages using only the provided context information.

Context plugins are Java objects which implement a dedicated service interface. They are loaded by the service platform during service initiation and as well support locally executed Web Services since the context plugins are not usable if they are not locally available. Context services, on the other hand do not need to be locally available as they have specific interfaces, defined by the WSDL standard protocol which are usually available on the Internet.

However, the framework uses insufficient number of context parameters among which are location and client addresses. The location specified through the SOAP message body of the Web Service is fixed, thus, it is not suitable for a dynamic environment like AMC system. Since the mobile device keeps changing location, the use of location as context will be irrelevant to service selection.

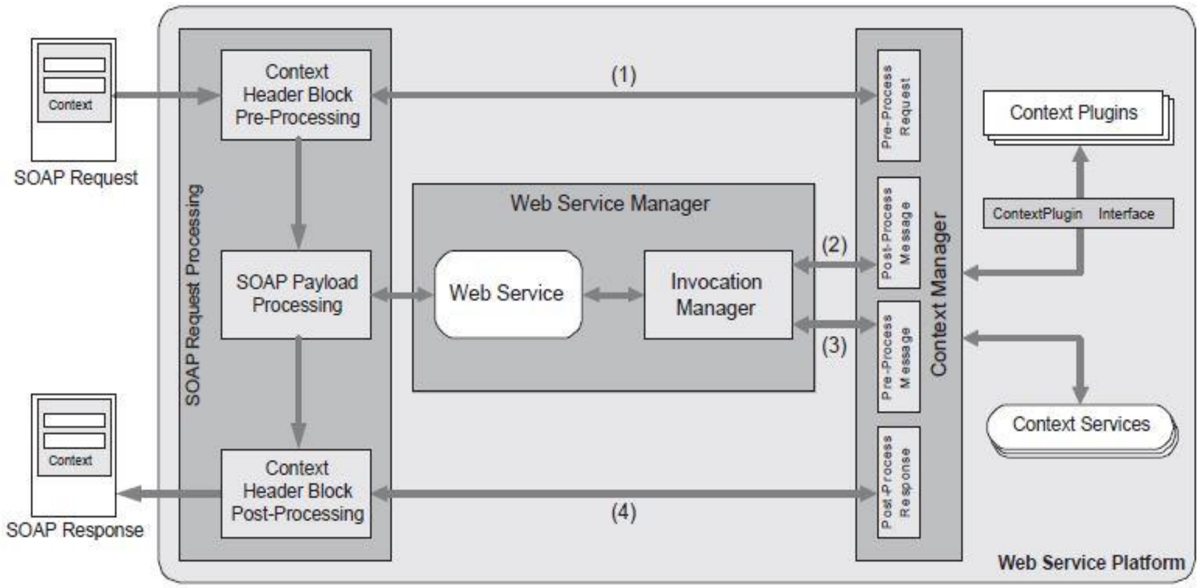

Figure 3.6 Web Service Context Processing Framework **(Keidl & Kemper, 2004).**

3.3.2.3 User-Based Selection Approach

The Internet World has recognised the importance of user feedback-based approach in situations that warrant accurate decision making Liu et al, (2004). The service consumers usually like to see previous users' experiences. The measure of the trustworthiness of a particular Web Service is termed the Reputation. This measure mainly depends on the end users' experience of using the particular service. Various end users may have dissimilar views about the same Web Service. However, the reputation is expressed as an average ranking given to a Web Service by the end users thus deriving a range of ranking from these end users. The acceptability of the reputation also depends on the amount of Trust that the service consumers' have on previous end users. The trust further becomes an important parameter not only between the service end users with regards to the reputation that was provided but also between both the service requesters and the providers. The trust enhances fair computation and ensures QoS of Web Services provide little or no differences between the user's request and provider's offering.

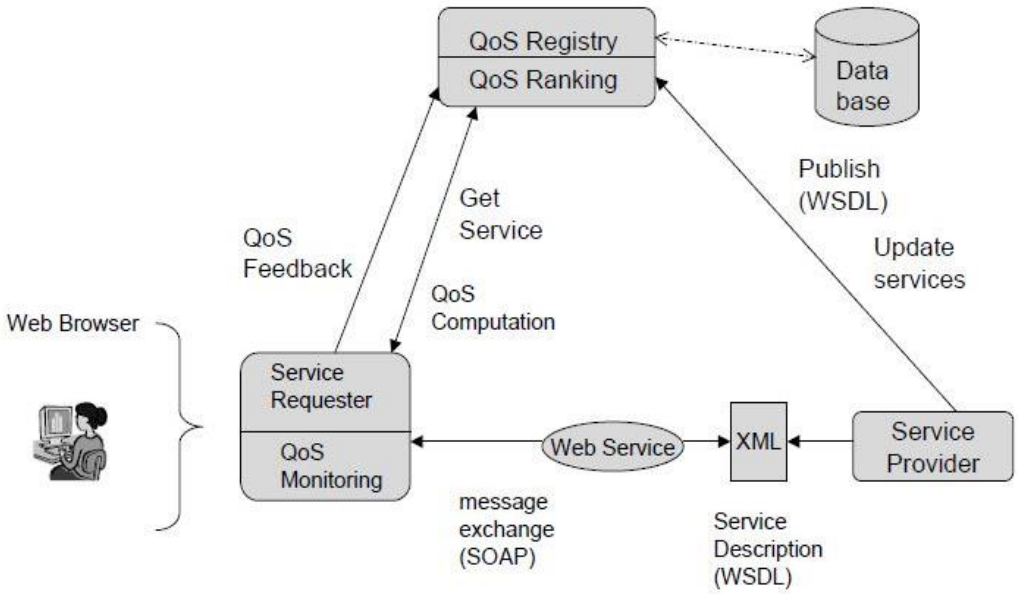

Figure 3.7 Architecture diagram for QoS Registry **(Liu et al., 2004).**

Thus, the reputation from end users enhances the privilege of building a trust for Web services. Various research studies have utilised these feedback tools to influence Web Service selection. One of the related studies is that of Liu et al, (2004), which proposed a QoS Registry that utilises the end user feedback to update the published Quality of Services in the registry. It used an active monitoring and active user's feedback to decipher an up to date QoS from the recent service consumer. The architecture consists of the Database, QoS monitoring and Service requester, QoS Registry/Ranking and the Service Provider. See Figure 3.7.

This system determines the QoS of a service provider through calculation of the difference between the published service provider value and the user feedback response. A higher difference value typically shows a lower QoS rating for that particular service provider or Web Service. The service users are allowed to provide a feedback through the use of a pair of key. The users authenticate with the aid of these keys and they are allowed to update the QoS criteria based on their experience. However, the provider would not always be available to affirm and see to the proper updating of the QoS properties. The high possibility is that both the user and consumer would likely not be available at a similar time therefore resulting into overloading of the system through unconsumed user requests.

3.4 Selection in the Ad-hoc Mobile Cloud Environment

In the early days, Web Service functionalities were sufficient to carry out Web Service selection but with the recent increase in Web Services available on service provisioning platforms, the use of Web Service functionalities has become inefficient (Liu et al., 2004). This inefficiency is even more apparent with the various service provisioning infrastructures coupled with the use of all forms of ubiquitous mobile devices that are capable of sharing resources (Elgazzar, Martin, & Hassanein, 2014b; Huang & Deng, 2014; Swarnamugi, 2013). Various service provisioning platforms have begun to consider the Non-functional properties of Web Services during service invocation. Thus, this calls for the use of various approaches to combine functional and Non-functional properties for Web Service selection within the Ad-hoc mobile environment. The common challenges related to these evolving approaches are summarised into the following three categories:

1) Most of these approaches have proved to be less efficient especially as the number of resourced-mobile devices increases (Shah-Hosseini, 2007).

2) Many of the implemented algorithms are not suitable for the AMC environment because of its dynamic nature.

3) Also, eradicating the approach of taking arbitrary decisions during Web Service ties is a challenge. This challenge of Web Service selection ties has been a major issue also in many service provisioning platforms which have not been fully addressed (Akingbesote

et al., 2013).

Therefore, improving service selection in the context of the Ad-hoc mobile environment involves the integration of functional properties with a number of individual Non-functional properties. This study formulates a service selection mechanism that ensures optimal service selection within a Mobile Cloud using an m-Learning use-case scenario. In the next two chapters the design and simulation of a service selection mechanism that incorporates the use of Qualities of Services are discussed.

3.6 Chapter Summary

In this chapter, the researcher examined the state-of-the-art on service selection in the context of SOA, Grid, Cloud MCC and AMC. Various approaches to service selection were studied. The issue of different methodologies was also studied alongside with the basic requirements for service selection. The chapter highlighted customer service requirement, service offerings and summation of evaluation results as key important requirement needed for any optimal service selection to take place. The chapter also discussed functional, Non-functional and user-based approaches as the basic methodologies through which selection of services are made. The chapter further analysis various research works that use each methodologies for service selection and their side effects. The review shows that several works implemented QoS-based service selection but were silent about the challenge of two or more Web Services, having the same QoS output computation. This opens up a gap in the area of service selection which is yet to be filled. This chapter pointed out that Web Services ties are a challenge in service-oriented computing because it determines the level of satisfaction that a particular user achieves. Thus, for users to achieve maximum satisfaction during service request there is need to proffer solution to the challenge of service selection ties. The solution proposed in this write up was discussed in chapter 4 and chapter 5 contains the implementation.

CHAPTER 4

DEVELOPMENT OF AN AD-HOC WEB SERVICE SELECTION MECHANISM

4.1 Introduction

As discussed in the previous chapter, one major challenge being faced by service providers is the selection of optimal Web Service when there are ties with the used criterion, that is, where the performance alternatives have the same score. This research work proposes a solution that builds on the existing selection mechanism through incorporating a user feedback rating mechanism to select the best service during Web Service ties. The user feedback here is the QoS indicators that were used to rate the performance of the consumed Web Services. This work is based on a single service. To achieve this, the researcher first normalises the parameters and then calculates the aggregate scores based on the user's preference. This mechanism that is built on existing selection method uses the user rating system to collect the user ratings and manages it. This is to ensure prompt optimal service selection responses in the AMC. The organisation of this chapter is as follows below:

Section 4.2 presents the domain specific scenario which highlights the need for optimal service selection in the AMC. Section 4.3 addresses the design requirements. Section 4.4 presents the proposed Service Selection Mechanism and its design components. In section 4.5, the researcher provides detailed discussion on each of the components in the design. The QoS-Aware selection process design is illustrated in section 4.6. The application as well as the contribution of the proposed selection mechanism to the GUISET infrastructure-less platform is explained in section 4.7 while section 4.8 concludes the chapter.

4.2 Domain Specific Usage Scenario

Mobile devices are important accessories to all and sundry in the current 21st century. The reason is attributed to the vast growth in their usage as a medium of communication as well as the easy service utilisation of mobile services that are available. Often times in organisations, institutions and places of learning, the use of mobile devices is unavoidable because each user connects via their smartphones for cloud services. As discussed in the previous chapter, the major challenges these users encounter is the intermittent disconnection that mobile devices experience- the phenomenon which gave birth to the *Ad-hoc Mobile Cloud*.

For a proper understanding of the application of this work, a typical school scenario is presented based on the assumption that mobile devices in these schools could easily form AMC through a wireless connection. The reason for this assumption is that most schools consume similar educational resources. This scenario visualises Martin as a student from a particular high school. He uses a Web Service in the cloud to download some study materials and invoke other arithmetic educational related applications. He experiences intermittent connections due to an unreliable internet connection. Every time, he initiates the service, he loses connectivity during the process and this repeated action has cost Martin in terms of data bundle subscriptions. He, therefore, connects to the local wireless access that is available within the school premises via a laptop which acts as a server.

The wireless access authenticates his device and on confirmation, he was able to join the virtualised Ad-hoc network. Martin then specifies the service name and service QoS properties he needs via a GUI on his mobile device. The returned service to Martin showed that he was not satisfied with the *maths_service* because the approximation was indefinite and yet costly. When Martin now asked some of his friends who had used the same service earlier, the report he got based on their evaluation about the *maths_service* was below average. If there had been a selection approach that used the feedback mechanism based on users rating, Martin would not have accepted such a costly and unfavourable *maths_service*. The motivation for this work is based on the non-availability in the literature of a feedback based mechanism for rating services as a yardstick for service selection.

4.3 Design Criteria

Based on this insightful knowledge in Section 4.2 and the literature review in Chapter 3, the following design criteria were extracted for the designing of the proposed Web Service selection mechanism for the AMC.

1. **Automate and Manage QoS data collection-** The proposed system must facilitate an easy way of collecting user request properties without experiencing any difficulty. Users should be able to interpret the QoS parameters specified on the user interface to describe their interest appropriately.
2. **Automate QoS data processing and request matching-** The system accepts the QoS parameters and ensures that they are properly normalised and accrued appropriate weights. These weights that were assigned become the tool for determining the matching of the service requests with the responses.
3. **Determine Web Service ties during the selection process**- The system should be able to recognise a situation where ties occur. This situation should prompt the feedback section to identify from the recorded ratings which of the services is rated among the ties.

The most rated of the services that fall within the same ties should be assigned for the service requestor.

4. **Auto request for rating especially after each service invocation** - The performance of the system needs to be enhanced with the feedback features that enable rating of every consumed service.

4.3.1 Basic Mechanism Assumptions

These are basic assumptions that were made towards the development of the proposed selection mechanism:

1. AMC regards portable laptops as mobile machines which can be used as a participating servers within AMC environment.
2. Smooth connections exist between the server PC and various connecting mobile devices within the specified environment and proper terms of agreement exist between the owners.
3. Devices within the environment are smartphones from various vendors but could perform similar functions.
4. Participating mobile devices upload values of Web Service QoS parameters correctly, in the form of WSDL files which are reliable and dependable.
5. The AMC participants generate honest reports about consumed Web Services to the feedback composer for service selection.

4.4 Mechanism Design

This section presents the design model of the service selection mechanism in the form of Unified Modeling Language (UML), the modelling language used for expressing software

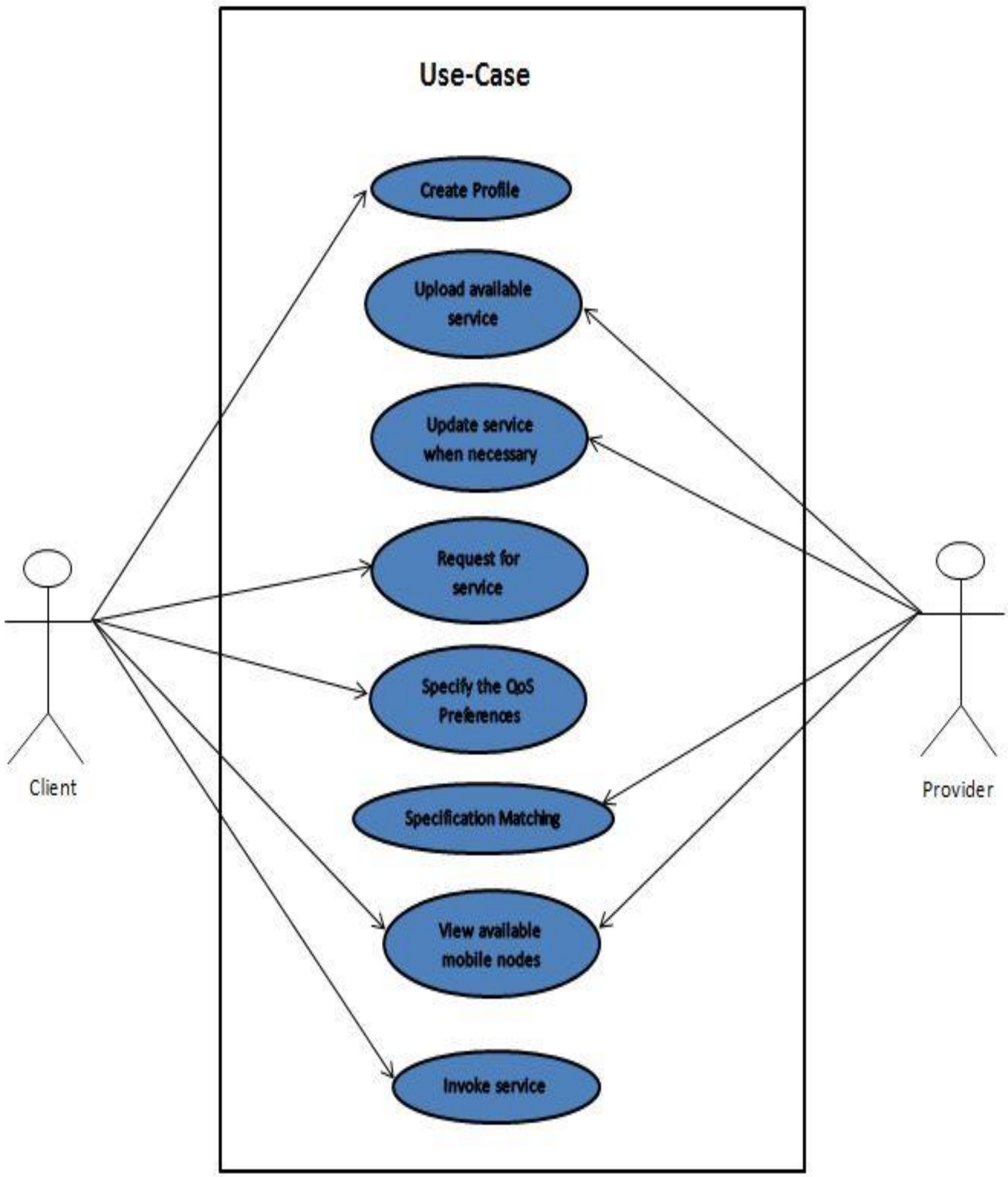

Figure 4.1 Use Case Diagram.

implementation design. This design details the use of case diagram, activity diagram, class diagram and entity relationship diagrams to express the mechanism of service selection.

4.4.1 Use Case Modelling

Taking into consideration the design requirements and the assumptions established earlier, this subsection uses a case scenario to demonstrate the usability of the selection model and how it can be evaluated.

Figure 4.1 illustrates the use case diagram for the model envisaged. This model shows the interaction between two main instant actors, the service consumer and the service provider at any point in time. It should, however, be noted that all devices within the AMC could serve as both consumer and provider.

4.4.1.1 Consumer or Client

The client makes use of the system to register so that the system will be able to uniquely identify each user that logs into the AMC system. The mobile user would be able to make use of the services available on the server as well as provide services for others. The major responsibilities of the user are to supply services to the system and also participate if he needs to make use of a service from other mobile devices. However, the consumer must specify their service preferences based on the template provided for him or her to be able to request a particular service. Failure to comply with this template prevents the client from accessing the required services.

4.4.1.2 Provider

The mobile devices within the AMC make up the provider of Ad-hoc mobile services. Each mobile device participates by populating the system with their services. This is done by uploading the services to the central server mobile device. Even though the main central

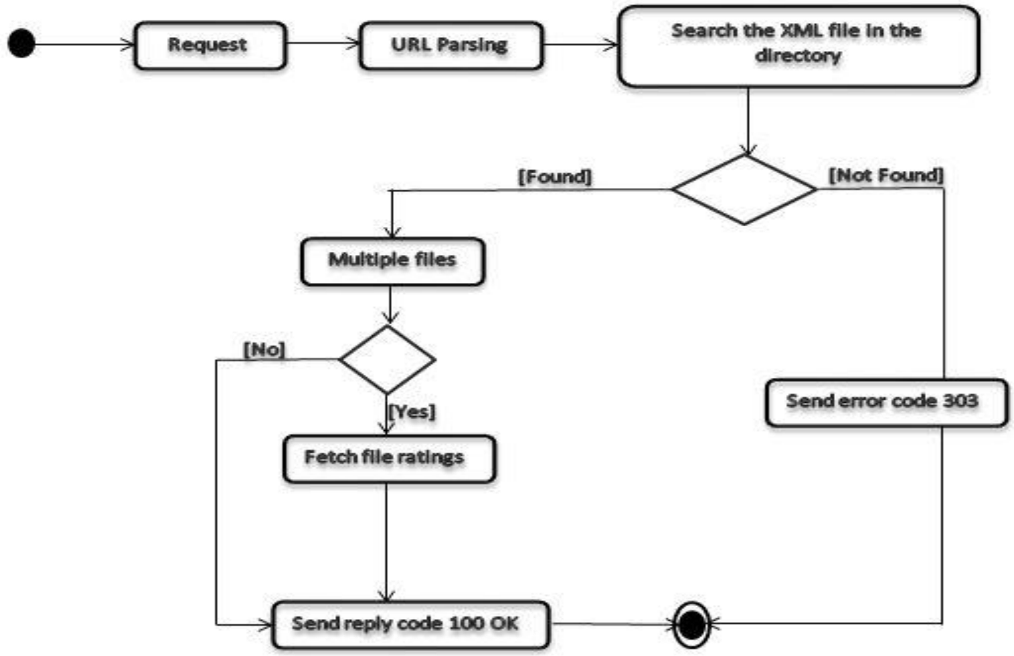

Figure 4.2 Activity Diagram for Ad-hoc Service Selection mechanism

server coordinates the activities of service provision, the distributed mobile devices form the major sources of the service provider. The central server node also provides service uploading templates for publishing services into the system. All the mobile nodes must conform to this template in order to make a service available for others to use. Any device that fails to conform to the QoS templates will not be able to act as a service provider within the Ad-hoc system and, therefore, will only participate by consuming one or more services.

