**Ain Shams University**
**Faculty of Computers and information Science**
**Computer Science Department**

# Reliable Web Services through Mobile Cloud Computing

A Thesis submitted in partial fulfillment of the requirements for the degree of Master of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University.

By

**Amr Elsayed Mohamed Abdelfattah**
Demonstrator at Computer Science Department.
Faculty of Computer and Information Sciences, Ain Shams University.

Under the Supervision of

**Prof. Dr. El-Sayed M. El-Horbaty**
Professor of Computer Science,
Head of Computer Science Department,
Faculty of Computer and Information Sciences, Ain Shams University.

**Dr. Tamer A. Mostafa**
Lecturer of Computer Science,
Department of Information Systems,
Faculty of Computer and Information Sciences, Ain Shams University.

Cairo 2016

**Ain Shams University**
**Faculty of Computers and information Science**
**Computer Science Department**

# Reliable Web Services Consumption through Mobile Cloud Computing

A Thesis submitted to the department of Computer Science, Faculty of Computer and Information Sciences, Ain Shams University, in partial fulfillment of the requirements for the degree of Master of Computer Science

By

**Amr Elsayed Mohamed Abdelfattah**

Approved by the discussion committee:

**Prof. Dr. El-Sayed M. El-Horbaty** — Member and supervisor
Professor of Computer Science,
Head of Computer Science Department,
Ain Shams University.

---

**Prof. Dr. Abdel-Badeeh M. Salem** — Member
Professor of Computer Science,
Ain Shams University.

---

**Prof. Dr. Mohamed S. Waheed** — Member
Professor of Computer Science,
Suez Canal University.

---

Computer Science Department
Faculty of Computer and Information Sciences
Ain Shams University
Cairo 2016

# Acknowledgements

Firstly, I want to thank ALLAH who always supports me with power and enthusiasm to comprehensive with this topic.

Secondly, I would like to express my sincere gratitude to my advisors Prof. El-Sayed M. El-Horbaty and Dr.Tamer A. Mostafa for the continuous support of my Master study and related research, for their patience, motivation, and immense knowledge. Their guidance helped me in all the time of research and writing of this thesis. I could not have imagined having better advisor and mentor for my Master study.

My sincere thanks also goes to Cryptech for SW Development company that provided me an opportunity to join their cloud team to practice and discuss the related point of research, Prof.Dr.Abdelbadeeh Salem for his encouragement and his support in the INASE conference, and to LA.Ahmed Salah for his appreciated comments and advices.

I thank my colleges and friends for their motivation and encouragement to me all the time of working in the Master.

Last but not the least, I would like to thank my wife and my family; my mother and to my brother and sisters for supporting me spiritually throughout working in the Master and my life in general.

# Abstract

The Mobile intermittent wireless connectivity limits the evolution of the mobile landscape, such that the mobile usage progress is coupled with the ubiquitous nature of the Internet and the web services. Achieving web service reliability results in low communication overhead and correct retrieval of the appropriate state response. The most recent approaches that achieve the reliability of web services consumption are the Middleware approach and Mobile Agent (MA) approach.

In this thesis, we discuss and analyze the two approaches that achieve the reliability of web services consumed by mobile devices, and propose three approaches based on an enhanced architecture that achieves the reliability under various conditions with minimum communication data overhead, these approaches are: Reliable Service Architecture using Middleware approach (RSAM), Reliable Approach using Middleware and Web Socket (RAMWS), and Reliable Approach using Mobile Agent and Web Socket (RMAWS).

The RSAM focuses on ensuring and tracking the request execution under the communication limitations and service temporal unavailability constraints. It considers the most measurement factors including: request size, response size, and consuming time. We conducted experiments to compare the enhanced architectures with the traditional one. In these experiments, we covered different cases to prove the achievement of reliability such that we considered, 3 request size classes (Small, Medium, Very Large), 4 response size classes (Small, Medium, Large, Very Large), and 3 consuming time classes (Small, Large, Very Large "Leads to Time-out"). The comparison results show that the request size was found to be constant, the response size is identical to the traditional architecture, and the increase in the consumption time was less than 5% of the transaction time with the different response sizes. The RAMWS and RMAWS have proven to achieve the reliable web services consumption in terms of overcoming the Time-out problem. The RMAWS is a hybrid approach between the mobile agent approach and web socket open connection communication protocol. The RAMWS architecture is a hybrid approach between the middleware approach and web socket protocol. Moreover, we introduce the Two-Response approach, which the first one is the temporary response that returns in a short time through the Representational State Transfer (REST) service response, and the other is the actual response, which returns smoothly through the open connection protocol.

