# Exploring the Microservice Architecture to develop a flexible web-service of some description

Jonathan James Mitchell
- 40311730

Submitted in partial fulfilment of
the requirements of Edinburgh Napier University
for the Degree of
BEng (Hons) Software Engineering

School of Computing

September 25, 2018

**Authorship Declaration**

I, Jonathan James Mitchell, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained informed consent from all people I have involved in the work in this dissertation following the School's ethical guidelines.

*Signed: J.Mitchell*

*Date: September 25, 2018*

*Matriculation no: 40311730*

**Data Protection Declaration**

Under the 1998 Data Protection Act, The University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below one of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

The University may not make this dissertation available to others.

**Abstract**

## Contents

## List of Tables

## List of Figures

**Acknowledgements**

Insert acknowledgements here

I would like to thank my Degus: Arthur and Napoleon. My parents, friends, family and other peoples. Give Xiaodong Liu a mention here too!

# 1 Chapter 1: Introduction

## 1.1 Background

Traditionally, software applications/systems are developed using the Monolithic Architecture – a unified model of the design of a software program [4]. They are designed to be self-contained: components of the program are interconnected and interdependent. For this tightly coupled architecture to work, each component and dependent component's must be present for the code to compiled or executed. If any component is missing, there is a high chance the program will not work correctly. The rapid increase of software systems has changed the shape of how businesses function in today's world. As businesses have expanded and added new products/services, problems arose where the Monolithic code base would become difficult to maintain and any future changes could result in problems arising when implemented. For the past thirty years, the software industry has been moving every closer to a service-orientated approach [3]. This evolution has resulted in a closer bond between businesses and IT. Instead of businesses making decisions controlled or constrained by software, they now make decision supported by software.

Service Orientated Architectures (SOA) were the first realisation of transforming monolithic systems into small building blocks – components – that work together to create applications that are easier to maintain and expand upon. A service is defined as: a function that is well-defined, self-contained, and does not depend on the context or state of other services [1]. Around seven years ago, at a workshop of architects in Venice, the participants saw a common architectural style they had all been recently exploring. The term "Microservices" was created [2]. It describes a particular way of designing software applications as suites of independently deployable services. That can be maintained and modified effectively to meet the demands of the business world in the present day.

By describing each function, of a software system, as an individual service; they can be separated into their own components, contained within their own domain model. Doing this entails each service has a single responsibility to perform and no more. This would also entail each service to have its own database with which to manipulate data. There would be no shared relational database for the entire system. Although there may be a need for different services to communicate with each if they share specific information in order to maintain the validity of data their databases. The development of containerised software has increased the capability of polyglot programming. Were each Microservice can be written in a different language, and when deployed via containers, they combine, like building blocks, to form the software system. This form of modularity allows for each Microservice to function independently and improves the stability of the system as a whole - if one Microservice was to fail, it would not result in the entire system from failing.

With the recent boom of the internet, particularly in the last decade, Microservices are beginning to be used by retail companies with a large online presence such as: Amazon, Netflix and Ebay. Although retail is not the only sector to use this architecture: Uber, the guardian and Capital One. As the internet has proven to be an excellent platform to deploy systems. More specifically web-based systems.

This project investigates and evaluates: designs, principles and technologies related to Microservices used too implement this architecture in developing an E-Commerce web system.

## 1.2 Motivation

why are you doing this? needed?

## 1.3 Aims, Objectives and Scope

The aim of this project is to explore how a microservice architecture (MSA) can be used to develop a flexible web based E-commerce system. Finding out the flexibility such an architecture provides and how this can be used to allow developers too maintain/update such systems in the fast-paced environment of today's world. This project will concentrate on developing a web-based system as the internet has been one of the fastest growing resources available to businesses to expand and meet the ever growing demands of customers.