4.4.2 Mechanism Activity Diagram

Figure 4.2 shows the flow of service selection activity in UML within the Ad-hoc system. The diagram describes the flow of operations in moving from service consumer to mobile service provider. When a request is made, it goes through the directory search being controlled by the middleware engine. This generates a computed value of QoS that will determine the match. This match is based on a service whose URL is retrieved based on the value of the service. A test is carried out to determine whether a service match exists or not. The outcome of the test determines which path to take. If “not found” then an error code will be generated otherwise, another test is carried out to determine if the service appears in multiples. When only one service is available then this will be sent to the consumer. The crux of this work lies in the situation where multiple services exist. In that case, a rating operation of those services will have to be carried out. This is done through the rating mechanism that does the recording of each service as observed by users when it was used. The idea of this is to allow the consumer to have the optimal service, as explained in the previous scenario. The detail of the rating operation is given in section 4.5.

4.5 Service Selection Mechanism

The database library in Figure 4.3 keeps the information pertaining to the Web Service quality and provides access to it any time the need arises. The feedback record also provides users information regarding the performance of a Web Service in the course of service usage. The QoS of Web Services plays a pivotal role in ensuring that mobile customers’ select optimal service in the course of issuing requests in the AMC. The service properties define how a particular service will perform in carrying out the specified task (Swarnamugi, 2013). The combination of the effects of the QoS parameters gives the detail about the Non-functional behaviour of the service, which is an important feature, to the service consumers.

This is important because the Non-functional behaviours determine the level of satisfaction that a consumer experiences in the course of using the service. The mobile environment also allows mobile devices to act as both a service user and a service provider. Once a mobile device is registered on the network, they can actively participate in the AMC system.

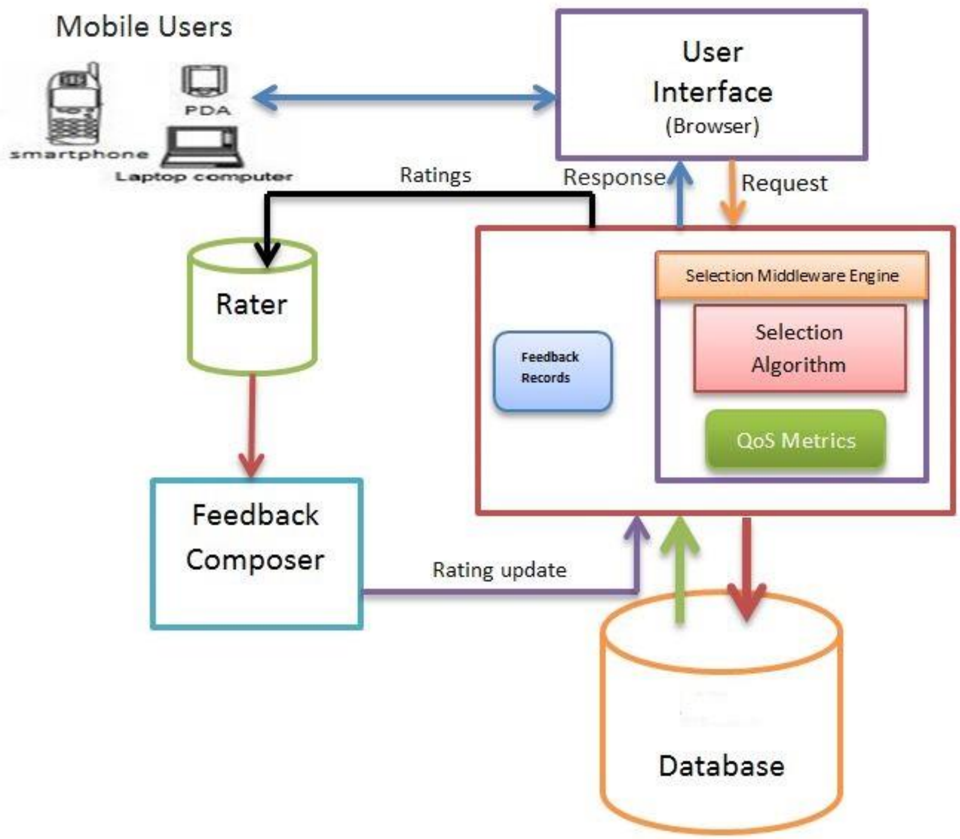

Figure 4.3 Rating Operation in Ad-hoc Mobile Service Selection Mechanism

4.5.1 Mechanism Analysis

Various components are integrated to ensure that the QoS properties are properly collected and processed to yield an optimal solution in the Ad-hoc mobile environment. The detail description of the QoS-based selection mechanism component is given in Figure 4.3. The figure describes all the transactions that take place within the Ad-hoc environment from joining the Ad-hoc cloud to specifying the service needed by the consumer. The service consumer first initiates the process by joining the participants within the AMC. This is carried out by registering as a new user with password and username. The system first implements a protocol through which mobile devices will be able to communicate and indicates its status after connection. It establishes authentication for the intending mobile device wanting to join the AMC.

On authentication, the system allows devices to publish their resources (e.g. audio, video and Web Service) with their QoS properties to the network which remains in the virtual server device. QoS metrics DB stores the Quality of Service parameters submitted by service providers. However, an assumption is made that each mobile node that uploads a service gives the correct values of service QoS parameters being uploaded.

There are monitor agents that help to categorise and monitor various Web Services and data within the main server machine (virtual server device). The QoS aggregate of Web Services is used to select services based on client's prescriptions. It also listens to the activities of each mobile phone, which could be referenced when the need arises.

4.5.1.1 Selection Middleware Engine (SME)

The crux of the work occurs in the selection middleware engine component especially the rating operation. It is connected to the virtual phone farm (main server) in which it resides. This engine connects the consumer to the virtual main server for service provisioning. It contains executable algorithms that aid service selection. The algorithm is based on a dynamic programming approach for Web Service selection within the Ad-hoc mobile environment. This engine also contains a database where the services are stored alongside their values of QoS properties. It allows expandable values of QoS properties to be stored for a proper description of Non-functional properties of Web Services. The management of the user feedback section is controlled by the SME. The SME also consists of a regulatory mechanism that rates service feedback and, as well, ensures the highest ranked services, are selected at every service provisioning.

4.5.1.2 Service Database (SQLite)

This resides on the mobile server PC in the Ad-hoc cloud community. This PC remains invisible to the user in that users cannot view the entire server content individually though they can send a request which can be processed by the virtual device. The SQLite database has been one of the popular databases that acts as a registry (Web Services data storage) during experimental studies. Some of such related data include the URL, XML data expressing service qualities (K. S. Wagh & Thool, 2013). The SME also has a synchroniser that helps in arranging and aligning services in the database.

4.5.1.3 Mobile Users

This represents the various mobile devices participating within the Mobile Cloud community. They are the various smartphones which could send a service request and at the same time provide services within the local cloud. The mobile devices have a browser through which they access the Ad-hoc cloud interface and as well issue a request for a Web Service.

4.5.1.4 Feedback Composer

The feedback composer assists in computing the user responses after each service invocation. There are five categories of feedback levels with Level 5 being the highest among the levels. Thus, a service feedback with a rating of 5 has an excellent record from the service user's perspective. A similar tool has been proven to be effective such that it helps differentiate between bad services and good ones (Y. Wang & Vassileva, 2007). The feedback composer is very important in the selection process especially during the occurrence of selection ties after QoS computations. This study posits that no matter the number of services with similar scores, the users' records will be different based on their view about the services consumed.

4.5.1.5 Rater

The rater collects all the user feedback information and compares it with the highest rating given to each of the services within the system. If a new rating is lower than the rating assigned previously to a service it discards it and keeps count. If a similar or lower rating occurs three times, the rater issues a new rating.

4.6 QoS-Aware Selection Process

The QoS-aware selection process explains what constitutes an optimal service selection based on QoS properties. Service selection is built by mapping relations (requests) onto virtual operations (available Web Services). This mapping process helps in locating the actual Web Service. The aggregation of the QoS for service selection is based on the QoS of the individual service quality parameters. This is carried out in two main steps as discussed below.

4.6.1 The Scaling Step

This study uses a simple scaling or normalisation technique. The concept of scaling is to allow even distribution of QoS properties. The distribution needs normalisation to properly align the various properties of Web Service along appraised positive and negative values for uniform expressions. This step ensures that the parameters are regularised along a similar plane.

This is done by considering the parameter values and observing what happens to that service when those criteria values increase or decrease. For example, when the response time of Web Service increases, it is to the disadvantage of the consumer whereas if the reliability of the same service increases, it is to the benefit of the same consumer. Therefore, the scaling technique represents the response time with 0 and reliability with 1. This representation is used to form the quality vector of service parameters. Therefore, representing:

$\{Ss\}$ = set of Ad-hoc mobile services of the same functionality, also

Let $\{q_{ij}^1, q_{ij}^2, q_{ij}^3, q_{ij}^4, and\ q_{ij}^5\}$ represent the QoS metrics. That is service cost, availability, response time, reliability and compliance. Therefore, for the 5 QoS parameters we have the matrix to be $\{q_{i1}, q_{i2}, q_{i3}, q_{i4}, q_{i5}\}$

This is represented in matrix form as a row by column arrangement with $\{Ss\}$ as the row and $q_{ij}^1 \ldots . q_{ij}^5$ (parameters) as the column.

$$P = P_{ij}\ ; 1 \leq i \leq n\ ; 1 \leq j \leq 5; \qquad (4.1)$$

where each row P_i refers to a mobile Web Service M_n and the column likewise refers to a quality parameter.

The quality vector of the selected QoS parameters is given as follows

$$QV_{(v)} = \{0,1,0,1,1\}$$

in the order of service cost, availability, response time, reliability and compliance.

The next step is to normalise. Several authors, for example Liu et al, (2004) and Makhlughian et al., (2012) proposed approaches to normalisation, but this research is in line with that of Liu and others. The justification for this is based on the fact that mobile devices need an approach that requires less computation time due to the battery consumption challenge.

Thus, applying the formulae from the work of Makhlughian et al., (2012):

$$V_{ij} = \begin{cases} \frac{Q_j^{max} - Q_{i,j}}{Q_j^{max} - Q_j^{min}} & if\ Q_j^{max} - Q_j^{min} \neq 0 \\ 1 & if\ Q_j^{max} - Q_j^{min} = 0 \end{cases} \quad (4.2)$$

$$V_{ij} = \begin{cases} \frac{Q_{i,j} - Q_j^{min}}{Q_j^{max} - Q_j^{min}} & if\ Q_j^{max} - Q_j^{min} \neq 0 \\ 1 & if\ Q_j^{max} - Q_j^{min} = 0 \end{cases} \quad (4.3)$$

where,

V_{ij} signifies the normalized value of QoS property j associated with the candidate service S_j. The value is calculated using the recent value of $Q_{i,j}$, Q_j^{max} and Q_j^{min} which respectively denote the maximum and minimum values of the QoS property j among all the candidate services

$Q_j^{max} = max(Q_{i,j}),\ 1 \leq i \leq n$ likewise $Q_j^{min} = min(Q_{i,j}),\ 1 \leq i \leq n$

and *n* is the number of Web Services being considered at a specific point in time.

Thus, applying the equation 4.2 when $Q_v = 0$

and also the equation 4.3 when $Q_v = 1$

the algorithm below shows the stepwise normalisation of QoS properties and the resulting matrix solution generated from the algorithm.

Algorithm 1: QoS Parameters Normalisation Algorithm.

Input:

A set of qualities of a Web Service **S(t)**= {s_1, s_2, ….s_m} that each describes a service

Output:

A matrix of normalised QoS parameters

Step 0: Initialisation. Create m by 5 matrix P using the formula (4.1)

Thus, having:

$for\ (i = 0; i < m; i++)\ do$

$for\ (j = 0; j < n; j++)\ do$

$if(qf[\mathrm{j}]eq\ 0)$

$if\ (diffqos[\mathrm{j}]! = 0)$

$\mathrm{v[i][j]} \leftarrow ((qmax[\mathrm{j}] - p[\mathrm{i}][\mathrm{j}])/\mathrm{diffqos[j]});$

else

$\mathrm{v[i][j]} \leftarrow 1;$

endif

else if$(qf[\mathrm{j}]\ eq\ 1)$

if $(diffqos[\mathrm{j}]! = 0)$

$\mathrm{v[i][j]} \leftarrow ((p[\mathrm{i}][\mathrm{j}] - qmin[\mathrm{j}])/\mathrm{diffqos[j]});$

else

$\mathrm{v[i][j]} \leftarrow 1$

endif

return

$V = (V_{i,j}; 1 \le i \le n, 1 \le j \le m;)$

endif

endif

From algorithm 1, a new matrix V was generated thus:

$$V = (V_{i,j}; 1 \le i \le n, 1 \le j \le m;)$$

where $Q_v = \{0,1,0,1,1\}$ which represents the quality vector for the normalisation derived.

For example, the output V_2 of a range of Web Service parameters after being normalised using the above algorithm is carried out by assuming that a set of ten mobile Web Services in U_2 for executing a request r_2 and the values for service cost are given by the set q_2 = (3.6, 4.8, 7.3, 7.5, 8.5, 3.8, 2.5, 4.6, 6.5, 7.8). Considering the cost of service as a negative parameter, using the equation (4.2) proposed, the maximum value determined through this process is 8.5 while the minimum value is 2.5.

Hence, the above generated a range of values represented with V_2 below thus:

$$V_2 = (0.82,\ 0.62,\ 0.59,\ 0.17,\ 1,\ 0.65,\ 0,\ 0.75,\ 0.78, 0.66)$$

4.6.2 The Weight Assignment Step

The selection of mobile Web Services that meet the required QoS is made by first giving them the weight that ranges from 1 to 0. This weight represents the degree of importance associated with a specific QoS property expressed as fractions whose sum must be equal to 1. For example, if a consumer's priority is on price for SMS service, then the higher weight is given to price. The service that has the highest score based on price becomes the chosen one, provided it is the only one. The score formula is given in equation 4.4 thus:

$$Aggregate\ Score = \sum_j^n v_{ij} * w_j \qquad (4.4)$$

where w_j is a value ranging from 0 to 1 and the summation of w_j equal 1. The clients or service consumers express their degree of satisfaction by the values given to w_j, as w_j depicts the value of the parameter *j*. When the number of mobile Web Services that have the maximum score is more than one, consideration is given to the rating operation to assist in clarifying the best among the Web Services. The study utilised the dynamic programming approach to solve the formulated optimisation problem and the most ranked of the Web Service is selected.

4.6.3 Collection of User Feedback

The feedback component of the mechanism is to enhance the performance of the selection process so that a no redundancy state is attained in the course of service selection. The effect of this redundancy state is usually felt in a real life situation where systems are populated with Web Services of similar QoS (Sarwar, Karypis, Konstan, & Riedl, 2001; Zhu et al., 2012). It should be noted that the system will not generate services that attain similar aggregate scores unless several Web Services have been uploaded into the system. This would provide an avenue for some proportion of the users to have recorded some feedback into the feedback records. The credibility of each user in the system is assumed to be true and was thoroughly elicited by allowing the individual human users to report their experience about a tested Web Service. This study also assumed a user of a service gives the honest judgment about the credibility of a service based on the behaviour of that service in relation to the claimed service parameters. Thus, five categories of credibility trust on Web Services were assigned namely, the most credible (l_5), more credible (l_4), credible (l_3), less credible (l_2), least credible (l_1). The associated strength to each user response can be normalised in equation 4.5 such that:

$$\sum_{i=1}^{5} l_i = 1 \qquad (4.5)$$

Using this definition, the researcher assigned the value of 0.2 to express the least credibility value l_1 which shows a lack of trust regarding the selected service. Other values starting from 0.4

and 0.6 are assigned to l_2 and l_3 respectively where they both typify a low credibility trust. The last sets of values of 0.8 and 1.0 which represent l_4 and l_5 express reliable trust from the service users. Every rating selection made by the user is computed by finding the average or mean of the feedback values selected. This is used to update the system for the latest rating regarding a particular service. This credibility trust utilised in the proposed system is coupled with a recommendation technique (see section 4.6.4) that will assist to make the right decision for the service user (Sarwar et al., 2001; Zhu et al., 2012). The recommendation technique adopted is a collaborative approach system that is termed the Collaborative Filtering Approach since various users collectively provided the user rating information. The special type of Collaborative filtering adopted in this study is the Item-Based Collaborative approach. This recommendation system is important for two major reasons:

- ❖ It suggests the best of the peer of Web Services among those attaining similar aggregate scores.
- ❖ It also computes final selection based on the services used by the requesting user, provided the user has invoked services in the system before.

These two qualities of this recommendation technique enhance the performance of the selection process and allow no room for the redundancy state during service ties. They also prevent arbitrary selection of any service as proposed by other research works during service selection ties which could likely result in consumer dissatisfaction (Elgazzar et al., 2014b). For example, if there are set of services such as $\{a_1, a_2, a_3, \ldots, a_k\}$ which have similar aggregate scores, the recommendation technique ensures the best out of the services were selected based on the user feedback. This technique uses the supplied feedback from the users to predict the best service based on the assessment that was recorded by previous users (Zheng, Ma, Lyu, & King, 2009).

The item-based collaborative approach uses the prediction creation method to select appropriate service based on feedback records. However, there are online tools that can be used to collect specified feedback of certain online Web Service, for example, Free Spider Simulator tool (FSS). This research makes use of this tool when applying the proposed model to data from a real life scenario.

4.6.4 Prediction Creation

This approach to service recommendation predicts or selects the best service for users based on the previous ratings recorded by the system from earlier service users. The system uses a function that rates the best out of a set of services within a particular domain in a given context such as m-Learning and m-Commerce. The prediction that is proposed to a requesting mobile device is computed via the binomial probability density distribution. In this approach, the following variables were used as expressed below:

Let k be the context of service selection such as m-Learning

If X_k is the user's rating for a service in context k

P_k is a user's preference for a service in context k

The feedback composer rates each service to select the optimal service for users according to the following density for random variables X_k for which x_k is a typical instance:

$$X_k \sim \frac{1}{n} \, Binomial\,(n, P_k), \quad \text{Where n = number of mobile devices (nodes)} \qquad (4.6)$$

$$E(X_k) = P_k, \;\; Var(X_k) = \frac{1}{n} P_k (1 - P_k) \qquad (4.7)$$

This study factors the user rating into the probability prediction for selection. The credibility values were recalled as discussed in equation 4.7

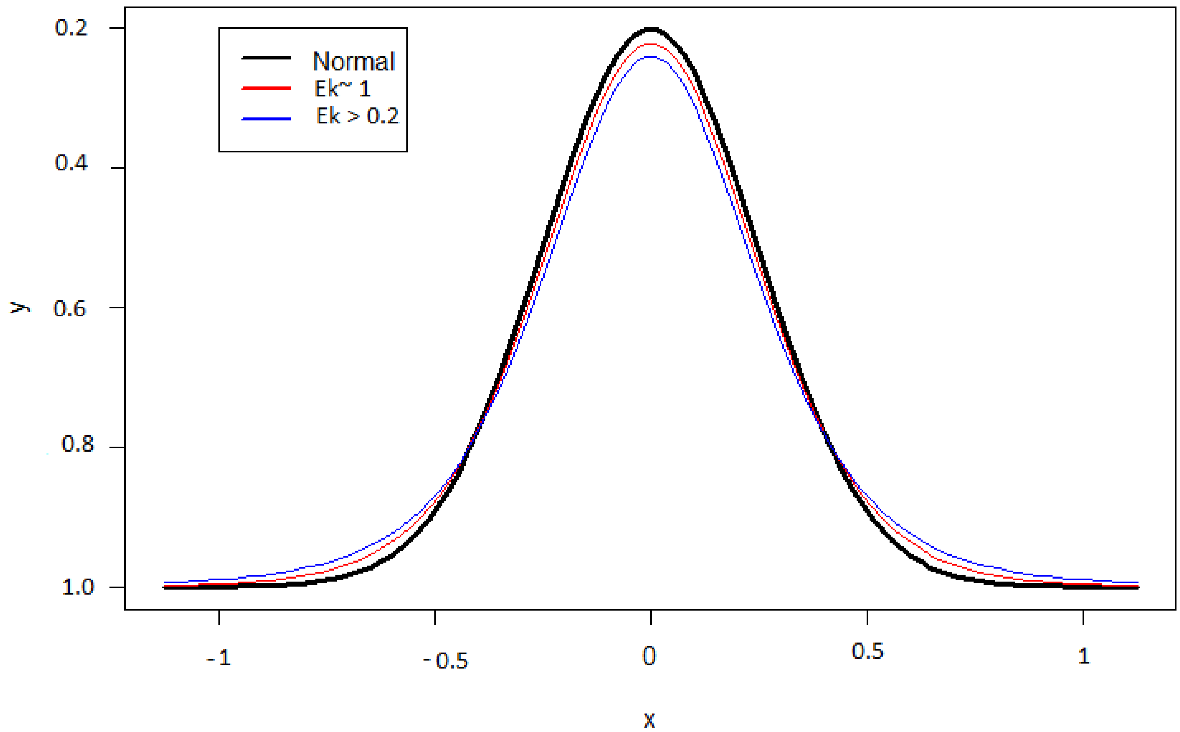

Figure 4.4 The Binomial Distribution of Mobile Web Services for ordering Selection

By expressing the ranges over the binomial distribution, the summary was explained in two ways thus:

- When the feedback is very low, the mean of the rating distribution X_k should correspond to a very low value. For example, if the credibility from the user based on quality experienced is l_2, the feedback composer normalises the probability to assign $E(X_k) = 0.2$
- Contrariwise, when the feedback is high, $E(X_k)$ is expected to be high also. Thus, the feedback composer chooses $E(X_k) = 1$ when the response is high.