# Contents

## List of Tables

# List of Figures

# List of Abbreviations

| | |
|---|---|
| **ACID** | **A**tomicity, **C**onsistency, **I**solation, and **D**urability |
| **BASE** | **B**asically **A**vailable **S**oft-state **E**ventual consistency |
| **CAP** | **C**onsistency, **A**vailability**,** and **P**artition-tolerance |
| **CSC** | **C**loud **S**ervice **C**onsumer |
| **CSCo** | **C**loud **S**ervice **Co**mponent |
| **HTTP** | **H**yper **T**ext **T**ransfer **P**rotocol |
| **JSON** | **J**avaScript **O**bject **N**otation |
| **MCC** | **M**obile **C**loud **C**omputing |
| **MA** | **M**obile **A**gent |
| **MCCo** | **M**obile **C**loud **Co**nsumer |
| **MSC** | **M**iddleware **S**ervice **C**omponent |
| **REST** | **RE**presentational **S**tate **T**ransfer |
| **RSAM** | **R**eliable **S**ervice **A**rchitecture using **M**iddleware |
| **RAMWS** | **R**eliable **A**pproach using **M**iddleware and **W**eb **S**ocket |
| **RMAWS** | **R**eliable Approach using **M**obile **A**gent and **W**eb **S**ocket |
| **SOAP** | **S**imple **O**bject **A**ccess **P**rotocol |
| **SCSC** | **S**ocket **C**loud **S**ervice **C**onsumer |
| **SMSC** | **S**ocket **M**iddleware **S**ervice **C**omponent |
| **UDDI** | **U**niversal **D**escription **D**iscovery and **I**ntegration |
| **WSA** | **W**eb **S**ervice **A**rchitecture |
| **WSDL** | **W**eb **S**ervice **D**escription **L**anguage |
| **XML** | E**X**tensible **M**arkup **L**anguage |
| **2PC** | **T**wo **P**hase **C**ommit protocol |
| **3PC** | **T**hree **P**hase **C**ommit protocol |

# Chapter (1)

## Introduction

# Introduction

The evolution of the mobile landscape coupled with the ubiquitous nature of the Internet and the recent explosion of the cloud computing technology is facilitating the deployment of web services. The web services are the perfect way to provide a standard platform and operating system independent mechanism for enterprise and personalized mobile applications communication over the Web.

Smart phones are gradually becoming the effective client platform to consume the services and the pool of data and information. This is because cloud computing improves scalability and consistency of services and data, and facilitates the deployment of enterprise and personalized mobile applications [1].

The Mobiles "thin clients" are uncomfortable and expensive in terms of time and effort. Because of their limited processing power and intermittent wireless connectivity, there is uncertainty of whether the web service request was successfully received, was lost in the Internet before reaching the server, or was partially processed. In the case the application retries the operation and resends the request, it may be duplicated or cause an error, such as two orders entered or two credit card charges.

Web services encapsulate Cloud Computing because Cloud Computing uses web services for connections. Most of these cloud oriented services and data are deployed as web services that are network-oriented applications [2]. The synchronization between a mobile device and a web service is achieved through initiating a conversation in a request response pattern.

Mobile Cloud Computing (MCC) is the combination of cloud computing, mobile computing and wireless networking to bring rich computational resources to mobile users, network operators, as well as cloud computing providers [3].

The reliable web services through mobile cloud computing achieved using approaches that focus on ensuring the request execution under the intermittent connectivity, services unavailability conditions, moreover ensuring the appropriate response according to the request state.

## 1.1 Client

Architecture enables client device to exchange data with a host [4]. The client architecture has two types, thin client and thick or fat client. They will be discussed in the next two sub sections.

### 1.1.1 Thin client

A client device allows exchanging data with a host with minimal processing at client level [4]. It has to be connected to the server all the time. It can run on any system support browser [5].

It is suited in data collection applications, especially in mobile environment because of limited capabilities of mobiles [5].

### 1.1.2 Thick client

Is allowed to exchange data, but it has a local copy of the database, a local GUI and some business logic on the handheld device. It doesn't need to be connected to the server all the time. It connects to synchronize with the server to get updates.

## 1.2 Web services types and limitations

Web services act as self-contained components, which are published, located and invoked over the web. The key concept behind web services is to provide a standard platform and operating system independent mechanism for application communication over the web.

