

School of Computing and Engineering

Final Year Project

Abstract

*Project Title:*

Designing and implementing

RESTful Web Services

*Undertaken By:*

Khan Najam ul Asre (21303779)

*Submitted On:*

??????

*Supervisor:*

??????

*Second Marker:*

??????

Contents

[1 Introduction 3](#_Toc511849184)

[1.1 Background 3](#_Toc511849185)

[1.1.1 Monolithic Applications 3](#_Toc511849186)

[1.1.2 Information Technology 3](#_Toc511849187)

[1.1.3 Distributed Systems 3](#_Toc511849188)

[1.1.4 Abundance of Computing Devices 3](#_Toc511849189)

[1.1.5 Heterogeneity 3](#_Toc511849190)

[1.1.6 System Agility 4](#_Toc511849191)

[1.1.7 Service Oriented Architecture (SOA) 4](#_Toc511849192)

[1.1.8 Microservices Architecture (MSA) 4](#_Toc511849193)

# Introduction

## Background

### Monolithic Applications

Early computers had limited processing, storage and communication capabilities. Despite, they were too expensive and too large in size for individuals to own or maintain. Hence, they were used only within the large organisations. The software systems were monolithic where functionally distinct concepts, e.g. data persistence and retrieval, business logic, user interface, error handling and logging, were strongly interwoven without any clear boundaries or architectural separation. Business requirements would change rarely and hence the software systems. It was very common that a version of software remained useful for the business for years, so monolithic applications worked really well.

### Information Technology

Over the last few decades, computing technology has seen dramatic advancement. Computing devices are becoming smaller and smaller in physical size, cheaper in cost but growing in computational power, data storage capacity and communication capabilities. The introduction of the Internet provided a global communication infrastructure. These factors together gave birth to the Information Technology which involves “the development, maintenance and use of computer systems, software and networks for the processing and distribution of data”[[1]](#footnote-1).

### Distributed Systems

The powerful devices and global communication network infrastructure revolutionised the business information and management systems. It was now possible for organisations to have geographically isolated locations with “autonomous computers, connected through a network and distribution middleware, which enabled computers to coordinate their activities and to share the resources of the system, so that users perceive the system as a single, integrated computing facility.”[[2]](#footnote-2)

### Abundance of Computing Devices

Once thought to be of the interest of large enterprises, computing technology has now become a household commodity. With the introduction of smart hand-held devices computing devices are now personal pocket-items. This has changed software requirements of the business organisations. The market has become competitive. The businesses have to reach vast customer-base across the globe.

### Heterogeneity

Lots of devices and manufacturers means lots of operating platforms and lots of software development frameworks. This poses the challenge of interoperability. Today’s ideal software systems have to be platform independent and capable of communicating and working with systems built using various frameworks and running on various platforms. As organisation’s customer base grows so does the need for system interoperability, to ensure that business is able to reach customer owning different devices running on different platform.

### System Agility

To keep going alongside the competitors, organisations have to change their marketing strategy and product presentation quickly and continuously. This requires the software systems that are agile and responsive, that can be changed quickly with or without the need of redeployment; or support Continuous Integration and Delivery.

### Service Oriented Architecture (SOA)

Unfortunately, monolithic applications lacked the architectural separation of concerns, therefore unable to become distributed systems. Monolithic software systems are mostly built for single platform using single framework. They are difficult, even impossible sometimes, to be changed or scaled. System designed to run on a single platform lack the interoperability, therefore reaching customer with devices running on different platform is not possible. Monolithic systems therefore failed to meet aforementioned challenges This lead the software architects to favour software systems composed of small, self-contained, independent and interoperable components rather than a giant monolithic system. Service oriented architecture (SOA) was one answer to such problems. SOA is a “system architecture in which application functions are built as components (services) that are loosely coupled and well-defined to support interoperability and to improve flexibility and reuse”[[3]](#footnote-3).

### Microservices Architecture (MSA)

“Microservices Architecture is basically Service Oriented Architecture done well”[[4]](#footnote-4). Microservices Architecture “is a method of developing software applications as a suit of independently deployable, small, modular services in which each service runs a unique process and communicates through a well-defined, lightweight mechanism to serve a business goal”[[5]](#footnote-5). Being able to meet almost all challenges of modern businesses, in the recent years MSA for many organisations has become a preferred architecture of creating enterprise applications. Martin Fowler notes Netflix, Amazon, eBay, Twitter, UK Government Digital Services and many other applications and websites have evolved from monolithic to MSA[[6]](#footnote-6).

There are number technology options to implement distributed software systems, for example, Remote Procedure Calls (RPC) and SOAP based services. RPC lacks the standardised contracts hence needs comprehensive custom documentation. SOAP on the other hand is a heavyweight protocol and services build using SOAP are not agile enough to respond change requests quickly. Change in service contract needs the modification, rebuilding and redeployment of client software. It is harder for components to evolve independently.

---- ilities --- falasies -- REST

1. https://www.merriam-webster.com/dictionary/information%20technology [↑](#footnote-ref-1)
2. http://www0.cs.ucl.ac.uk/staff/ucacwxe/lectures/ds98-99/dsee3.pdf [↑](#footnote-ref-2)
3. Service-Oriented Architecture (SOA) Compass: Business Value, Planning, and Enterprise Roadmap Bieberstein, Bose, Fiammante, Jones, & Shah, 2006 [↑](#footnote-ref-3)
4. https://www.pluralsight.com/courses/microservices-architecture [↑](#footnote-ref-4)
5. https://smartbear.com/learn/api-design/what-are-microservices/ [↑](#footnote-ref-5)
6. https://martinfowler.com/articles/microservices.html [↑](#footnote-ref-6)