This study implements the following variation on the mean X_k ,

thus:

$$E(X_k) = \mu_k = P_k + 2\left(q_k - \frac{1}{2}\right)(q_k - P_k) \qquad (4.8)$$

every service user draws a rating from the mean X_k, for every service that the user rated. These

set of all users' ratings are the inputs to the composer for proper selection to take place. With the calculation carried out by the binomial probability density distribution which tends towards reaching the peak of "1" but may never touch the line of distribution. This is because it continually finds the mean of the ratings submitted by the users which always tends to reach the maximum value of ratings as shown in the binomial distribution diagram in Figure 4.4. The ratings maintain their value if similar values are provided by the new users who makes no difference from the earlier value in the feedback record. But whenever the value of the new ratings is more than the previous record, the rating increases thus the feedback rating falls and rises along the binomial curve that is described in Figure 4.4. The peak of the dome-shaped top in Figure 4.4 corresponds to the least rating from the users and such services are not often predicted to users for use because of the low value or rating typifies that the service performs poorly within the specified service functionality. The flattened edge tending towards 1.0 shows the best services to be predicted to the user for use and the higher it moves from the dome-shaped curve towards the flattened end, the better the Web Services.

4.7 Application of the proposed Selection Mechanism to GUISET

Chapter one of the dissertation introduced the function of GUISET as a middleware technological solution tailored to meeting local needs in rural places, especially the SMMEs, who are faced with a predominant challenge of the cost of IT infrastructures (Adigun et al., 2006). The fundamental significance of the GUISET platform is to provide an enabling infrastructure and also an infrastructure-less service oriented provisioning system that is able to

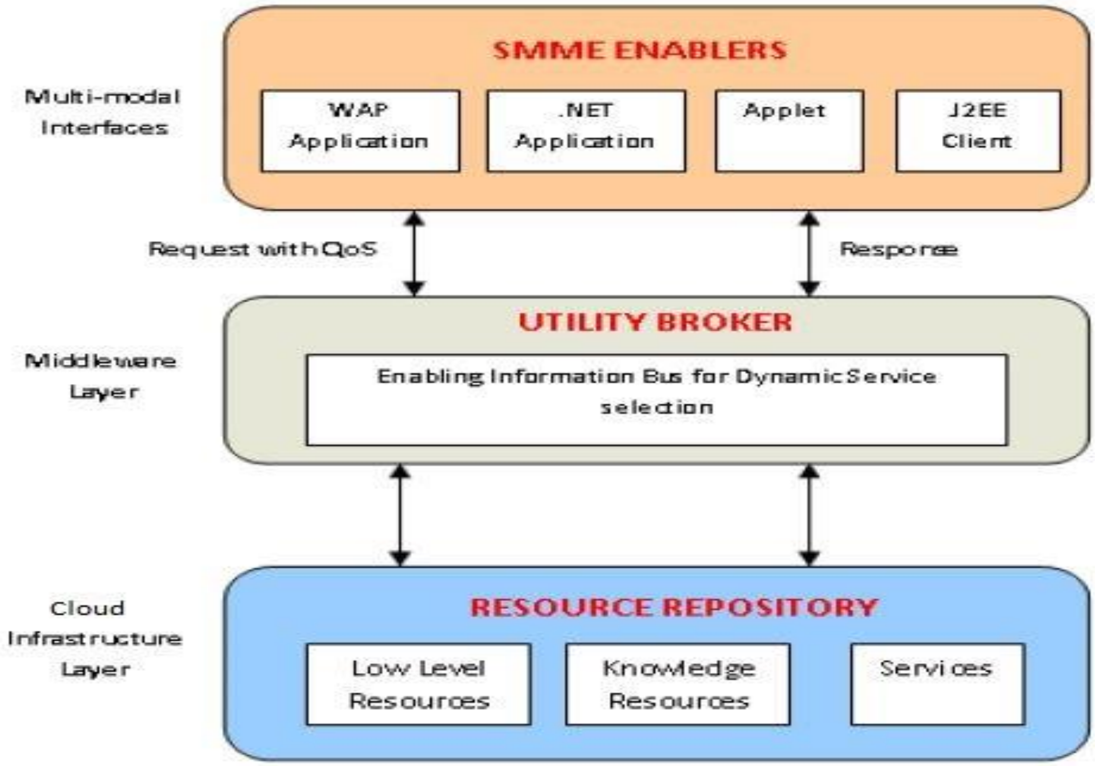

Figure 4.5 The GUISET Architecture **(Adigun et al., 2006).**

integrate both knowledge and resources for sharing and collaboration among individual clients with service providers (Buthelezi et al., 2008). This idea is motivated by the ongoing technological convergence that exists between *Cloud Computing* and *Mobile Cloud Computing*, tending towards creating IT services provisioning as on-demand utilities (Software, Platform, and Infrastructure) through the latest AMC Computing platform thereby merging the benefits of the two earlier mentioned paradigms.

GUISET is a three-tier layer architecture consisting of the cloud infrastructure layer, middleware infrastructure layer, and the Multi-modal interfaces. The Cloud infrastructure layer provides services and resources which are coordinated and managed by the middleware infrastructure layer. The services and resources are routed through the repository of the business services

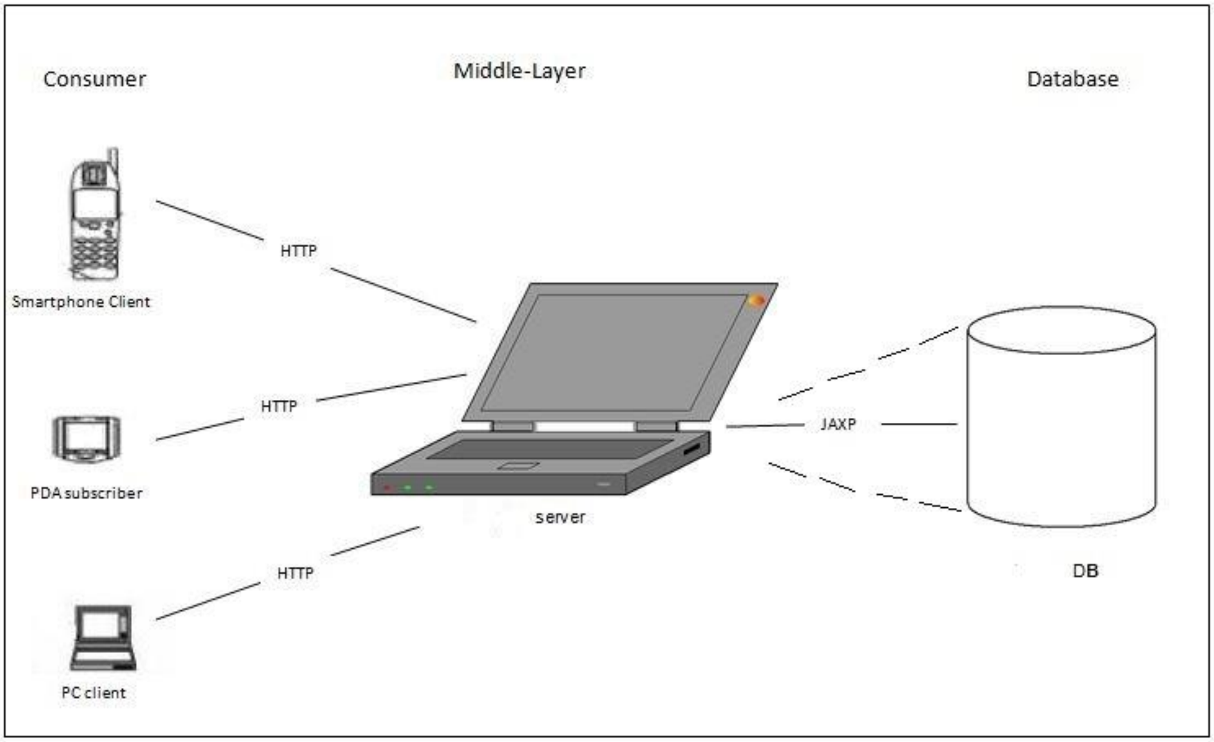

Figure 4.6 Network Model for Ad-hoc Mobile Service Selection

which are usually consumed by on-demand requests via the GUI. The architecture of the GUISET platform is shown in Figure 4.5. The multi-modal interface provides the support for universal access to all kind of clients' devices, user varieties and various execution environments.

The service provisioning AMC environment (infrastructure-less) is an integral evolution from the infrastructure GUISET platform. The cloud infrastructure layer and the middle layer have been fused together and remain inseparable within mobile machines in the AMC. The network model that operates in the multi-modal interface of the GUISET platform is as shown in Figure 4.6 This figure shows a physical connection of mobile nodes to the central server mobile laptop, which houses the uploaded resources.

The individual devices in the Ad-hoc mobile environment have an interface and a wireless connection that link one device to the central mobile server machine (via HTTP). The mobile devices within the Ad-hoc mobile environment offer services to one another within the execution environment being provided by the GUISET middleware (Akinola, Adigun, & Akingbesote, 2015).

Various mobile devices join and leave the GUISET infrastructure-less environment with each one serving as both service requestor and provider. The influx of Web Services available for use motivated the need for Web Service-Defined properties such as Quality of Services which define the behaviour of each Web Service.

The proposed service selection mechanism in section 4.5, therefore, carries out the function of ensuring that optimal service is selected during Web Services selection ties. The mechanism ensures that GUISET users derive an optimal benefit that is available via the middleware platform by accessing best-rated Web Services at run-time.

4.8 Chapter Summary

This chapter has presented a service selection mechanism which aims to enable AMC customers to select and consume their preferred and requested services. The mechanism provides an interactive interface for a collection of QoS parameters that describe various Web Services. The collected parameters are used to ensure that proper descriptions of services within the cloud are made available in the service registry section for easy retrieval of preferred services. In order to achieve the optimal selection goal, a detailed description of the published Web Services is stored and serves as the major concept used in selecting services when a request is issued from the AMC. A typical scenario of how the proposed mechanism will work is given.

The chapter also contains the core explanation of the manipulation of QoS properties from normalisation through to prediction of optimal Web Service based on user feedback. The relevance and importance of the selection mechanism were also outlined in the proper functioning of the middleware platform called GUISET. The selection mechanism serves as a tool to ensure that the infrastructure-less service provisioning platform of GUISET provides optimal satisfaction to subscribing mobile users. The next chapter presents a detailed implementation.

CHAPTER 5

IMPLEMENTATION OF THE PROPOSED WEB SERVICE SELECTION MECHANISM

5.1 Introduction

The previous chapter presented the overall development of the proposed mechanism for AMC service selection. This chapter presents the implementation of the proposed service selection mechanism. It comprises the following sections. Section 5.2 presents the underlying implementation of the simulation setup for the Web Service selection mechanism. Section 5.3 introduces the description of the UML implementations of the proposed mechanism while section 5.4 presents the detailed explanations of the proposed service selection emulator screenshots. The Apache JMeter simulator tool is briefly discussed in section 5.5 as one of the major evaluation tool used while Section 5.6 summarises the chapter.

5.2 Implementation of Service Selection Mechanism

The selection mechanism is implemented as an application that supports message parsing through a simple web form processed by servlets running in a container. That is, messages/requests run via HTTP that linked one mobile emulator to another. Messages are stored as simple XML documents accessed using the JAXP (Java API for XML Processing) and acknowledged signals are sent to clients with the aid of WMABridge API. The WMABridge API allows J2SE (JAVA 2 Standard Edition) application to easily integrate with MIDlets defined by the J2ME specification. The mechanism is divided into three implementation tiers which include the Service Consumer (Client), middle-layer, and the Database tier. The service consumer enables intermingling the J2ME designed emulator issue requests with the aid of Sun Wireless Toolkit 2.5 version for testing. This also enhances the bi-directional flow of information from the clients to the server.

The middle layer runs in a servlet container which is Glassfish Web Server 3.1.2. The web container is a module which handles the processing of web components, managing various aspects of concurrency control and the device status, thus, making the programmer not to be concerned with the integration challenges posed by mobile devices. The database tier is a collection of XML documents which is also referred to as the XML repository. This collection is accessed through the use of JAXP for editing the XML files. DOM Parser is a tool for organising the saved information (XML files) in a taxonomical manner within the database. Figure 5.1 shows the Web Service selection simulation setup and how the three tiers were integrated for evaluation purposes.

5.2.1 Environment Specification

The setup shows the typical connections of mobile emulators within the Ad-hoc mobile environment which helps to simplify the integration of each components of the selection mechanism for evaluation. To test this study's approach, this work deploys the use of the Sun Wireless Toolkit 2.5 Beta Version J2ME from the Sun Microsystems packages to test the behaviour of the selection mechanism. The SQLite database is shown outside of the physical connection setup to depict the three layers of the setup but the DB was actually embedded within the server machine.

The emulator interface design is developed to run on the J2ME enabled devices such that it can be easily deployed on real smartphone mobile devices that support communication through the Hypertext Transfer Protocol networking/interconnections.

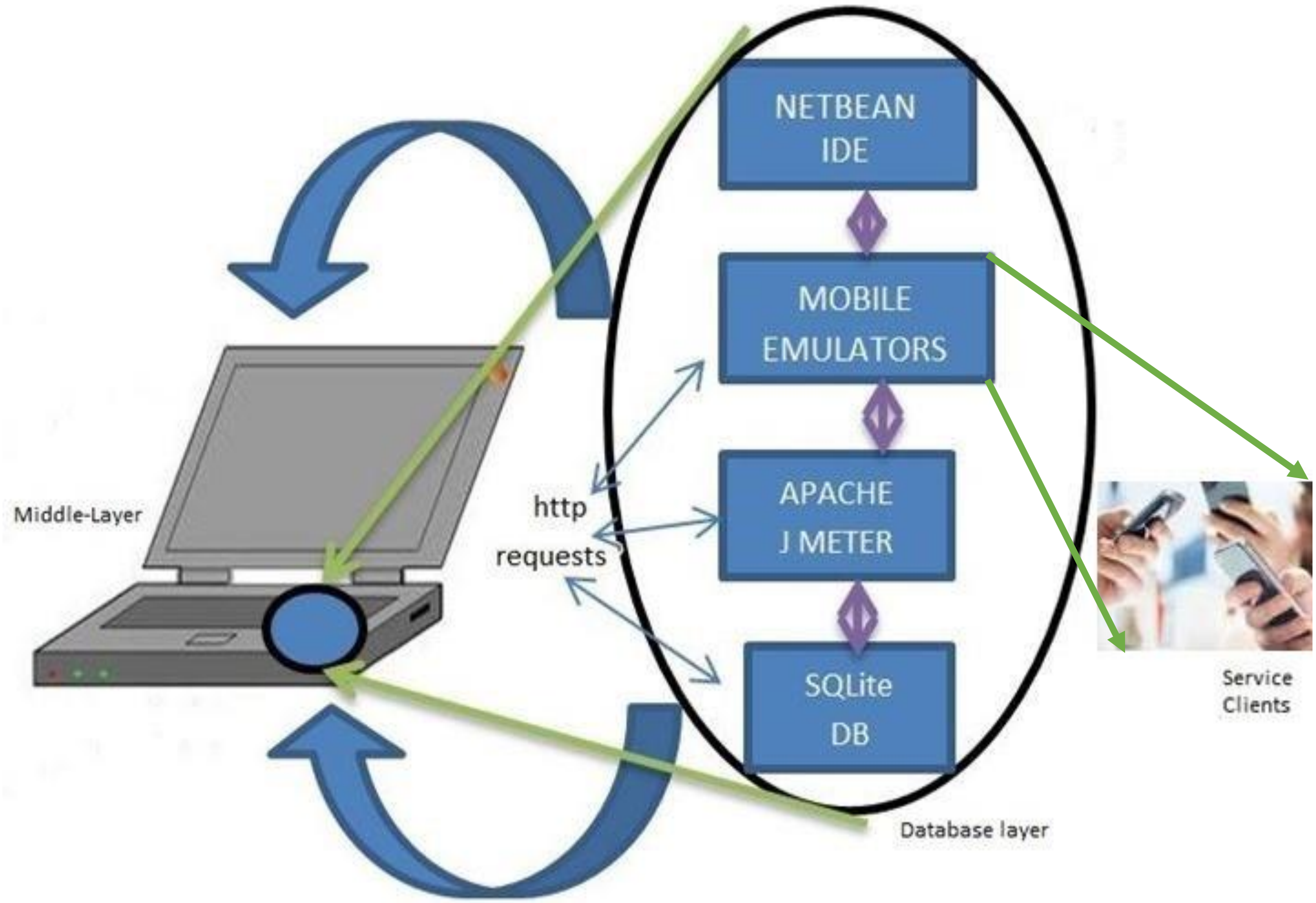

Figure 5.1 Web Service Selection Simulation Setup

The implementation of the server is accomplished by the use of J2EE compliant servers such as Glassfish Application Server 3.1.2. The server-side components are configured to run on any server that conforms to the J2EE Specification. The server handles messages from clients through the use of WMA Bridge API, which enhances message interaction during client service invocation from the server. The server contains the web component which runs within the servlet container and also uses the JAXP to communicate with the external data sources. The external data sources originate as information from the XML database, which is a collection of mobile Web Services.

The whole setup was tested using a mobile laptop running on a Windows 7 Professional Edition as an operating system. The laptop had an Intel Core(TM) *i*7-4500U Processor with a processing speed of 2.40GHz and 8.00GB of RAM. The application consumed 10.4MB of the hard-drive storage which is favourable to the mobile computing environment due to low memory consumption. The simulation setup is depicted in Figure 5.1 which shows an embedded replica of the network model depicted in section 4.7 and Figure 4.6 in Chapter 4. Figure 5.1 contains the mobile emulators developed from the Netbeans IDE together with the Apache Jmeter load generator. The emulator shows the typical interface of the nature of service requests which is similar to requests generated by a typical apache JMeter for the purpose of testing the performance of the proposed selection model. The apache JMeter allows the setting of network parameters like latency and jitters, during the experiments. Therefore, this setup allows for mimicking the network environment of the AMC system.

