

EVALUATION OF MICROSERVICE ARCHITECTURE PATTERNS

R.P.D.M Weerasooriya

(209393E)

Degree of Master of Science in Computer Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

July 2022

EVALUATION OF MICROSERVICE ARCHITECTURE PATTERNS

R.P.D.M Weerasooriya

(209393E)

Thesis/Dissertation was submitted in partial fulfillment of the requirements for the degree
MSc in Computer Science specializing in Software Architecture

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

July 2022

Declaration

I declare that this is my own work and this thesis/dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: *UOM Verified Signature*

Date: 2022-07-22

The above candidate has carried out research for the PhD/MPhil/Masters thesis/dissertation under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of the supervisor: Prof. Indika Perera

Signature of the supervisor: *UOM Verified Signature*

Abstract

Microservices has been a very popular word in the Software Industry for quite a long time. Microservices architecture is regarded as a rising trend. There has been a rise in the research carried out in the field of microservices which encourage enterprise software architects and IT executives to witness and be a part of the new evolution. Most of the time we might only have a glance at such topics when our attention is caught. The concept of microservices describes a style of software systems that is highly effective in building enterprise solutions in the current times. The software industry has witnessed that there have been many companies such as Netflix, Amazon, Spotify who have benefited greatly with the use of microservices. So that for many other software organizations are rapidly adhering to incorporating microservices into their enterprise solutions. This is becoming the first choice style for building enterprise applications. But, however, there's not much guidance or information which will help a beginner to determine what the microservice style should be used in the project and how to do it. In order to fill this gap, and be of assistance for architects and developers to identify the most appropriate patterns which most suit their enterprise application, I aim to carry out my research targeted to find out and characterize various microservice architecture patterns reported in the known literature, and perform an evaluation of architectural patterns by case studies and with implementations[1].

ACKNOWLEDGEMENT

First and foremost, I would like to express my gratitude to my supervisor, Dr. Indika Perera for providing abundant guidance, support and encouragement throughout this research. Also I would like to thank my colleagues for sharing knowledge, support and constant encouragement. Last but not least, I express my love and gratitude to my parents for their love, help and support.

Thank you

Table of Contents

Abstract	ii
ACKNOWLEDGEMENT	iii
Table of Figures	vi
LIST OF ABBREVIATIONS	vii
1. Introduction	1
1.2 Motivation for the Research	3
1.3 Research Problem	4
1.4 Research Objective	5
1.7 Organization of the Thesis	6
2. Literature Review	7
2.1 Different Architectural Patterns	7
2.2 The Coordination Patterns in Microservices	7
2.4 Deployment Strategies & Patterns	7
2.5. Data Storage Patterns	8
3. Methodology	8
3.1.1 Pattern Identification	9
3.2 Implementation	10
3.2.1 Client Side Discovery Pattern	10
3.2.2 Server side discovery pattern	11
3.2.3 API Gateway Pattern	12
3.2.3 State management Pattern	13
3.2.4 Deployment patterns	14
4. Evaluation	16
4.1 The Coordination Patterns in Microservices	16
4.1.1 Fine Grained SOA	16
4.1.2 Service Discovery Patterns	18
4.1.2.1 The Client Side Discovery Pattern	19
4.1.2.2 The Server Side Discovery Pattern	20

The Service Registry	21
4.1.3 The API-Gateway Pattern	22
Advantages	23
Disadvantages	23
4.1.4 Backends for frontends	24
Advantages Of BFF	25
Disadvantages of BFF Pattern	25
When to use BFF	27
Solve BFF Associated Problems	27
4.2 Managed State Patterns	29
Advantages	31
Disadvantages	31
4.2.2 Event-Driven State Management	32
4.2.2.1 Saga Pattern	33
Advantages	34
Disadvantages	34
4.2.3 (Event Sourcing) Replicating State in Layered APIs	35
4.3 Deployment Patterns	38
4.3.1 The Multiple Services per Host Pattern.	39
4.3.2 Single Service per Host Pattern	39
4.3.2.1 Service Instance per Virtual Machine Pattern	40
4.3.2.2 Service Instance per Container Pattern	40
4.3.3 Serverless Deployment	41
Single Service per host	43
Serverless	43
4.4. Data Storage Patterns	43
4.4.1 The Database-per-Service Pattern.	43
4.4.2 The Database Cluster Pattern.	45
Database Cluster Pattern	46
5. Conclusion	46
Future Work	48
REFERENCES	49

Table of Figures

Figure	Page Number
1 - Implementation of Client Side Discovery	18
2 - Client-side service discovery config	18
3 - Client side delivery	19
4 - Server Side Discovery Pattern	20
5 - API Gateway Pattern	21
6- Dockerizing an application	22
7 - Serverless applications	23
8 - Overview of Fine Grained SOA	25
9 - Overview of Service Discovery	27
10 - Client Side Discovery Pattern	29
11 - Server Side Discovery Pattern	29
12 - Overview of API Gateway	31
13 - Default Api Gateway Pattern	33
14 - Fan out	35

15 - Fuse	36
16 - Solve Fanout	37
17 - Message Oriented State Patterns	41
18 - Event driven state management	43
19 - Event Sourcing	46
20 Multiple Services Per Host Pattern	50
21 Single Service Per Host Pattern	51
23 Serverless Architecture Overview	52
24 Database per service pattern	56
25 Shared Database Pattern	57

LIST OF ABBREVIATIONS

- API - Application Programming Interface
- REST - Representational State Transfer
- SQL - Structured Query Language
- NoSql - Not Only SQL
- RDBMS - Relational Database Management Section
- HTTP - Hypertext Transfer Protocol
- SOA - Service Oriented Architecture
- GCP - Google Cloud Platform