The web service Architecture (WSA) [6] proposed by W3 organization relies on a number of Web standards, such as Simple Object Access Protocol (SOAP), Web Service Description Language (WSDL), EXtensible Markup Language (XML), and Universal Description Discovery and Integration (UDDI) that allows services to be searched, described and integrated by any application [7].

There are two types of web services: Simple Object Access Protocol (SOAP) and Representational State Transfer (REST).

### 1.2.1 Simple Object Access Protocol (SOAP)

SOAP web services development, their clients require extra effort mostly due to the lack of native support. The two major SOAP-based web services limitations are categorized mainly to communication and computation overhead [7]:

I. Communication overhead

- The verbose nature of the XML-based messaging system was not intended for communication efficiency.
- Communication performance issues due to lack of support for transaction in communication with a web service.
- Communication may suffer poor performance over busy or unstable networks in comparison with other traditional approaches of distributed computing such as CORBA or DCOM.

- It makes this kind of data exchange protocols stateless, as the web service provider and web service consumer don't have knowledge of each other's state.

II. Computation overhead

- Encoding/Decoding XML requests/responses definitely degrades the overall application performance.
- Performance issues due to XML processing and parsing overhead.

### 1.2.2 Representational State Transfer (REST)

The REST architecture is fundamentally client-server architecture, and is designed to use a stateless communication protocol, typically Hyper Text Transfer Protocol (HTTP). In the REST architecture, clients and servers exchange representations of resources using a standardized interface and protocol. These principles encourage REST applications to be simple, lightweight, and have high performance. Therefore, regarding the scope of reliability RESTFUL services overcome SOAP services limitations and achieve better results especially in mobile communications [8]-[10]. Using REST as a service architecture is preferred for mobile devices because REST services use HTTP request and response, which means that a mobile device connected with the internet can access the service without additional overhead, unlike SOAP web services [11].

In [12], the authors observed that using REST enabled their system to scale with multiple users and devices, but also complained about request-response mechanism, which sometimes limits the system at the network level.

In [13], the authors argue that many firewalls permit only the GET and POST methods. There is also a size limit on the URI for the GET method encoding; and HTTPS is not cacheable [1].

According to [14], combining RESTful design with other technologies such as caching provides good system scalability.

## 1.3 Web Socket

The web socket Protocol is an independent TCP-based protocol. It makes more interaction between a client and a server, facilitates the real-time data transfer from and to the server. This is made possible by providing a standardized way for the server to send content to the client without being solicited by the client, and allows messages to be passed back and forth while keeping the connection open [15].

## 1.4 Reliability and Challenges

The challenges in mobile web services consumption are worth investigating from two perspectives: Mobile limitations, and connectivity limitations from Mobile side and Cloud service provider side. In addition the network bandwidth limitation and unreliable wireless communication are decreasing the overall support for web services consumption on mobile devices [7]. The challenges are listed in the following points:

- Connection loss: Form the perspective of Smart phones, clients have intermittent connection because of their mobility. They can be momentarily removed from the connected network and later join the available network [11]. From the perspective of Cloud/ Server, it may lose the connection and their web services become unreachable from clients.
- Latency/ Bandwidth: Mobile Cellular networks have a very limited bandwidth and are often billed based on the amount of data transferred.
- Reliability: The public Internet is unreliable if a client calls a web service and doesn't get a response within the timeout period, the client doesn't know whether the request was received, partially processed, or lost to perform the appropriate action.
- Longer Transaction time: Web service consumption will take longer to execute than an equivalent direct query against a local database. The consumption will be slower because of the HTTP overhead, the XML overhead, and the network overhead to a remote server. Therefore the differences in performance between Web services and traditional database queries need to be factored into the application architecture to prevent unexpectedly poor performance due to the latency of the web services consumption failures.
- Time-out: The service Time-out problem is one of the most effected issues in the mobile experience, such that the services response size, database relations' communication time, and the required computations in the services are continuously increasing corresponding to the application usage during the time, which causes repeat timeout through the consumption of these services.

However, there is even a bigger challenge that has been overlooked recently by researchers within the distributed mobile network, which is the "CAP Theorem" [16], discussed in Chapter 2. In a highly distributed system where web services are scattered across multiple platforms, three system guarantees are required: Consistency of the data, Availability of the system/data, and Partition tolerance to fault. However, the CAP theorem states that at most only two of the three can be guaranteed simultaneously. In distributed mobile systems where the mobile node is employed as the client platform of the web services, partition tolerance is a given because of the intermittent connectivity losses. This means we are forced to choose between Availability and Consistency [1].