5.2.2 The Internal Structure of the WSDL file

Several scholars designed impressionistic models for extending the service implementation document of a Web Service to accommodate the description of various Web Service qualities (Cubera, Duftler, Nagy, Mukhi, & Weerawarana, 2002; D'Ambrogio, 2006). A typical Amazon Web Service WSDL document that depicts the specification of Quality of Service criteria (retrieved from http://webservices.amazon.com/AWSECommerceService /AWSECommerceService.wsdl) is as shown in Figure 5.2. The extended WSDL provides many request operations and, additionally complements the features of Web Service qualities as contained in

```xml
<?xml version="1.0" encoding="UTF-8"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://webservices.amazon.com/
AWSECommerceService/2006-02-15" targetNamespace="http://webservices.amazon.com/
AWSECommerceService/2006-02-15">
  <message name="ItemSearchRequestMsg">
  </message>
  <message name="ItemLookupRequestMsg">
  </message>
  <portType name="AWSECommerceServicePortType">
        <operation name="ItemSearch">
         <input message="tns:ItemSearchRequestMsg"/>
        </operation>
        <operation name="ItemLookup">
         <input message="tns:ItemLookupRequestMsg"/>
        </operation>
  </portType>
  ......

  <service name="AWSECommerceService">
        <port name="AWSECommerceServicePort" binding="tns:AWSECommerceServiceBinding">
         <soap:address location=" http://soap.amazon.com/onca/soap?Service=AWSECommerceService"/>
        </port>
        <QoSCriteria>
         <FailureProbability>
          <Availability>
           <qValue  qlevel="High">
            <Min>90  </Min>
            <Max>99  </Max>
            <Preferred> 95  </Preferred>
           </qValue>
           <unit>Percentage  </unit>
           <Weight>0.5 </Weight>
          </Availability>
         </FailureProbability>
         <Trustworthiness>
          <Reputation>
          <qValue  qlevel="High">
           <Min>4  </Min>
           <Max>5  </Max>
           <Preferred>4.5   </Preferred>
          </qValue>
          <unit>None  </unit>
          <Weight> 0.3 </Weight>
         </Reputation>
         </Trustworthiness>
         <Cost>
          <ServicePrice>
          <qValue  qlevel="Medium">
           <Min>30  </Min>
           <Max>60  </Max>
           <Preferred>40   </Preferred>
          </qValue>
          <unit>Pound  </unit>
          <Weight> 0.2 </Weight>
          </ServicePrice>
         </Cost>
        </QoSCriteria>
  </service>
</definitions>
```

Figure 5.2 Amazon Web Service's WSDL extension for QoS parameters (Eleyan & Zhao, 2010).

the <service> element in the service implementation definition section. This study uses a similar format to inform the Web Services which were uploaded in the AMC interface. Four categories of selected Quality of Services were used to minimise the number of data to be computed and various mobile devices that upload Web Services uploaded the actual parameters contained in

```xml
<?xml version="1.0" encoding="UTF"?>
<definitions xmlns="http://schemas.xmlsoap.org/wsdl/" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://webservice.unizulu.com/
AWSEMathService/2016-03-20" tartgetNamespace="http://webservice.unizulu.com/
AWSEMathService/2016-03-20">
  <message name="ItemSearchRequestMsg">
  </message>
  <message name="ItemLookupRequestMsg">
  </message>
  <portType name="AWSEMathServicePortType">
                  <operation name="ItemSearch">
                  <input message="tns:ItemSearchRequestMsg"/>
                  </operation>
                 <operation name="ItemLookup">
                  <input message="tns:ItemLookupRequestMsg"/>
  </operation>
</portType >
………
  <service name="mathservice" GET="yes"> <description> Returns mathservice data </description>
                <filepath>/dc8/mathservice/cgi
                </filepath>
                <parameter name="north_lat" type="latitude"/>
                <parameter name="south_lat" type="latitude"/>
                <parameter name="east_lon" type="longitude"/>
                <parameter name="west_lon" type="longitude"/>
               </service>
                <parameter_type name="latitude"> <regex>^(\d{1,2}\.\d{0,5})$</regex>
              <match num='1' type='float' max='90.0' min='90.0'/> </regex>
               </parameter_type>
              <parameter_type name="longitude"> <significant figures>^(\d{1,3}\.\d{0,5})$</regex>
   <match num='1' type='float' max='180.0' min='-180.0'/>
    </regex> </parameter_type>
    </portType> …...
     <service name="mathService">
    <port name="mathServicePort"binding="tns:mathServiceBinding"><soap:address
location="http://soap.uzulu.com/onca/soap?Service=mathService"/> </port>
            <QoSParameter>
              <Availability>
             <qValue qlevel="High">
             <Min>90 </Min>
             <Max>99 </Max>
            <Preferred> 95 </Preferred>
            </qValue>
           <unit>Percentage </unit>
     <Weight>0.4 </Weight>
     </Availability>
    <throughput>
    <qValue qlevel="High">
     <Min>4 </Min>
     <Max>5 </Max>
     <Preferred>5.5 </Preferred>
      </qValue>
     <unit>None </unit>
      <Weight> 0.4 </Weight>
     </throughput>
     <serviceCost>
     <ServicePrice>
     <qValue qlevel="Medium">
     <Min>20 </Min>
     <Max>40 </Max>
     <Preferred>30 </Preferred>
     </qValue>
     <unit>Rands </unit>
      <Weight>0.2 </Weight>
     </ServicePrice>
     </serviceCost>
     </QoSParameter>
```

Figure 5.3: Ad-hoc Mobile Cloud WSDL file QoS Expression

the WSDL files. The Ad-hoc mobile interface has been designed in such a way that individual users are to provide the quality of such services they want to be made available in the Mobile Cloud as specified in the WSDL document. This research assumed that individual users could read the value of such parameters as it is provided in the WSDL files.

This study developed the AMC Web Service WSDL files through adopting the strategy of extending the Web Service parameters as described by Amazon. Figure 5.3 shows a WSDL file for a typical Web Service called *math_service* which carries out the arithmetical operation and provides the answer in a specified significant figure.

5.3 UML Implementation

This section presents the modelled implementation of the service selection mechanism in the form of Unified Modeling Language (UML), the modelling language used for expressing software implementation design. This design uses the sequence diagram and class diagram to illustrate the process flow of the mechanism for Web Service selection.

5.3.1 Mechanism Sequence Diagram

Taking into consideration the design requirements and the assumptions established earlier in chapter 4, a sequence diagram was used to demonstrate the usability of the selection model. Figure 5.4 shows the sequence diagram for the selection mechanism.

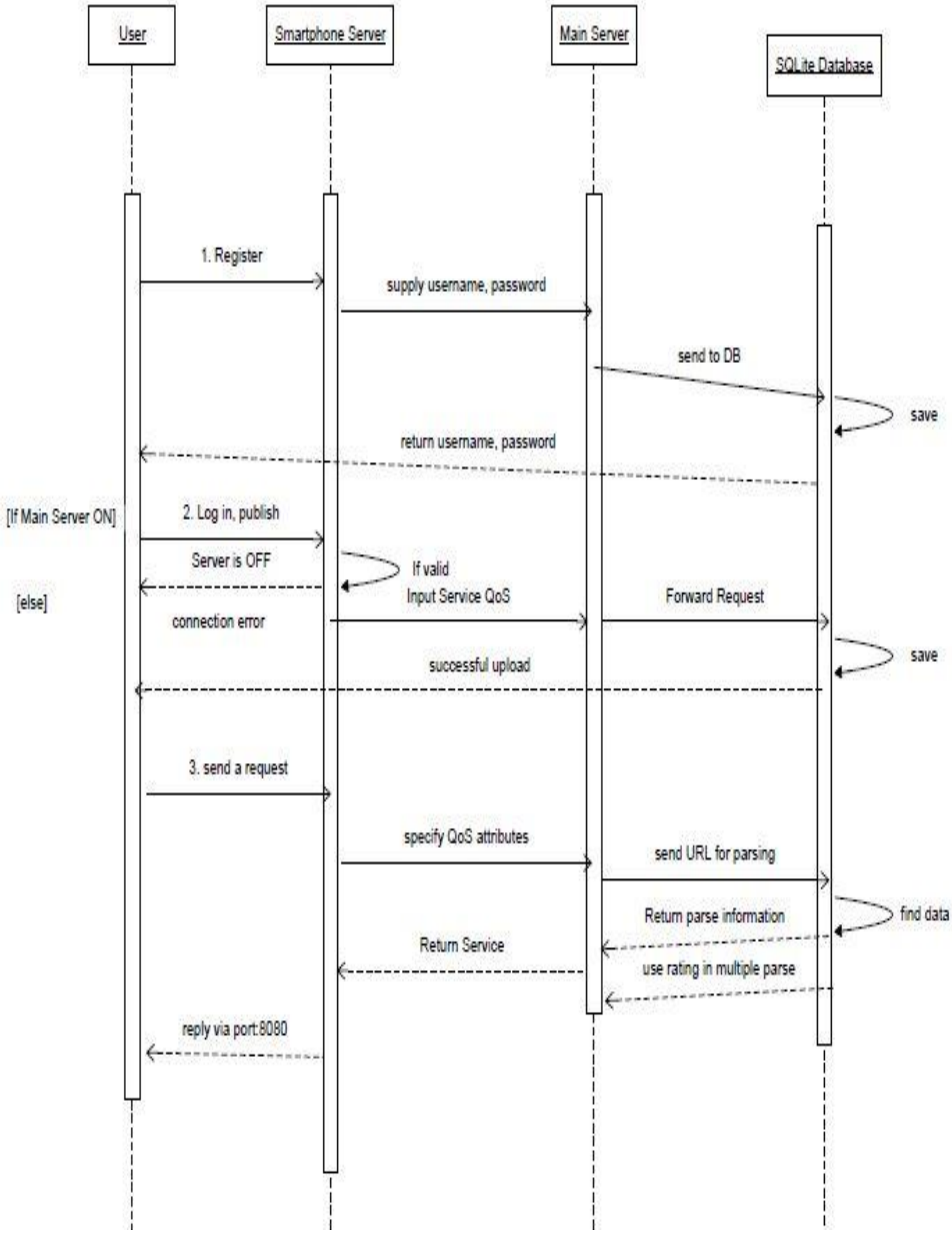

Figure 5.4 Mechanism Sequence Diagram.

5.3.2 Class Diagram

`

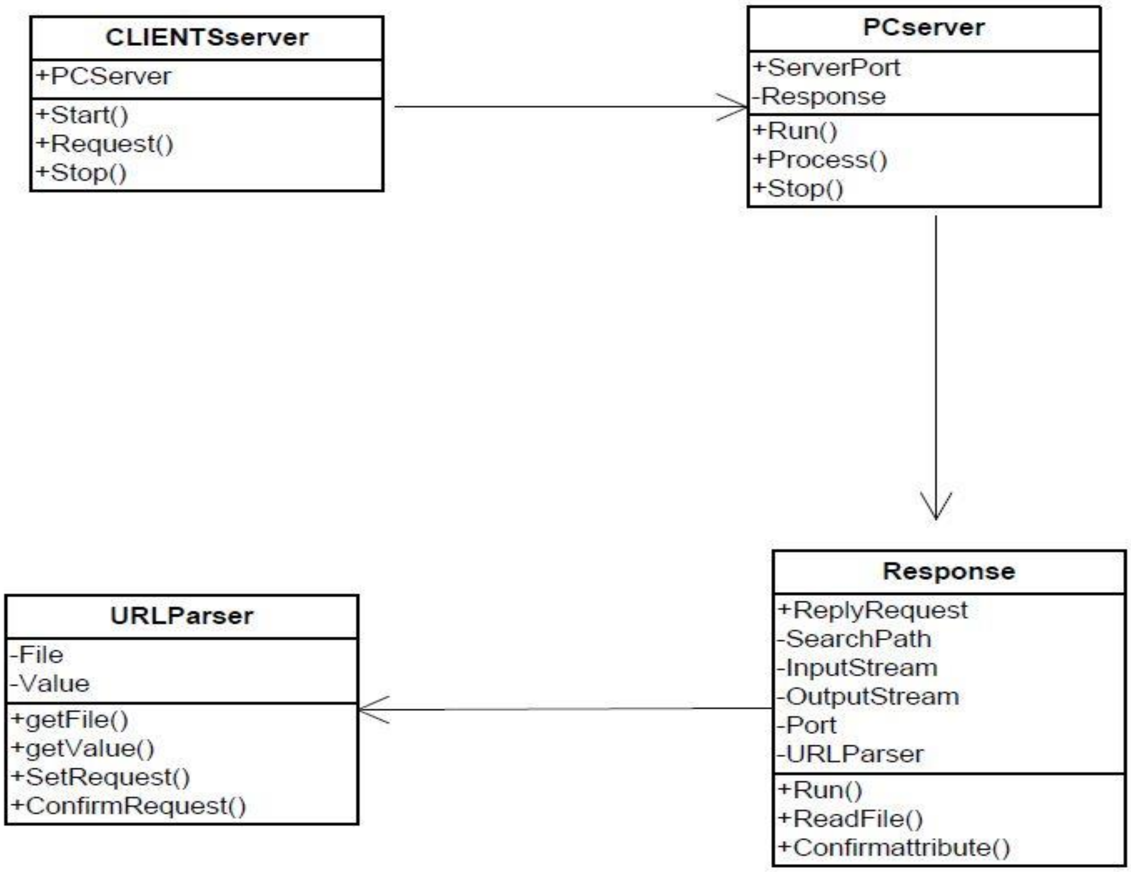

Figure 5.5 Class Diagram

The user sends a service request in the form of an HTTP request which is passed via the laptop that serves as a server machine. The server laptop searches through the database for a match to the request received, checking through the directory search and using the document ratings to select the one that is most appropriate for the user.

Figure 5.5 presents the class diagram for the mobile user or client and the Ad-hoc service provider. The class diagram describes the attributes of the system and provides the operations of a class. The operation specifies the various instructions connecting user and provider interactions within the Ad-hoc mobile community. It also shows a collection of classes, interfaces, associations and expresses the logical constraints that inform some vital decisions in the system. Confirmation of the request is also carried out to see if the service provided actually matches the request.

5.4 Implementation Screenshots

5.4.1 Emulator Interface Screenshots

This section presents the interfaces used to accomplish the service requests and responses within the simulated AMC environment setup. These interfaces were developed on the IDE mobile emulator that is deployed on smartphones mobile devices. This step tests the performance of the selection process on the simulation setup. These interfaces provide the avenue to request and respond to service invocation within the AMC.

5.4.1.1 Registering and uploading of Web Services

Figure 5.6 shows a mobile portal interface for connecting mobile devices to the Ad-hoc environment. This interface allows the first time Ad-hoc mobile devices to subscribe to Ad-hoc mobile services by registering to create an account with a password and username. The username and password are used to authenticate the clients (mobile users) before they can either issue a service request or serve as a service provider. Each mobile emulator sends messages via HTTP through the server personal computer which set up the wireless connection. The wireless connection allows a limitless number of devices to have access to the Mobile Cloud irrespective of their location within the Mobile Cloud environment.

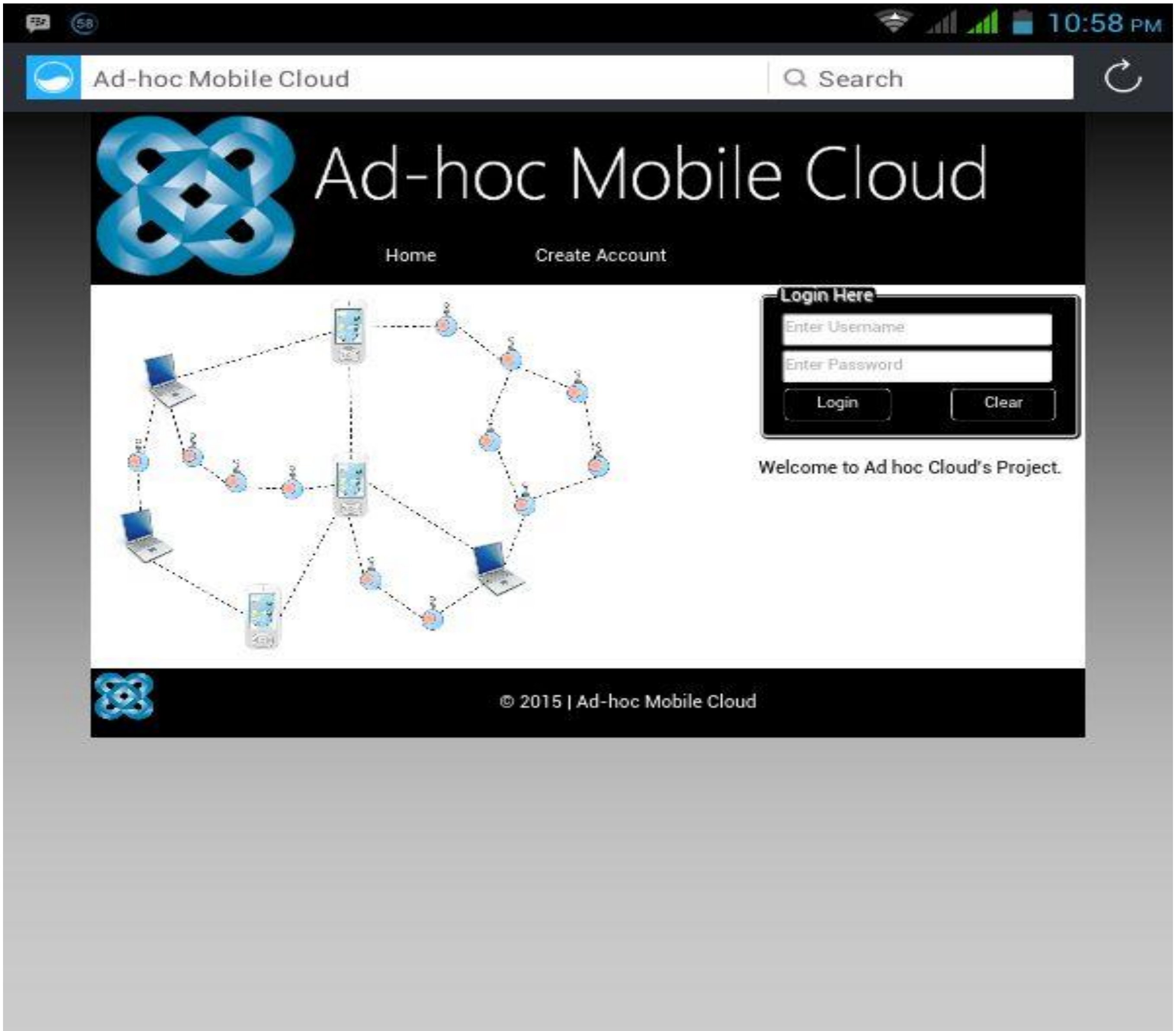

Figure 5.6 User Login Page

The login page takes the user to the AMC homepage where the user can select the option to either make a request for Web Service or carry out the intention to upload a new service for the use of other connected devices. This page accepts the distinct and unique username that was created by the user, along with the password to acknowledge the device as a member of the community.