These aims are achieved by examining the literature available on Microservices, the principles involved to enforce the Microservice Architecture and the technologies available to create this architecture for developing software to define research questions. A software development environment is then chosen to implement this architecture in a prototype system. Literature research will include: academic white papers, professional journals, lectures, on-line articles and software books. The collected/researched information will be used to answer the questions and provide the necessary understanding to develop a prototype using the microservice architecture. Following this, the project is evaluated, considering the original aims and objectives.

As MSA is a relatively new architecture; there is still a broad view for successful implementation. Due to time and cost constraints, this project will focus on web-based design patterns and protocols. As well as software and hardware to design a web system. There will be a limit to the exploration of applicable principles, design patterns, hardware and software available that cna be used to implement a microservice architecture.

## 1.4 Outline

This dissertation is structured as follows:

- **Chapter 1: Introduction**
  Introduces the topic of microservices, names the aims and objectives and outlines the scope and constraints for this project.
- **Chapter 2: Literature Review**
  Encompasses all the subjects and terms that are related/representative of a microservice architecture. It will go into detail about why Microservices are being used, why it is preferred over Service-orientated architecture and the technologies currently in use and those being developed. This chapter will also provide a critical and objective analysis of these subjects.
- **Chapter 3: Project Planning**
  Provides an overview of the management of the project. With use of a Gantt chart. The methodologies used during the development cycle will also be described. With justification. This chapter will also include a description of the high level functionality needed for the project using the project management tool MoSCoW.
- **Chapter 4: Prototype**
  This chapter contains the following sections:
  - **Analysis**
    This chapter will describe the requirements for the development of a prototype. And an analysis of the high-level design of the prototype. Including a diagram of the service interaction.
  - **Design**
    This chapter will provide a detailed design of the structure of the prototype – UI designs, Class diagrams, E-R models etc.
  - **Implementation**
    This chapter will detail the actual development of the prototype. Each staged will be documented. All issues/problems will also be documented along with the solutions. Also, providing a reference for all sources used. Screen shots of the development stages will be provided via appendices.
  - **Testing**
    This chapter will consist of all testing documentation and testing conducted on the prototype. With a description of the testing methodologies used.

  - **Evaluation**
    This chapter will provide a critical and objective analysis of the developed prototype. Providing a detailed description of its success/failure.

- Chapter 5: Conclusions
  Discusses the project, how effective Microservices architecture is etc. And going forward what further research etc. will be done/required.
- Chapter 6: References
  Contains all references used throughout this document.
- Chapter 7: Appendices
  Houses all the appendices. This includes Gantt charts, UML diagrams, testing documentation and screen shots of: development progress at various stages and working prototype.

## 1.5 Summary

This project encompasses the following principles, design patterns and technology:

- principle 1
- principle 2
- principle 3
- Design pattern 1
- Design pattern 2
- Design pattern 3

Being a prototype, and a university project, the system will not implement a payment system nor will the login/authentication microservice require any more than a login and password. No personal details will be requested.

# 2 Chapter 2: Literary Review

## 2.1 Microservice

blah

## 2.2 Domain Driven Design

blah

## 2.3 Single-Responsibility Principle

blah

## 2.4 RESTful

blah

### 2.4.1 Communication

blah blah

### 2.4.2 API's

blah blah

### 2.4.3 CRUD Service

blah blah

## 2.5 polyglot Programming

blah

## References

[1] Douglas K. Barry. Service architecture, 2000.

[2] Martine Fowler. Microservices, 2014.

[3] Judith Hurwitz, Robin Bloor, Marcia Kaufman, and Fern Halper. *Service Orientated Architecture for dummies*. Wiley, 2 edition, 2009.

[4] Margaret Rouse. Monolithic architecture, 2016.

# Appendices

## A Project Overview

### A.A Example sub appendices

...

## B Second Formal Review Output

Insert a copy of the project review form you were given at the end of the review by the second marker

## C Diary Sheets (or other project management evidence)

Insert diary sheets here together with any project management plan you have

## D Appendix 4 and following

insert content here and for each of the other appendices, the title may be just on a page by itself, the pages of the appendices are not numbered, unless an included document such as a user manual or design document is itself pager numbered.