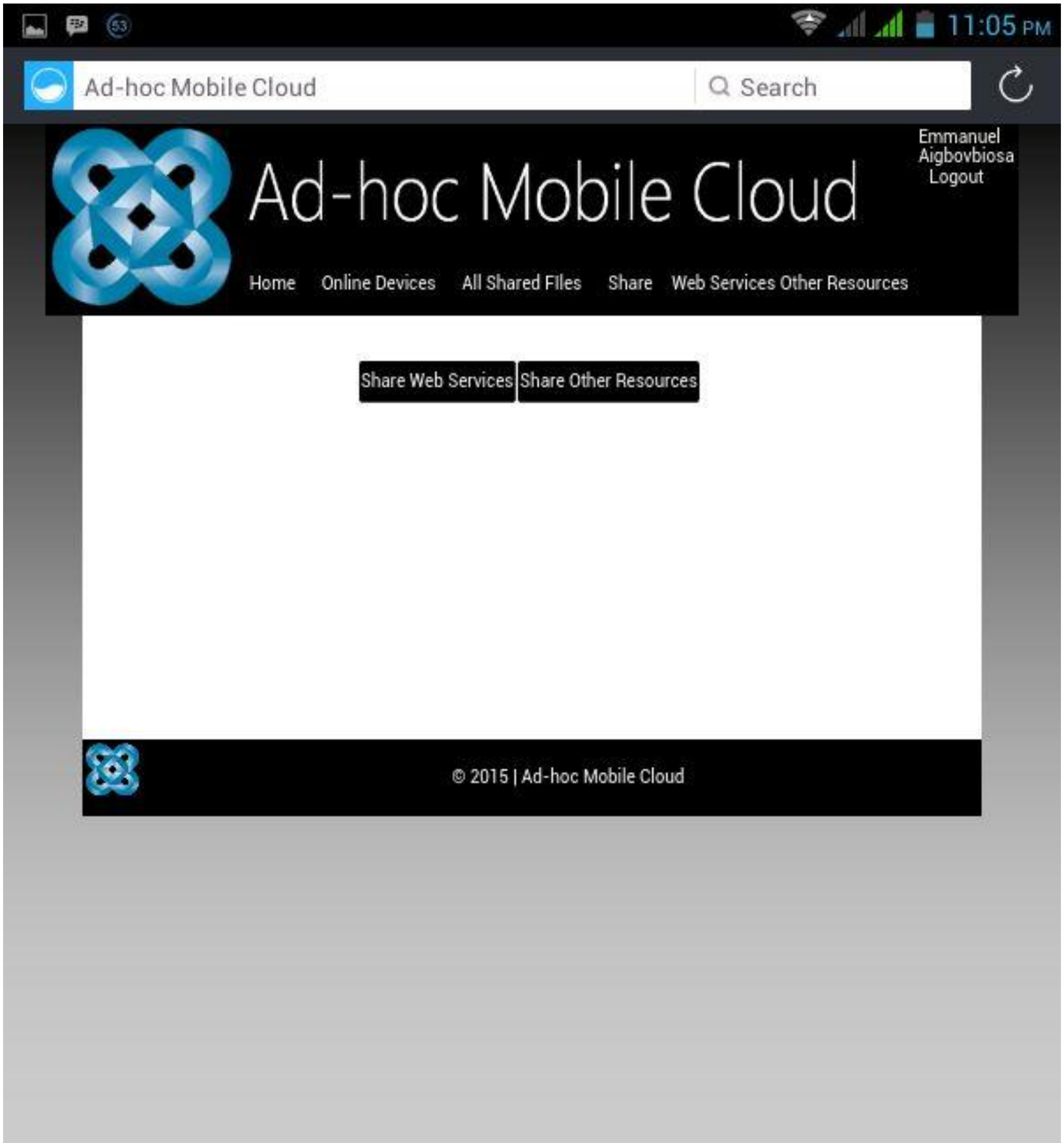

Figure 5.7: An Enabling interface for service uploading and sharing.

Figure 5.7 shows the platform for making an upload either for a Web Service or for a file service. The user specifies the qualities of the service to be uploaded to the system before it can be made available for selection in the Ad-hoc mobile system. The upload will not be completed unless the

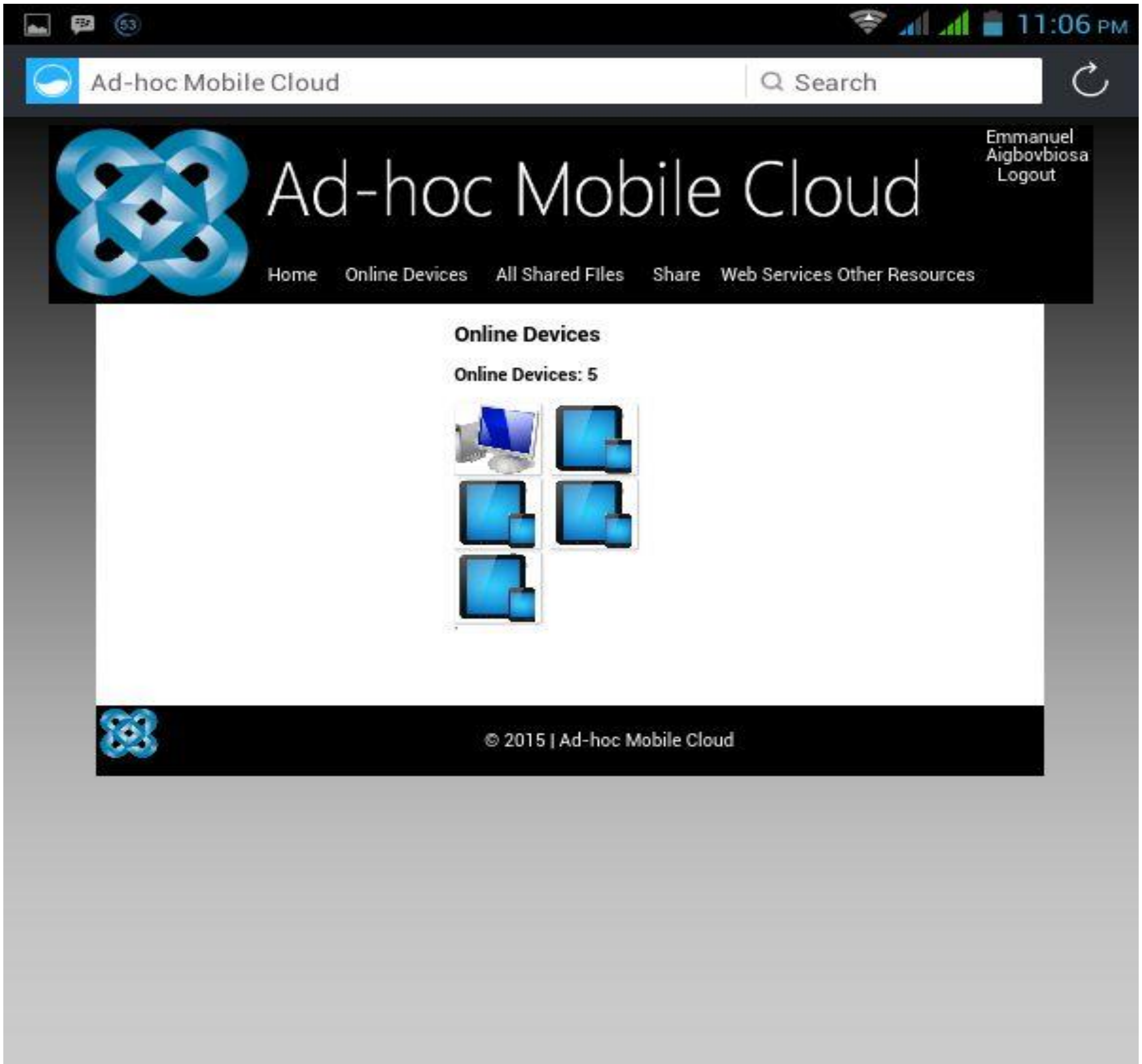

Figure 5.8 Connected devices in the Ad-hoc Mobile Cloud

qualities of such services are clearly specified. The user is added to the number of devices that are connected to the cloud and this increases the number of available devices in the Ad-hoc cloud for service provisioning.

5.4.1.2 Connecting to the Ad-hoc Mobile Cloud

Figure 5.8 shows the number of devices that are already connected in the *Ad-hoc Mobile Cloud*. The wireless set up from the server PC serves to interconnect all the devices in such a manner that they form a Mobile Cloud community. The more interconnected devices in the Mobile Cloud, the more possibilities of different varieties of Web Services within the system. However, this increase in the number of mobile devices makes Web Service selection a tasking challenge within the AMC environment due to numerous competing Web Services.

5.4.1.3 Web Service Query Interface

Figure 5.9 shows the user query interface. This interface allows the mobile user to specify the desired service from the Mobile Cloud community. The user accesses this interface as soon as the authentication is verified, thus giving it the permission to issue a service request. The parameters that are specified are the response time, service cost, throughput and service reliability or availability. The qualities can also be increased depending on the nature of the environment whose requirement needs to be met. The issued request from this platform goes to the server node machine within the cloud which then issues a response based on the available resources within the Mobile Cloud environment. In a case where there is no Web Service for the request, an error message is displayed which will be interpreted as failed request.

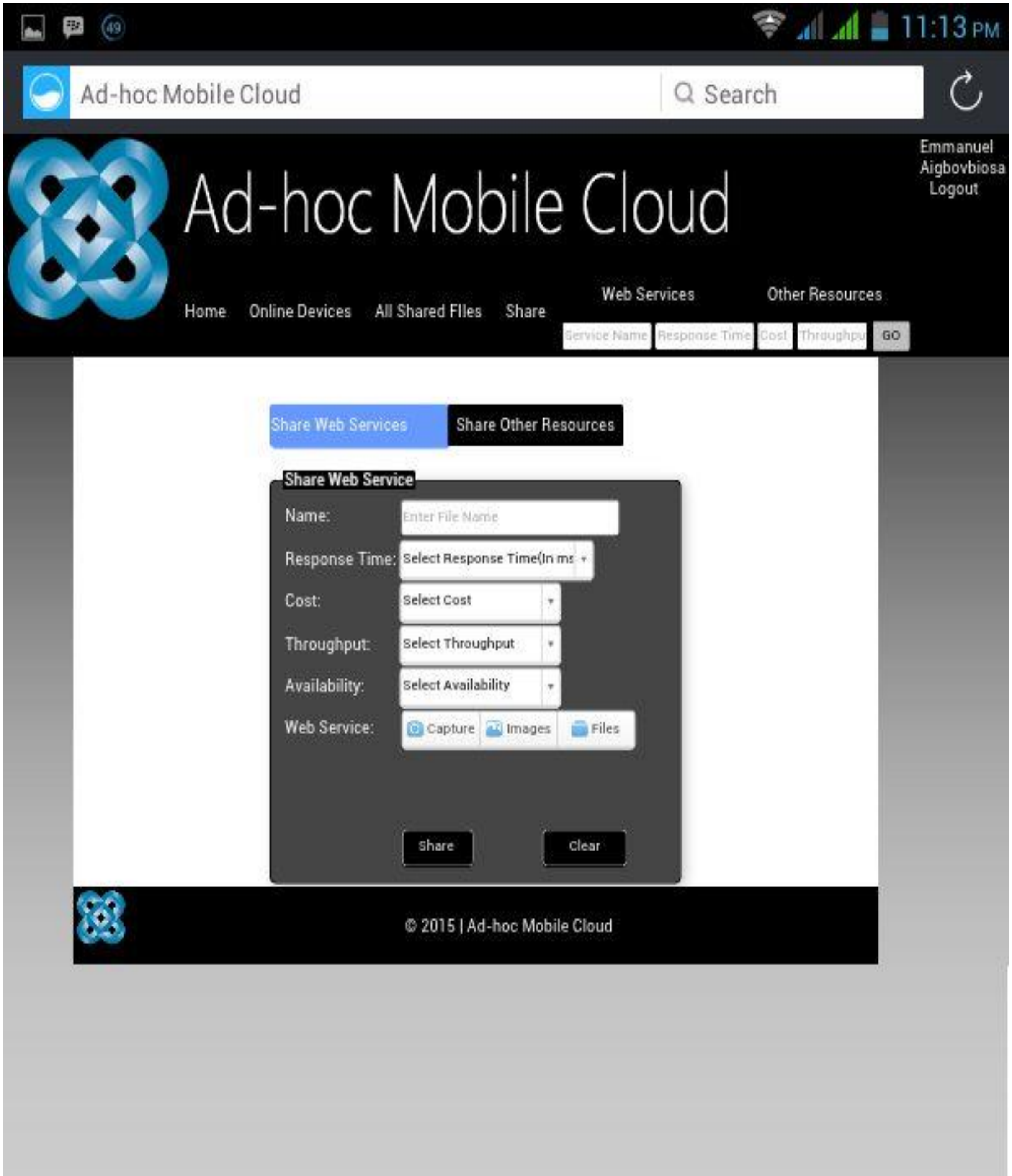

Figure 5.9 Service Specification and Query Interface

5.5 Apache Jmeter

This is a Java application tool developed for load testing. This tool was originally developed for testing web application performances but its expansion over the years has enabled it to be used for simulating a heavy load of requests to Web Services. These requests allow for testing the performance of the system or mechanism in question in terms of related parameters like throughput, and response time (Nevedrov, 2006). Being a very versatile tool, it can also perform mobile testing. It provides some portable features to prepare and run mobile specific performance test scripts. When the proxy is properly in tune with settings on the mobile devices, such performance metrics are easy to record during an experiment. Moreover, the JMeter works well on all versions of the operating system in that it runs a Java Virtual Machine. Figure 5.10 shows the interface of the Jmeter tool running on the server PC of the experimental set up for the mechanism discussed in this chapter.

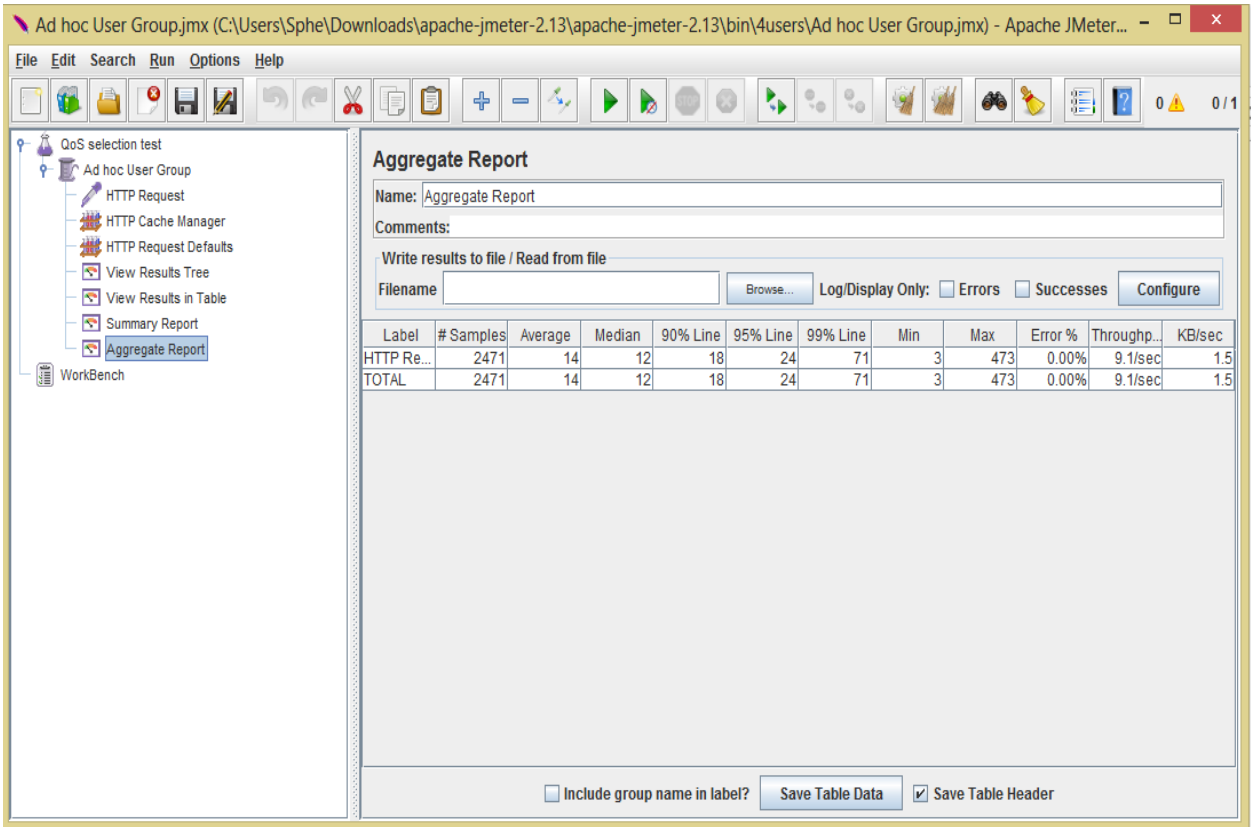

Figure 5.10: Apache Jmeter's load testing interface.

5.6 Chapter Summary

This chapter discussed the implementation details of the Ad-hoc mobile Web Service selection mechanism. The implementation design of the proposed mechanism that explains the sequence diagram alongside with the class diagram is discussed. The shot screen for the implementation as it was designed on the emulator being used for the experiments was also depicted in the figures. In addition, the chapter explained the Apache Jmeter software. This is a load generation simulator that mimics the sending of request like a mobile emulator for the purpose of testing a system. This simulator allows statistical regulation of network parameters and resources to be used for testing the effectiveness of any proposed system. This study regulated some network parameters to give room for their occurrence in real-life situation so as to test how the proposed mechanism behaves. The next chapter deals with the feasibility and performance of the proposed model.

CHAPTER 6

PERFORMANCE EVALUATION OF WEB SERVICE SELECTION MECHANISM

6.1 Introduction

The performance of the proposed Web Service selection is described in this chapter. The effectiveness of the selection method is examined with the test on response time of the selection mechanism, the service throughput (the number of successfully delivered requests) as well as the Web Service availability. The concluding aspect of the performance evaluation is to evaluate the effect of the user feedback system within the proposed selection mechanism. This evaluation also deploys data from other studies in order to access and evaluate how this model performs in solving selection ties during QoS computation.

Furthermore, in section 6.2, this chapter presents the various performance metrics tests that were carried out on the proposed selection mechanism. The results of these various tests are discussed in detail in section 6.3 while section 6.4 concludes the chapter.

6.2 Performance Evaluation

This study based the evaluation on quantitative assessment using certain service performance parameters, namely, execution time, throughput and service availability. An experiment was conducted to investigate the performance of the mechanism through the selected QoS properties for service selection. This was carried out with respect to the increments in the number of service providers and service requests. The researcher in this study also carried out an experiment to verify the effect of user feedback on user satisfaction thereby determining the percentage of user satisfaction. As far as this study is concerned, there are yet no solutions to service selection ties in AMC. Other approaches implemented in cloud computing to solve this challenge includes Web service Ranking Function approach (WsRF) as well as ordinary non- feedback selection (Al-masri & Mahmoud, 2007; Singh & Pattanaik, 2013). This approaches were first implemented in AMC and compared with our approach in this write to see how our approach performs. This is also used to affirm the fact that the use of extensive QoS properties will actually result in optimal service selection as required within AMC computing paradigm.

6.2.1 Execution Time Performance

This evaluation first conducted an experiment on the QoS selection mechanism execution time. A mobile device is not expected to take several minutes to receive a response from the intending provider of Web Service. "Execution time" refers to the time it takes for an application to return the result of a request to the client node (requestor) based on the operation of the service selection mechanism. The basic goal of this experiment is to analyse the response of the proposed selection mechanism as the number of requests increases. The average execution time calculation is based on the apache JMeter framework. This is given by the expression in equation 6.1

$$1/k \sum_{c=0}^{k} \left({}_{c}^{k} E - S \right) c \qquad (6.1)$$

where c represents the requested index, *E* represents the request received time and *S* represents the request sent time. The execution time of a service invocation is a pivotal issue because it determines how pleased the user is with the performance of the proposed system and has a significant relation to the level of user satisfaction. See Table 6.1 and Figure 6.1 respectively. In table 6.1, it is observed that as the number of mobile devices and service requests increase, there is a gradual increment in service execution time. It might be expected that the execution time of the mechanism would increase with a further increase in the number of mobile devices or nodes in

Table 6-1 Data from Execution Time Experiments

No of Mobile Nodes	No of simultaneous issued service requests	Execution Time (ms)
10	4	0.06
20	8	0.08
30	12	0.14
40	16	0.18
50	20	0.32
60	24	0.56
70	28	0.72

the AMC system. However, the rate of increment in execution time does not increase in the same geometric ratio as the number of the service requests. This is because as the number of mobile nodes increases, there are more mobile Web Services that are available within the reach for Web Service selection. Hence, this enables the selection mechanism to use its feedback system to select a Web Service for the query issued.

This shows that the proposed service selection mechanism is able to handle a relative increase in service request without significant delay. The execution time is a very important quality of any service provisioning platform as users will be intolerable of any excessive time delay when Web Services are being requested. A small delay of 1 second could go a long way to affect a service provisioning system to become less efficient. This is the reason why in the real-life service oriented architecture or platform, it abhors execution time delay at all cost to the least possible so that the users will enjoy optimal satisfaction while consuming Web Services on the platform. In addition, this research also envisaged that in a real life-scenario, different mobile users could disconnect and connect to the system while requesting for a Web Service thus the execution time delay becomes less with reduction in the number of connecting mobile devices as well.

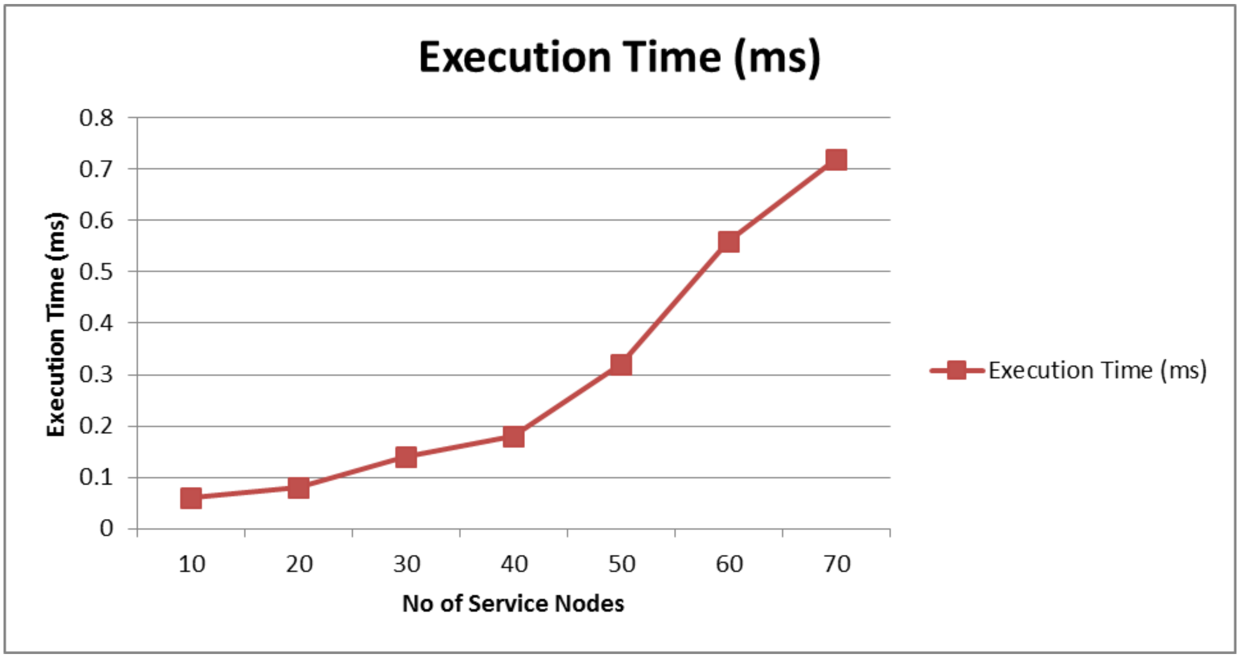

Figure 6.1: Service selection Mechanism average response time

This “in and out” pattern of mobile device movement in the AMC can further instil a relatively stable system with a significant increase in execution time through a session. Moreover, the provisioning of mobile service that was made available from neighbouring devices accounts for a considerable steep execution time experienced. Thus, this could serve as a good tradeoff with respect to service availability within the system.

Figure 6.1 also shows that the QoS could only be relatively stable for a reasonable number of users. When it goes above a certain number it increases steadily, bringing about an increase in execution time. However, the execution time can never be fixed indefinitely as the number of requests for mobile device increases.

6.2.2 Throughput Performance

A similar experiment was carried out on the throughput of the proposed selection mechanism. Throughput measures the rate of successful service request delivery over the AMC

Table 6-2 Data from Throughput Experiments

No of Mobile Nodes	No of simultaneous issued service requests	Throughput values (Tps)
10	4	0.09905806
20	8	0.13596488
30	12	0.19596488
40	16	0.30477120
50	20	0.45226360
60	24	0.55263640
70	28	0.57863640

communication channel. It determines the time it takes to perform a transaction (*E*) over a number of sessions. The request simulating tool called JMeter is used to measure the throughput of the experiment. Since the proposed Ad-hoc mobile system consists of connecting mobile devices, the experiment is vital to access the performance of service selection mechanism. Thus, using the expression

$$1/k \sum_{c=0}^{k} \mathrm{ET} \qquad (6.2)$$

to compute the throughput of the system, where the variable *c* represents the request index and *k* is the number of requests at a particular session. The Table 6-2 shows the results of the throughput for the Load generator.

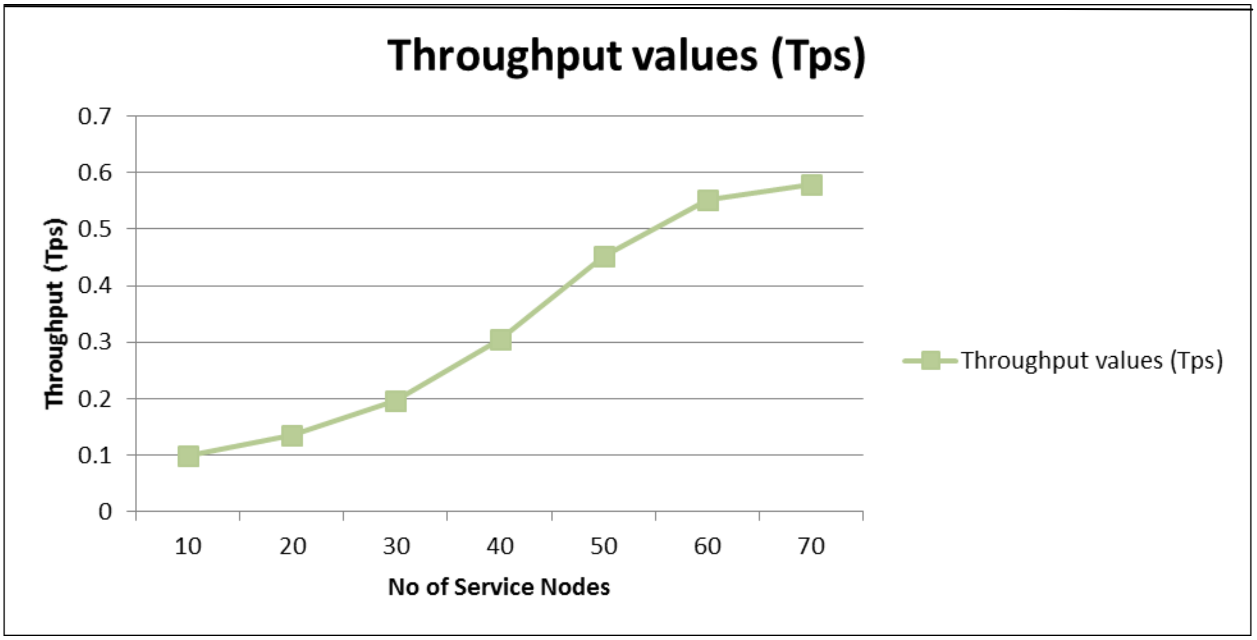

Figure 6.2 Mechanism Selection Throughputs

The number of mobile nodes (users) represented the number of available mobile devices within the system at the time the experiment was conducted while the number of simultaneously issued service requests showed the number of users that were making an HTTP request during the experiment.

Figure 6.2 shows the throughput performance of the Web Service selection mechanism as the number of users increases. The throughput of a system clearly showed the number of completed service deliveries to the requesting service consumer. The proposed selection mechanism performed very well when the test for throughput was conducted. The linear increment in the values of the throughput with increasing number of nodes request performs in a realistic way. The system ensures that an optimal result is provided for any requesting node and the throughput averagely increases with the increasing number of requests, which confirms a good performance from the system.

6.2.3 *Service Availability Performance*

There are many options for quantifying the rate of service availability within interconnected systems. One of the popular options is the use of statistical analyses (Kokash & D'Andrea, 2007; Kondratyeva, Kushik, Cavalli, & Yevtushenko, 2013; Zeng et al., 2004). These approaches are only feasible in a very stable environment where mobility is not occurring. Thus, the constant dynamic nature of the AMC will not favour these approaches. However, this study assigned some weight indicators to certain parameters (fundamental resource availability i.e HTTP authorisation manager, Response assertion, HTTP response default and Hypertext Markup Language (HTML) link parser) in the apache Load generator which is indicated at runtime by assigning values ranging from 0 and 1 for each of the parameters. Based on these parameters, suppose P_{xy} represents the weight of set parameters S_{xy}, where S_{xy} sum up the indicator values I_{xy} , and given that,

$$\sum_{x=1}^{x=a} I_{xy} = 1 \tag{6.3}$$

the service availability of the AMC is therefore calculated using the expression

$$Ava = \sum_{x=1}^{a} \sum_{y=1}^{b} P_{xy} S_{xy} \tag{6.4}$$

Table 6-3 Data from Service Availability Tests

No of mobile Nodes	**No of simultaneous issued service requests**	**Service Availability Rates (%)**
10	4	0.54082
20	8	0.76290
30	12	0.79301
40	16	0.85501
50	20	0.89028
60	24	0.90104
70	28	0.91024

In this experiment, the number of the mobile nodes in the Mobile Cloud is increased, which relatively increases the number of requests and invariably has the corresponding impact on service availability. Table 6-3 shows the computed service availability rate and Figure 6.3 depicts the graph interpretation.

The experiment considers the effect of the selection mode on the rate of service availability to each requesting mobile device within the AMC. As the number of the service provider nodes increases, the service availability climbed steeply at the initial stage but increases more at 20 nodes (i.e as the nodes number doubled). Which means that, other things being equal, the greater the number of service providers, the more the number of services that is available for consumption.

Figure 6.3 Service Selection effect on Web Service Availability

It was noticed that as the number of nodes increases within the Mobile Cloud their rate of service availability was not so significant in increment, due to relatively similar Web Services which also perform similar Non-functional qualities. This can be assumed to be the result of similar recurring service requests which exist within the cloud. The central server PC is stabilised and the feedback records are properly aligned to enhance relatively optimal selection without excessive redundancy of the query.

6.2.4 Evaluation of the Service Selection Feedback Mechanism

This section describes the tests conducted with regards to the effect of using user ratings on Ad-hoc mobile service selection. The proposed mechanism is intended to solve the existence of Web Service ties through the use of user feedback ratings. This proof of concept system is implemented by deploying the Glassfish Web Server 3.1.2 platform on the central server PC within the AMC. This platform works together with the SQLite database which acquires the

WSDL file as well as the QoS information provided by the various mobile service providers within the system. The test deployed a total number of 100 Web Services on the Glassfish Web Server 3.1.2 for this experiment to simulate a real world selection challenge in terms of user dissatisfaction.

To evaluate the Feedback-enabled QoS selection, this experiment evaluated the number of service selection that coincides between feedback-enabled and non-feedback service selection which was adopted from (Al-masri & Mahmoud, 2007; Singh & Pattanaik, 2013). The experiment showed how precise are the two approaches at ensuring the most relevant service is selected. In the first approach, the service was selected directly without the aid of feedback ratings of the Ad-hoc mobile selection mechanism (an approach that is already used in cloud computing), while in the other approach the selection was carried out with the aid of the feedback selection ratings. The same values of QoS parameters are requested with increasing number of users in each approach explained above. This evaluation revealed the kind of services retrieved after each service request and recorded the level of satisfaction ranging from 1-star through to 5-star rating. Starting with the feedback-enabled approach, the expression for percentage precision is given thus:

$$Precision = \frac{|\{relevant\ web\ services\} \cap \{retrived\ web\ services\}|}{|\{retrived\ web\ services\}|}\ \% \quad (6.5)$$

Table 6-4: Data from Web Service selection with and without Feedback

No of mobile Nodes	No of simultaneous issued service requests	Precision-Feedback enabled %	Precision-Non-feedback enabled %
10	4	99.02	85.50
20	8	89.18	80.73
30	12	97.74	84.02
40	16	85.40	85.38
50	20	97.01	73.74
60	24	95.07	80.28
70	28	80.97	65.84

This gives the percentage precision through calculating the mean of responses from the number of users at each level of the service request. This precision is expressed to determine the percentage level of satisfaction derived by each of the service users.

Table 6.4 shows the calculated precision in relation to user satisfaction from feedback-enabled and non-feedback enabled Web Service selection expressed as a percentage.

Table 6-4 was used to generate the graph shown in figure 6.4. The figure shows that the feedback indicates highly satisfactory service selection at every incidence of service invocation when compared with selection based on service quality alone. At some point during selection ties the non-feedback effect can make a selection that coincides with the optimal service at that point and this accounted for both approaches achieving a similar value at experiment number four. In addition, the feedback enabled selection mechanism enhanced the selection of Web Services

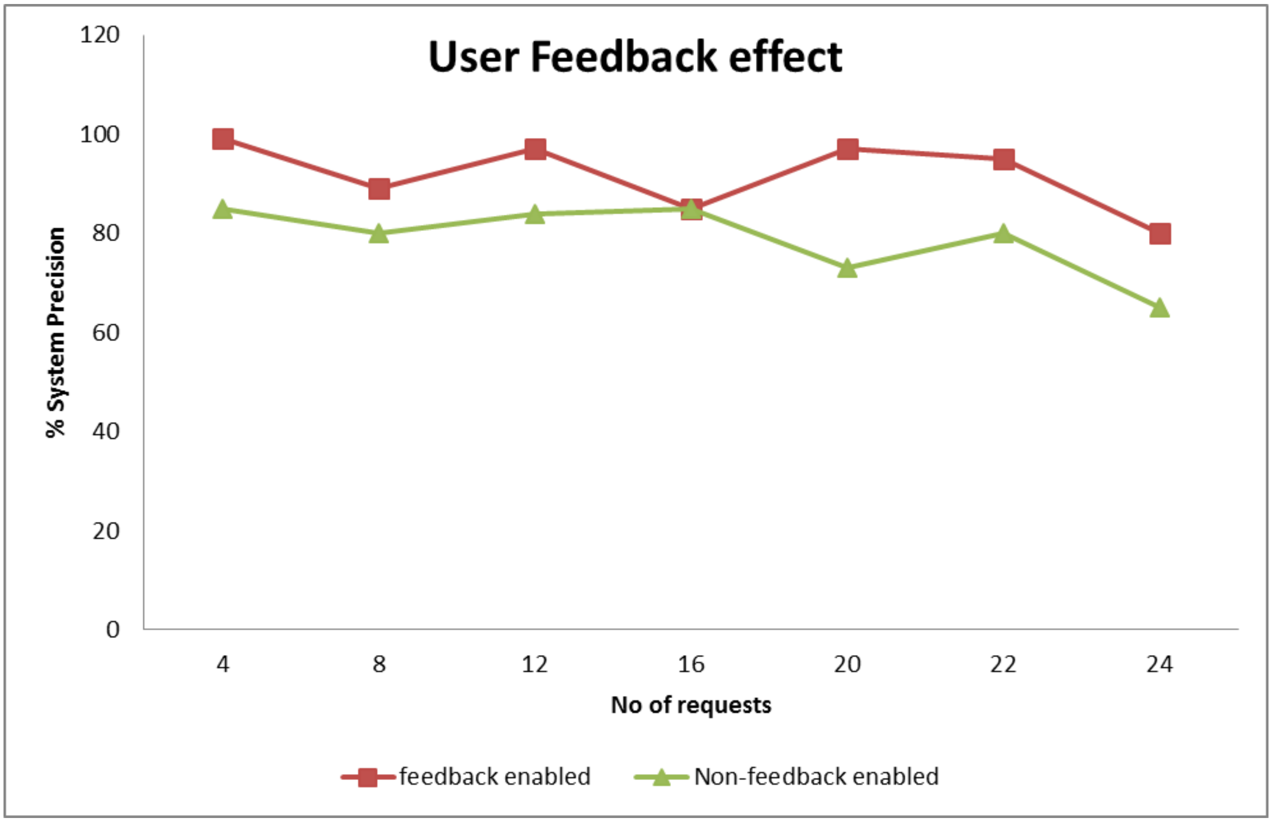

Figure 6.4: The selection precision based on feedback effect

and also gave a high tendency to solve the issue of service selection ties, thereby providing an optimal service for users.

The user feedback effect experiment shows that the use of a continuous updated and unlimited range of users' Web Service assessment only enhance the selection of optimal services for the service users. It should, however, be noted that the greater the number of services that attains the selection ties, the higher the effect of the feedback-enabled system in selecting appropriate and more satisfying services for the user. In a situation where there were few numbers of services, both approaches often, though not always, assumed the same output result. The feedback rating system, however, enhanced the selection of the highly rated service among the selection ties for the user at each point of service invocation.

6.2.5 Application of Proposed Mechanism to a Real-Life Scenario

This experiment used data samples from the work of Al Masri and Mahmoud Al-masri & Mahmoud, (2007). This gives us the Quality of Services of seven different Web Services that were chosen from the same domain. These Web Services are email Web Servics that share the same functionality. The data were provided in StrikeIron.com, XMLLogic, and XMethods.net. A total number of twenty Web Services were used in this experiment to actually see the effect of the proposed selection mechanism at solving the occurrence of Web Service ties, although the seven samples used for the experiments in Al Masri and Mahmoud has the service ranking that tends to be closer together. However, if more Web Services are involved, there is a high tendency for the existence of Web Services with ties. The last column of the table also specifies the feedback of each of the Web Services as derived from the free spider simulator tool (FSS). FSS is an online tool that helps to get a free report about Web Service actionable insights. The feedbacks for over two-weeks are interpreted and presented in the format that aligns with the representation explained in Chapter Five of this dissertation.

This section of the experiment evaluates the effect of the service selection mechanism on a larger number of Web Services (twenty) of similar functionality. The QoS of the Web Services are specified by the data provided in Table 6-5 which shows the QoS values of twenty Web Services as measured by WS-QoSMan. WS-QoSMan is a module that is responsible for collecting and measuring the QoS information of Web Services. The values of the QoS metrics are normalised

Table 6-5: QoS Metrics for Various Available Email Verification Web Services

ID	Web Services	Response Time (ms)	Throughput (bits/s)	Availability (%)	Service Cost (Rands)	Reliability (%)	Feedback (%)
1	XMLLogic	720	6	85	1.2	87	59
2	XWebservices	1100	1.74	81	1	79	49
3	StrikeIron Emails	710	12	98	1	96	86
4	CDYNE	912	11	90	2	91	65
5	Webservicex	910	4	87	0	83	89
6	ServiceObjects	1232	9	99	5	99	94
7	StrikeIron Address	391	10	96	7	94	70
8	Kickbox	428	5	86	8	60	67
9	Byteplant	601	8	70	4	75	100
10	QuickEmail	205	3.5	95	5	89	62
11	TowerData	504	9	75	1	77	79
12	Leadspend	832	8	81	3	83	76
13	Briteverify	911	3	80	5.1	78	26
14	Mailbox	604	10	89	5	87	69
15	Emailanswers	505	7	85	6	95	29
16	BulkEmailVerifier	600	5	90	4	88	96
17	BulkEmailChecker	195	3.5	85	0	89	55
18	Emailtor	220	6	87	8	75	28
19	Verifalia	950	5	90	3	86	38
20	Xverify	350	8	96	4	94	56

and assigned weights accordingly. The results of various Web Services were computed as shown in Table 6-6.

Table 6-6 comprises the Web service Ranking Function (WsRF) approach in cloud computing from (Al-masri & Mahmoud, 2007). It also has Non-feedback enabled outputs typically adopted from (Singh & Pattanaik, 2013), Feedback-enabled outputs and the rankings obtained by various QoS-based Web Service selections methods. The graphical representation of Table 6-6 is shown in Figure 6.5.

Table 6-6: Results of WsRF, Non-FB and FB QoS Metrics Computation.

ID	Web Services	WsRF computation	Rank	Non-Feedback enabled computation	Feedback-enabled computation	Rank
1	XMLLogic	3.6638	13	4.7648	0.68	13
2	XWebservices	3.2166	15	4.3176	0.64	16
3	StrikeIron Emails	4.6103	4	5.7113	0.87	5
4	CDYNE	3.9246	11	5.0256	0.72	11
5	Webservicex	4.2679	5	5.3689	0.89	4
6	ServiceObjects	4.6700	1	5.7705	0.90	3
7	StrikeIron Address	4.1955	7	5.2965	0.79	8
8	Kickbox	4.0428	10	5.1438	0.74	10
9	Byteplant	4.6700	1	5.7704	0.97	1
10	QuickEmail	3.9205	12	5.0215	0.69	12
11	TowerData	4.2504	6	5.3514	0.83	6
12	Leadspend	4.0832	8	5.1842	0.81	7
13	Briteverify	2.0911	20	3.1921	0.58	20
14	Mailbox	4.0604	9	5.1614	0.75	9
15	Emailanswers	3.0505	18	4.1617	0.62	18
16	BulkEmailVerifier	4.6700	1	5.7706	0.92	2
17	BulkEmailChecker	3.2195	16	4.3205	0.65	15
18	Emailtor	2.3020	19	3.4031	0.60	19
19	Verifalia	3.0950	17	4.1961	0.63	17
20	Xverify	3.5077	14	4.6087	0.66	14

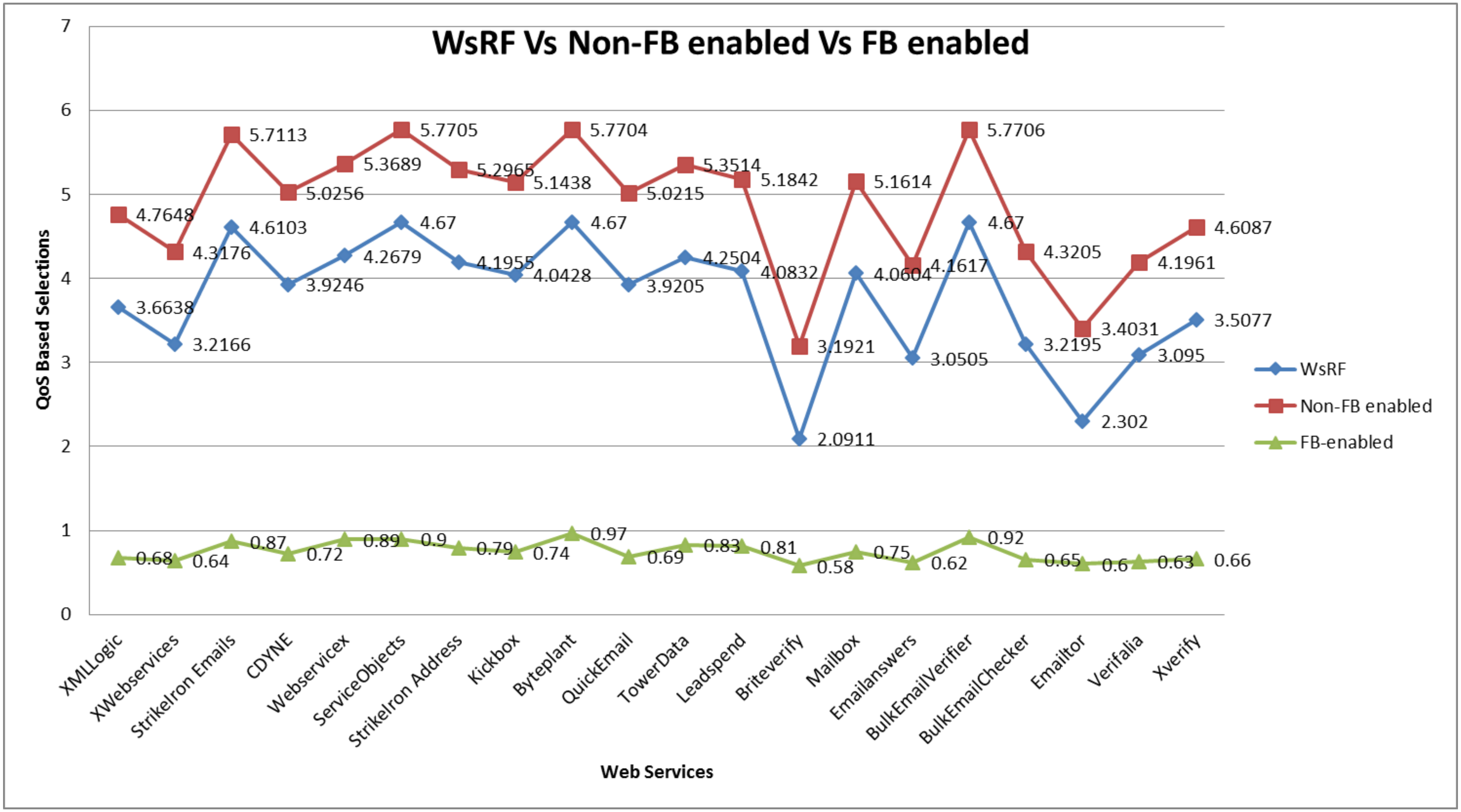

Figure 6.5: The comparison of WsRF, Non-FB enabled and FB enabled Web Service selection

Figure 6.5 depicts twenty Web Services that have similar functionality but possess different QoS. The various QoS properties of the Web Services were computed using the WsRF and Normalisation techniques proposed in Al-Masri and Mahmoud as well as in Chapter Five of this dissertation. The critical observation revealed here is that both techniques achieved the same ordering of Web Services according to QoS based selection although different values were obtained even though the same ordering was realised. This confirmed that both techniques achieved similar results. However, the WsRF technique only selects at random from the set of services that attain similar QoS scores. The idea discussed in Chapter Four helps to solve the challenge by doing the prioritisation based on the trusted user feedback data that was generated by the AMC Web service selection mechanism.

The Email Validation Web Services of number 6, 9 and 16 respectively ("ServiceObjects", "Byteplant", and "BulkEmailVerifier") all achieved similar QoS computation. It should be noted however that the normalisation technique found in Zeng et al. (2004); Liu et al. (2004) and Makhlughian et al. (2012) ensure that it provides the three best Web Services amongst those considered. However, this information is not enough to justify the best of the three Web Services as the feedback rating system shows quite clearly that the Web Service number 9 (Byteplant) is rated above others of comparable performance. The rating of the Byteplant Email validation was seen to be relatively constant over a two-week period and maintained a rating of 100% from different service users. This final choice of the Byteplant over the other two is not only confirmed by the high range of computation selected based on the Normalisation during QoS computation but also affirmed by the extensive feedback range of quality maintenance. The Binomial probability distribution used by the recommendation system predicts in such a manner that Web Services are arranged orderly even to the least significant value to bring about the needed differences to differentiate a Web Service from another one of similar computation. The user feedback mechanism, therefore, made the difference when optimal (most relevant) Web Service is to be selected in an environment that has many Web Services with similar functionality but different QoS. This is achieved through the "breaking" of any occurrence of Web Services ties at run-time during invocation. Moreover, it ensures more user satisfaction, which is actually the major goal of the service provider, thus giving room for high competition and better improvement in provider Quality of Services.

6.3 Discussion

This study reported on the effectiveness of using QoS properties to optimally select out of a number of Web Services the most relevant Web Service that satisfy users' requests. The subject of selecting Web Services has constantly been a challenge. This is because the market is competitive and service providers seek to ensure better user satisfaction. The core of this study focused on utilisation of a continuous updated and unlimited range of users' Web Service assessment from users in the form of feedback for optimal (most relevant) service provisioning. In order to achieve this, this study implemented the common QoS normalisation technique (Liu et al., 2004; Makhlughian et al., 2012; Zeng et al., 2004), to regularise the positive and negative QoS properties. This step also ensures an even distribution of the QoS properties around comparable dimensions. To simplify the normalisation, this work deployed separate QoS normalising equations. The normalisation step leads to the expression of individual service users' preferences. This stage uses the assignment of weight technique to express the interest of intending service users. However, the normalisation equation could not solve for a situation where a Web Service tie occurs. That is a situation where Web Services of the same functionality have the same computation score.

How to resolve this issue of service ties after the service preference selection necessitates the use of feedback ratings. This allows service users to rate the performance of the Web Services differently within the system based on their experience. Series of experiments were designed and implemented to prove the feasibility and applicability of the QoS selection mechanism through the proof of concept methodology. The performance of the selection mechanism was measured based on the following performance metrics: execution time, throughput and service availability. The selected metrics were chosen because of their relevance within a Mobile Cloud environment. The time for service retrieval, successful service delivery and the availability of services are very important performance metrics that show if the system is ideal for a mobile environment. This researcher in this study also performed an experiment on how optimal service could be selected based on user feedback. The Proof of Concept is achieved with the use of live data in a real-life service provisioning environment. This study compared two approaches, namely, where feedback were not used to aid selection of Web Services and where the experiment was aided by feedback ratings.

The overall results showed that the service selection mechanism yielded a good throughput during service selection requests as the number of mobile requests increases. The average execution time has a relatively insignificant size of increment during service request even though the number of issued requests increases. This further shows that the system is able to execute service invocation within a very short period of time as required in a mobile environment to bring about optimal service provisioning. The result of service availability also shows a steep upward increment in service availability rate as the number of request increases. The increased requests are typically accompanied by an increased number of available mobile devices within the environment, thus providing various complementing Web Services for invocation. The availability was low at the initial stage of the test because the smaller quantity of Web Services is available for usage and with the increase in provider inputs, the availability rate increases. Moreover, the feedback effect test shows that users are more satisfied with the use of the prediction effect which enables the selection of best-rated services to users during selection ties. The possibility of selecting the optimal service without the use of user feedback is low and this often results in user dissatisfaction. From the results and the experiments performed, it was observed that out of seven outcomes, only one outcome produces optimal selection with a probability of 0.13. Therefore, the study concluded that the feedback section of the proposed system enhances better performance in Web Service selection as against the non-feedback.

The evaluation additionally uses some Web Service data from other sources which are real life Quality of Services. Section 6.2.5 utilised the data from the work of (Al-masri & Mahmoud, 2007) to test the performance of the QoS-based Web Service selection mechanism. There exist three top services whose performances were averagely the best out of the twenty Web Services that were used for the experiment. However, the Web service Ranking Function approach ranked the email validation services 'ServiceOjects', 'Byteplant', and 'BulkEmailVerifier' at the same level. Any of the three is selected at random because the system sees them as similar. The use of user feedback therefore simplifies the selection challenge and shows clearly that Web Service 'Byteplant' is better than the other two, based on user ratings currently recorded. The significance of this study is very useful especially when the number of Web Services increases, a situation that is prone to AMC Computing.

6.4 Chapter Summary

This chapter presented the results for the Ad-hoc mobile service selection mechanism. The experiments performed were based on execution time, throughput and service availability. The

Quality of Service parameters, together with user feedback ratings which also describe the quality of a service, were collected, stored and accessed for carrying out service selection. The purpose of conducting this experiment was to determine if the proposed design concepts actually fulfilled the earlier stated hypothesis. This study envisaged that incorporating extensive QoS properties into the formulated service selection mechanism would enhance more optimal service selection in the AMC. This goal was achieved by the service selection mechanism in that more optimal service was selected during service invocation especially when there are service selection ties within the AMC environment.

CHAPTER 7

CONCLUSION AND FUTURE WORK

7.1 Introduction

In recent years, Mobile devices have become an indispensable utility to human existence as a means of communication. It significantly enhanced the medium of communication between two different geographical locations, thereby conveying vital information and data via telecommunication networks. Gradually, mobile devices progressed towards transferring Jpeg files and other picture formats via a different communication network such as 2G/3G up to the recent 5G networks. With the increasing rate of mobile device performance, there have been some limitations in terms of battery capacity and memory capacity which obviously lower their performance in comparison with desktops and personal computers. The memory challenge is aided by the integration of the cloud computing platform with mobile devices for the benefit of accessing unlimited storage facilities. This also provides access to unlimited mobile service resources for users' consumption.

Mobile devices are capable of accessing Web Services at any geographical location from the cloud platform via the 3G/4G networks. The connectivity provided by the network providers is the major source of connection to the service provisioning platform. In a situation where the network facilities are not available or are quite unstable, it becomes a challenge for mobile devices to access the available service from the cloud server. The latest AMC system addresses this situation within a local community consisting of a number of mobile devices that could perform a dual role of service provider and the requestor. The AMC provides runtime service selection where the service interface is already discovered but its implementation still needs to be selected based on the Non-functional (QoS) service properties. This platform opens up the challenge of determining which Web Service attends to what request with a view to providing such a Web Service to the service requestor. This research systematically analysed the use of Quality of Service to determine the selection of optimal Web Service among the available service providers. This research used a platform that allows using additional QoS properties as well as the use of a continuous updated and unlimited range of users' Web Service assessment through user feedback data to inform the determination of the optimal Web Service for any issued request within the AMC system.

7.2 Discussion and Conclusion

Considering the importance of response time in any mobile environment, service selection has been seen, from the literature, as a general challenge. However, with a view to managing and ensuring that service consumers have access to optimal service during service invocation, the study proposed a service selection mechanism in the AMC which basically uses the QoS of Web Service in service selection. The QoS based selection mechanism seeks to identify the optimal Web Service selection during service invocation. Several Web Services in the Mobile Cloud server compete to meet up with the requests issued by the users. This competition, especially within a large number of Web Services, results in the occurrence of similar computation scores. Scholars who have used only certain QoS properties have not been able to fully address this issue.

This challenge necessitates the deployment of several QoS parameters to assist in determining the optimal Web Service within the cloud system. Both QoS factors (Web Service QoS and service feedback ratings) are essential in ensuring that appropriate Web Services are selected in the Ad-hoc mobile service provisioning platform. This work evaluated the proposed solution approach and determined the impact factor of user feedback as “extended” QoS in service selection through its influence on customers’ satisfaction.

The study deployed selected Web Service QoS factors and used the feedback information to differentiate between services that have similar computation. Several assumptions were also made during this study. Users with similar interests often end up with different services being provided for them. The feedback system was tailored towards providing an avenue that would ensure the ratings help to differentiate one service from another no matter how small the value is found to be. The prediction system of the feedback system helps to ensure that users achieve an optimal Web Service in the course of service invocation.

Moreover, the AMC provides a relatively easy and cost effective way of accessing services inherent in other mobile devices within a community, but the selecting of Web Services within this environment is a great challenge, especially in situations where there are a lot of mobile service providers as well as Web Services with similar functionalities.

In Section 1.5, three distinct research questions for the dissertation were defined. This section highlights how these questions were answered.

RQ1: Why are the current QoS-based service selection mechanisms not adequate for an AMC scenario?

The dissertation discussed the state-of-the-art in service selection mechanism for AMC computing in Chapter 3. The study discovered that most of these approaches have focused on the use of various techniques and tools to select services in service oriented paradigms and the various techniques used in Ad-hoc mobile computing have not yielded an optimal solution during service selection due to the occurrence of Web Service selection ties. This study also identified the dissatisfaction accrued due to the sub-optimal Web Service provided in the Ad-hoc mobile environment being attributed to the provisioning of sub-optimal service. Although the interest of service users should be paramount, providers are faced with the challenge of ensuring that the required services are made available to each of the requests received for their clients. As a result of this investigation, several ideas were synthesised and a service selection mechanism for AMC computing environment to address the aforementioned shortcomings was modelled.

RQ2: How can QoS information be used to enhance optimal Web Service selection in Ad-hoc mobile environments?

The research work envisaged that the incorporation of QoS properties that gives a continuous updated and unlimited range of users' Web Service assessment (e.g. user feedback) into service selection provide the opportunity to arrive at providing optimal service to service users. Various architectures, frameworks and mechanisms have been implemented to ensure that optimal services are selected. However, a necessary precaution has to be taken to ensure that situations where services attain the same QoS computation are taking care of to enhance optimal service provisioning. The researcher collected the various qualities of service properties and used normalisation techniques to ensure that qualities were regularised. A simple weight assignment system was implemented to ensure that user preferences were influenced during service query. The user ratings were used in the feedback mechanism to carry out appropriate optimal service selection that satisfies users' requirements.

RQ3: How can QoS-based service selection be evaluated in typical use case scenarios?

Chapter 4 proposed the Ad-hoc mobile service selection mechanism, consisting of five major components namely, Selection Middleware Engine, SQLite Database, Feedback Composer, Service Rater, and Mobile Users. The main focus of the work was on the selection middleware engine in collaboration with the feedback composer section. This component retrieves QoS information as well as the user ratings from the service database. This QoS information is further processed in relation to each uploaded Web Service in the Ad-hoc system. The QoS information carries the weights that describe the properties of such services. These values are stored in the SQLite database of the personal computer that serves as a central server within the system. The feedback composer also keeps a record of the ratings generated by each user of the service in the course of using the service. These are kept in the feedback records for usage when the need arises to check for the best rated during the occurrence of ties among Web Services. The mobile interfaces request for the service parameters which collect the functional and the Non-functional qualities of the required Web Services for processing.

The evaluation of the service selection mechanism to prove its feasibility and applicability were described in chapter 5. This study achieved this through the use of an m-Learning "proof-of-concept" prototype that allows users to specify the type of services needed from the AMC. Several services' WSDL files and Non-functional qualities were uploaded to test the performance of the selection mechanism. The work used the quantitative metrics of execution time, throughput and service availability to evaluate the effectiveness of the proposed Ad-hoc selection mechanism. This study further used the experiment to investigate the relationship between the use of QoS properties for service selection with respect to the increment in the number of service provider nodes and service requests. Moreover, the study also carried out an experiment to verify the effect of user feedback on user satisfaction, thereby statistically determining the percentage of user satisfaction. This has also helped to determine that the use of feedback-enabled system will actually enhance the selection of optimal (most relevant) Web Services as required within AMC computing paradigm.

The results showed that the service selection mechanism yields a good throughput performance during service selection requests even with increasing mobile requests. The average execution time has a relatively insignificant size of increment during service request even though the number of issued requests increases. This shows that the system is able to execute service invocation within a very short period of time, as required in a mobile environment, to bring about optimal service provisioning. The increased requests are typically accompanied by an increased number of available mobile devices within the environment, resulting in competing Web Services within the Ad-hoc cloud. The service availability was low when there were fewer requests, as a smaller number of Web Services are available for invocation but with the increased provider inputs, the service availability rates increases.

Further results show that the feedback effect ensures that users are more satisfied with the use of the prediction system that generates best-rated services to users during service selection ties. The possibility of selecting the optimal service without the use of user feedback is low as the previous systems automatically select any of the services randomly during selection ties. From the statistics and the experiment performed, it could be reasonably concluded that out of seven outcomes, the non-feedback effect is able to make an optimal selection only once thus only 13% probability is realisable in selecting an optimal service via QoS computation alone. Therefore, the feedback section of the system enhances the Quality of Services to influence the selection that ensures users are satisfied with the option that is returned to them.

In summary, this dissertation has addressed the following:

i. Reason why the current QoS-based service selection mechanisms are not adequate for an AMC scenario.

ii. The issue of how QoS information can be used to enhance optimal Web Service selection in AMC environments and

iii. How QoS-based service selection can be evaluated in a typical use case scenario.

7.3 Contributions to Knowledge

This research reports typically focussed on solving the challenge of Web Service selection ties during Web Service selection to clients. This work contributes to knowledge by filling in the gap to ensure that suitable and relevant Web Service(s) are selected at any Web Service request against previous approaches used in cloud computing which does not take care of selection ties. This is achieved by proposing a feedback-enhanced system that takes Web Service ratings and process it to influence optimal Web Service selection for intending Web Service users. This contribution alleviates a situation where any Web Service is selected at random whenever Web Service instances attain the same or similar aggregate computation scores.

The contribution also ensures that users' satisfaction is guaranteed based on available Web Services in the system. This work first implemented the approaches from other works (namely WsRF and non-feedback approaches) since solutions to selection ties were not implemented yet in AMC, and then compared it with our feedback enabled approach. The proposed approach in this write up shows through the results and the experiments performed that out of seven outcomes, only one outcome produces optimal selection of 0.13 probability with other approach being in use. Therefore, the study concluded that the feedback section of the proposed system enhances better performance in Web Service selection of probability of 1, as against the non-feedback and WsRF.

7.4 Limitations and Future work

Although the mechanism described in this work has been proved to enable optimal service selection among mobile devices, it has some limitations which need to be addressed in future research. The proposed mechanism advocates the use of feedback-enabled mechanism in addition to the considered Non-functional properties for service selection in a mobile environment. However, looking at the Ad-hoc environment today, the context of mobile devices is a vital area that considerable attention must be given. The context of mobile devices has some influence on the kind of services it could consume (Keidl & Kemper, 2004). The context also relies on the fact that Web Services are aware of their current environment e.g. location, user preferences and mobile device type. The context contains details about the environment of the device and the information surrounding it. Context also includes the consumer's name, address and the type of client device (software or hardware). In this research work, the model assumed that device contexts are relatively constant and that various users within the AMC environment consume almost similar services since they reside within a particular community such as m-Learning or m-Health. In other words, if these entire context parameters are embedded in the selection mechanism, it will likely enhance improved service selection, but the effect must be implemented in such a manner that it would not generate an overhead effect on the mechanism execution time.

Another limitation is the possibility of device failure in terms of the central personal computer which acts as a server within the AMC environment. The assumption throughout this work is that the server machine is well equipped with uninterrupted resources to make it continuously available within the Ad-hoc community such as a stable supply of power. Obviously, several devices exist within the Mobile Cloud and any of the devices could leave or join at any moment. The impact of the server PC will be clearly seen if it is down at any moment but the solution to this is via the auto-switching mechanism. This mechanism reconfigures to assign the task of the mobile server to a neighbouring compatible and more stable PC within the environment. Several experiments in the networking section of University of Zululand's Computer Science Department's Centre of Excellence, have addressed the reconfiguration challenges posed by a network failure. The task of reconfiguration during this event shows that the Ad-hoc cloud would contain several hundreds of mobile devices that would allow access to a potential server personal computer.

7.5 Recommendations

Having conducted the research on QoS-based Web Service selection in AMC computing, the results from this research help us to understand that the use of user feedback can be helpful in ensuring that more user satisfaction is enhance in the context of service provisioning. The solutions from the results showed that the slightest degree of feedback response can be used to make a difference in the type of Web Service that is selected for the user. Based on the results of this research, it is therefore recommended that implementing user feedbacks in distinguishing between Web Services with similar computation scores (Ties). The feedback ratings give the little difference that is needed to make the distinction that single out a Web Service to be better than the others who have achieved the same computation scores. However, the effect of the feedback system will be appreciative and meaningful when the users constantly supply the performance of the Web Services they consume to keep the database updated. This will go a long way to enhance better selection performance of the proposed selection mechanism.

Bibliography

Aagedal, J. Ø., & Earl F. Ecklund, J. (2002). Modelling QoS: Towards a UML Profile. In *Proceedings of the 5th International Conference on The Unified Modeling Language* (pp. 275–289).

Adigun, O., Emuoyibofarhe, O., & Migira, S. (2006). Challenges to Access and Opportunity to use SMME enabling Technologies in Africa. In *1st All Africa Technology Diffusion Conference, Johannesburg, South Africa*.

Akingbesote, A. O., Adigun, M. O., Oladosu, J. B., & Jembere, E. (2013). A Quality of Service Aware Multi-Level Strategy for Selection of Optimal Web Service. *IEEE International Conference on Adaptive Science and Technology, ICAST*. http://doi.org/10.1109/ICASTech.2013.6707518

Akinola, A. ., Adigun, M. ., & Akingbesote, A. . (2015). Qos-Aware Single Service Selection Mechanism for Ad-Hoc Mobile Cloud Computing. In *Proceedings on International Conference on Computing, Communication and Security (ICCCS) IEEE, Mauritius* (pp. 281–286).

Akinola, A. ., Adigun, M. ., Akingbesote, A. ., & Mba, I. . (2015). Optimal Route Service Selection in Ad-Hoc Mobile E-marketplaces with Dynamic Programming Algorithm using TSP Approach. In *proceedings on International Conference on E-Learning Engineering and Computer Softwares, kualar Lampur, Malaysia* (pp. 74–81).

Akinola, A. T., Adigun, M. O., & Mudali, P. (2016). A Federated Web Service Selection Approach for Ad-Hoc Mobile Cloud Computing. In *19th Southern African Telecommunication Networks and Applications Conference (SATNAC) Frankhurt, South Africa* (pp. 386–373).

Al-masri, E., & Mahmoud, Q. H. (2007). QoS-based Discovery and Ranking of Web Services. In *IEEE 4th International Conference on Cloud Computing Technology and Science* (pp. 529–534).

AlShahwan, F., & Faisal, M. (2014). Mobile Cloud Computing for Providing Complex Mobile Web Services. In *2014 2nd IEEE International Conference on Mobile Cloud Computing,* *Services, and Engineering* (pp. 77–84). http://doi.org/10.1109/MobileCloud.2014.12

Amoretti, M., Laghi, M. C., Carubelli, A., Zanichelli, F., & Conte, G. (2008). Reputation-based service selection in a peer-to-peer mobile environment. In *IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks, WoWMoM* (pp. 1–8). http://doi.org/10.1109/WOWMOM.2008.4594859

Averbakh, A., Krause, D., & Skoutas, D. (2009). Expoiting User Feedback to Improve Semantic Web Service Discovery. *Springer-Verlag Berlin Heidelberg*, *3*(2), 33–48.

Badidi, E. (2011). A Framework for Brokered Service Level Agreements in SOA Environments. In *2011 7th International Conference on Next Generation Web Services Practices (NWeSP)* (pp. 37–42).

Bahl, P., Han, R. Y., Li, L. E., & Satyanarayanan, M. (2012). Advancing the state of mobile cloud computing. In *Proceedings of the third ACM workshop on Mobile cloud computing and services - MCS '12* (p. 21). http://doi.org/10.1145/2307849.2307856

Bandyopadhyay, A., Si, T., Mondal, P. P., & Mallik, S. (2012). Proposed Conceptual Model for Semantically Enabled Web Services Based On QoS. *Procedia Technology*, *4*(31), 579–583. http://doi.org/10.1016/j.protcy.2012.05.092

Buthelezi, M. E., Adigun, M. O., Ekabua, O. O., & Iyilade, J. S. (2008). Accounting, pricing and charging service models for a GUISET grid-based service provisioning environment. In *Proceedings of the 2008 International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government, EEE 2008* (pp. 350–355).

Chen, C., Won, M., Stoleru, R., & Xie, G. G. (2014). Energy-Efficient Fault-Tolerant Data Storage & Processing in Mobile Cloud. *IEEE Transactions on Cloud Computing*, *3*(1), 1–14. http://doi.org/10.1109/TCC.2014.2326169

Coulouris, G., Dollimore, J., & Kingberg, T. (2001). Distributed Systems: Concept and Design. *3rd Edition, Addison-Wesley,*.

Cubera, F., Duftler, M., Nagy, R., Mukhi, W., & Weerawarana, S. (2002). Unraveling the Web Services Web: An Introduction to SOAP, WSDL and UDDI. *IEEE Internet Computing*, *6*(2), 86–93.

D'Ambrogio, A. (2006). A Model-driven WSDL Extension for Describing the QoS of Web Services. In *IEEE International Conference on Web services* (pp. 69–76).

Deora, V., Shao, J., Gray, W. A., & Fiddian, N. J. (2003). A Quality of Service Management Framework Based on User Expectations. *Springer-Verlag Berlin Heidelberg ICSOC, LNCS 2910*, *2*(3), 104–114.

Dev, D., & Baishnab, K. L. (2014). A Review and Research Towards Mobile Cloud Computing. In *2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering* (pp. 252–256). Ieee. http://doi.org/10.1109/MobileCloud.2014.41

Dinh, H. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing : architecture , applications , and approaches. *Wireless Communication and Mobile Computing*, *13*(18), 1587–1611. http://doi.org/10.1002/wcm

Eleyan, A., & Zhao, L. (2010). Extending WSDL and UDDI with Quality Service Selection Criteria. In *Proceedings of the 3rd International Symposium on Web Services: 3rd International Symposium on Web Services* (pp. 1–10).

Elgazzar, K., Martin, P., & Hassanein, H. (2014a). DaaS : Cloud-based mobile Web service discovery. *Pervasive and Mobile Computing*, *13*(25), 67–84. http://doi.org/10.1016/j.pmcj.2013.10.015

Elgazzar, K., Martin, P., & Hassanein, H. (2014b). Mobile Web Services : State of the Art and Challenges. *International Journal of Advance Computer Science and Applications*, *5*(3), 173–188.

Fangming, L., Peng, S., Hai, J., Linjie, D., & Jie, Y. (2013). Gearing Resource -Poor Mobile Devices with Powerful Clouds: Architectures, Challenges, and Applications. In *IEEE Wireless Communications* (pp. 2–10).

Gouscos, D., Kalikakis, M., & Georgiadis, P. (2004). An Approach to Modeling Web Service QoS and Provision Price. In *Proceedings of the fourth International Conference on Web Information Systems Engineering Workshops*.

Guitart, J., Torres, J., & Ayguade, E. (2010). A survey on performance management for internet. *Wiley InterScience, Concurrency Comput.: Pract. Exper.*, *9*(12), 68–106.

http://doi.org/10.1002/cpe

Hamad, H., Saad, M., & Abed, R. (2010). Performance evaluation of restful web services for mobile devices. *International Arab Journal of E-Technology, 1*(3), 72–78.

Huang, L., & Deng, S. (2014). Service Selection for Mobile Service Orchestration. In *IEEE International Conference on Mobile Services* (pp. 147–148). http://doi.org/10.1109/MobServ.2014.29

Huerta-canepa, G., & Lee, D. (2010). A Virtual Cloud Computing Provider for Mobile Devices. In *ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond* (pp. 35–39).

IBM. (2010). Strategies for assessing cloud security.

Iordache, R., & Moldoveanu, F. (2014). QoS-Aware Web Service Semantic Selection Based on Preferences. *Procedia Engineering : 24th DAAAM International Symposium on Intelligent Manufacturing and Automation*, *69*, 1152–1161. http://doi.org/10.1016/j.proeng.2014.03.104

Jarir, Z., Boumhamdi, K., & Quafafou, M. (2009). Context- and QoS Aware Personalization for Web Services : Web Information Extraction Case Study. In *Fifth International Conference on Signal Image Technology and Internet Based Systems* (pp. 401–406). http://doi.org/10.1109/SITIS.2009.69

Jung, E., & Wang, Y. (2010). User-Profile-Driven Collaborative Bandwidth Sharing on Mobile Phones.

Kaushik, N., & Kumar, J. (2014). A Literature Survey on Mobile Cloud Computing : Open Issues and Future Directions. *International Journal of Engineering and Computer Science*, *3*(5), 6165–6172.

Kayastha, R. R., & Baria, J. (2015). A Survey on Web Service Selection and Ranking Methods. *International Journal of Computer Applications*, *117*(16), 5–8.

Keidl, M., & Kemper, A. (2004). Towards context-aware adaptable web services. In *13th international conference on World Wide Web - ACM'04 New York, USA.* (pp. 55–65).

http://doi.org/10.1145/1010432.1010443

Khan, F. H., Bashir, S., Javed, M. Y., Khan, A., & Khiyal, M. S. H. (2010). QoS Based Dynamic Web Services Composition & Execution. *International Journal of Computer Science and Information Security*, *7*(2), 147–152.

Kirby, G., Dearle, A., & Macdonald, A. (2011). An Approach to Ad hoc Cloud Computing. *Proceedings - International Conference on Next Generation Web Services Practices*.

Klusch, M., & Kapahnke, P. (2008). Semantic web service selection with SAWSDL-MX. In *Proceedings of the Second International Workshop on Service Matchmaking and Resource Retrieval in the Semantic Web (SMRR 2008)* (pp. 1–15).

Kokash, N., & D'Andrea, V. (2007). Evaluating Quality of Web Services: A Risk-driven Approach. In *Proceeding of Business Information System Conference, Germany, Springer* (pp. 180–194). http://doi.org/10.1109/ICWS.2013.83

Kondratyeva, O., Kushik, N., Cavalli, A., & Yevtushenko, N. (2013). Evaluating Quality of Web Services: A Short Survey. In *Proc. of 20th International Conference on Web Services IEEE* (pp. 587–594). http://doi.org/10.1109/ICWS.2013.83

Korkmaz, T., & Krunz, M. (2001). Multi-constrained optimal path selection. In *Proceedings IEEE INFOCOM Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)* (Vol. 2, pp. 834–843). http://doi.org/10.1109/INFCOM.2001.916274

Kovachev, D., Cao, Y., & Klamma, R. (2011). Mobile Cloud Computing: A Comparison of Application Models. In *Networking and Internet Architecture: Distributed , paralleland Cluster Computing* (pp. 1–8). http://doi.org/10.1007/s11042-012-1100-6

Kovachev, D., Cao, Y., & Klamma, R. (2012). Mobile Cloud Computing : A Comparison of Application Models. In *Distributed, Parallel, and Cluster Computing (cs.DC);*

Kun, Z., Hong, Z., Liming, J., & Jian, X. (2010). Composite Agent Service Selection Algorithm for Non-functional Attributes based on Simulated Annealing. In *Second WRI World Congress on Software Engineering* (pp. 101–106). http://doi.org/10.1109/WCSE.2010.79

Lacuesta, R., Lloret, J., Sendra, S., & Peñalver, L. (2014). Spontaneous Ad Hoc Mobile Cloud Computing Network. *The Scientific World Journal*, *2014*(10), 19. http://doi.org/10.1155/2014/232419

Liu, G., Wang, Y., Orgun, M. ., & Lim, E. P. (2010). A Heuristic Algorithm for Trust-Oriented Service Provider Selection in Complex Social Networks. In *Services Computing (SCC), 2010 IEEE International Conference* (pp. 130–137).

Liu, S., Guo, S., Chen, X., & Lan, M. (2009). A QoS Based Web Service Selection Model. In *2009 International Forum on Information Technology and Applications* (pp. 145–148). http://doi.org/10.1109/IFITA.2009.145

Liu, Y., Ngu, A. H. H., & Zeng, L. (2004). QoS Computation and Policing in Dynamic Web Service Selection. In *proceedings of the 13th International WWW Conference, ACM* (pp. 66–73).

Makhlughian, M., Hashemi, S. M., Rastegari, Y., & Pejman, E. (2012). Web Service Selection Based on Ranking of QoS using Associative Classification. *International Journal on Web Service Computing (IJWSC)*, *3*(1), 1–14.

Marinelli, E. E. (2009). Hyrax : Cloud Computing on Mobile Devices using MapReduce. *Master Thesis, Carnegie Mellon University*, *389*(September).

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing — The business perspective. *Decision Support Systems*, *51*(1), 176–189. http://doi.org/10.1016/j.dss.2010.12.006

Marwaha, P., Banati, H., & Bedi, P. (2013). UDDI Extensions for Temporally Customized Web Services. In *International Conference on Computational Intelligence: Modeling Techniques and Applications (CIMTA) 2013* (Vol. 10, pp. 184–190). Elsevier B.V. http://doi.org/10.1016/j.protcy.2013.12.351

Maximilien, E. M., & Singh, M. P. (2005). Multiagent system for dynamic web services selection. In *Proceedings of 1st Workshop on Service-Oriented Computing and Agent-Based Engineering (SOCABE at AAMAS)* (pp. 25–29).

Nevedrov, D. (2006). *Using JMeter to Performance Test Web Services*.

Papazoglou, M. P. (2008). Web Services: Principles and Technology. *1st Edition, INFOLAB/CRISM, Tilburg University, Netherlands, Prentice Hall*.

Ran, S. (2003). A model for web services discovery with QoS. *SIGecom Exch, ACM*, *4*(1), 1–10. http://doi.org/10.1145/844357.844360

Sabahi, F. (2011). Cloud computing security threats and responses. *2011 IEEE 3rd International Conference on Communication Software and Networks*, 245–249. http://doi.org/10.1109/ICCSN.2011.6014715

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International World Wide Web Conference, Hong Kong* (Vol. 1, pp. 285–295). http://doi.org/10.1145/371920.372071

Sathya, M., Swarnamugi, M., Dhavachelvan, P., & Sureshkumar, G. (2012). Evaluation of QoS Based Web- Service Selection Techniques for Service Composition. *International Journal of Software Engineering (IJSE)*, *1*(5), 73–90.

Schafer, J. Ben, Frankowski, D., Herlocker, J., & Sen, S. (2007). Collaborative Filtering Recommender Systems. *The Adaptive Web*, *4*(2), 81–173. http://doi.org/10.1561/1100000009

Shah-Hosseini, H. (2007). Problem Solving by Intelligent Water Drops. In *IEEE Congress on Evolutionary Computation, Singapore* (pp. 3226–3231).

Shaikh, F. B. F., & Haider, S. (2011). Security threats in cloud computing. *2011 International Conference for Internet Technology and Secured Transactions*, 214–219.

Singh, R. P., & Pattanaik, K. K. (2013). An Approach to Composite QoS Parameter based Web Service Selection. *Procedia Computer Science*, *19*(2), 470–477. http://doi.org/10.1016/j.procs.2013.06.063

Sivakumar, M., & Magendiran, M. E. (2012). Web Service Filtering And Ranking With Qos Properties. *International Journal of Advanced Research In Technology*, *2*(4), 46–49.

Srirama, S. N., & Paniagua, C. (2013). Mobile Web Service Provisioning and Discovery in Android Days. *IEEE Software*, *30*(1), c2-1. http://doi.org/10.1109/MS.2013.14

Susila, S., Vadivel, S., & Julka, A. (2012). Broker architecture for Web Service selection using SOAPUI. In *Cloud Computing Technologies, Applications and Management (ICCCTAM), 2012 International Conference on* (pp. 219–222). http://doi.org/10.1109/ICCCTAM.2012.6488102

Swarnamugi, M. (2013). Taxonomy of Web Service Selection Approaches. *International Journal of Computer Applications*, *2*(4), 18–22.

Sycara, K., Paolucci, M., Ankolekar, A., & Srinivasan, N. (2003). Automated discovery, interaction and composition of Semantic Web services. *Web Semantics*, *1*(1), 27–46. http://doi.org/10.1016/j.websem.2003.07.002

Varshavsky, A., Reid, B., & Lara, E. De. (2005). A Cross-Layer Approach to Service Discovery and Selection in MANETs. In *IEEE Transaction Software Engineering* (pp. 66–73).

Voas, J., & Zhang, J. (2009). Cloud Computing: New Wine or Just a New Bottle? *IT Professional*, *11*(2), 15–17. http://doi.org/10.1109/MITP.2009.23

Wagh, K. S., & Thool, R. . (2013). Web Service Provisioning on Android Mobile Host. *International Journal of Computer Application*, *81*(14), 1–7.

Wagh, K., & Thool, R. (2014). Mobile Web Service Provisioning and Performance Evaluation of Mobile Host. *International Journal on Web Service Computing (IJWSC)*, *5*(2), 2573–2583.

Wang, P., Chao, K.-M., & Lo, C.-C. (2010). On optimal decision for QoS-aware composite service selection. *Information Technology Journal*, *9*(6), 1207–1211. http://doi.org/10.1016/j.eswa.2009.05.070

Wang, S., Zibin, Z., Qibo, S., Hua, Z., & Fangchun, Y. (2011). Reliable web service selection via QoS uncertainty computing. *International Journal of Web and Grid Services*, *volume 7*(issue 4), pages 410-426.

Wang, X., Vitvar, T., Kerrigan, M., & Toma, I. (2008). A QoS-aware Selection Model for Semantic Web Services. In *Proceedings - International Conference on Next Generation Web Services Practices* (pp. 1–12).

Wang, Y., & Vassileva, J. (2007). A Review on Trust and Reputation for Web Service Selection.

In *27th International Conference on Distributed Computing Systems Workshops* (pp. 3–8). http://doi.org/10.1109/ICDCSW.2007.16

Wu, H., Wang, Q., & Wolter, K. (2012). Methods of Cloud-path Selection for Offloading in Mobile Cloud Computing Systems. In *IEEE 4th International Conference on Cloud Computing Technology and Science* (pp. 443–448).

Xin, M., Lu, M., & Zhang, R. (2014). A Location-based Services Selection Model and Algorithm with QoS Constraints under Mobile Internet Environment. In *International Conference on Service Sciences* (pp. 124–129). http://doi.org/10.1109/ICSS.2014.51

Yang, K., Galis, A., & Chen, H.-H. (2010). QoS-Aware Service Selection Algorithms for Pervasive Service Composition in Mobile Wireless Environments. *Mobile Networks and Applications, Springer Science*, *15*(4), 488–501. http://doi.org/10.1007/s11036-009-0189-y

Yang, X., & Bouguettaya, A. (2006). Efficient Access to Wireless Web Services. In *Proceedings of the 7th International Conference on Mobile Data Management (MDM'06)* (pp. 1–6).

Yu, T., & Lin, K.-J. (2005). Service selection algorithms for Web services with end-to-end QoS constraints. *Springer-Verlag 2005*, *3*(10), 103–126. http://doi.org/10.1007/s10257-005-0052-z

Zeng, L., Benatallah, B., Ngu, A., Dumas, M., Kalagnanam, J., & Chang, H. (2004). QoS-Aware Middleware for Web Services Composition. *IEEE Transactions on Software Engineering*, *30*(5), 311–327.

Zhang, L., Ding, X., Wan, Z., Gu, M., & Li, X. (2010). WiFace : A Secure GeoSocial Networking System Using WiFi-based Multi-hop MANET. In *MCS' 10, ACM, San Francisco, USA.*

Zhang, Z., Sun, W., Chen, W., & Peng, B. (2008). An Integrated Approach to Service Selection in Mobile Ad Hoc Networks. In *4th IEEE International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2008)* (pp. 1–4).

Zheng, Z., Ma, H., Lyu, M. R., & King, I. (2009). WSRec: A Collaborative Filtering Based Web Service Recommender System. In *2009 IEEE International Conference on Web Services* (pp. 437–444). http://doi.org/10.1109/ICWS.2009.30

Zhu, J., Kang, Y., Zheng, Z., & Lyu, M. R. (2012). A Clustering-Based QoS Prediction Approach for Web Service Recommendation. In *Proceedings - 15th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops,* (pp. 93–98). http://doi.org/10.1109/ISORCW.2012.27

Zulkernine, F., Martin, P., Craddock, C., & Wilson, K. (2009). A Policy-based Middleware for Web Services SLA Negotiation. In *2009 IEEE International Conference on Web Services* (pp. 1043–1050).