

A COMMON TRANSACTIONAL ARCHITECTURE FOR MICROSERVICES WITH OPEN API SPECIFICATION

Nambukara Gamage Nipun Thilakshan

(219414G)

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

July 2023

A COMMON TRANSACTIONAL ARCHITECTURE FOR MICROSERVICES WITH OPEN API SPECIFICATION

Nambukara Gamage Nipun Thilakshan

(219414G)

Thesis/Dissertation submitted in partial fulfillment of the requirements for the degree
Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

July 2023

DECLARATION

I declare that this is my own work, and this 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.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date: 2023/07/27

The above candidate has carried out research for the Masters dissertation under my supervision.

Name of the supervisor: Prof. Indika Perera

Signature of the supervisor:

Date 27/07/2023

ACKNOWLEDGEMENT

I would like to show my gratitude to Prof. Indika Perera for guiding me to initiate and finding a better methodology to conduct the research. His supervision greatly helped me in setting goals and engaging in the research.

I would like to express my deepest gratitude to the Department of Computer Science and Engineering at the University of Moratuwa for providing the support necessary to overcome this endeavor.

Last but not least, my greatest gratitude goes to my parents and friends who supported me throughout this effort.

ABSTRACT

The domain of Information Systems (IS) is no longer isolated in modern world, with the high-speed network facilities more applications and sub systems such as management, EDI, CRM are being developed which access different independent sources inside and outside the organization. Nowadays, the microservice architecture is widely utilized in the development industry for creating larger and more intricate applications. A good microservice architecture helps to maintain the interoperability between different Opcos by reducing the cost of interconnectivity.

The existing gap in the telecom industry is not having a common standard set of APIs when designing their internal systems which leads to more labor cost and time consuming when integrating them with existing vendors or systems. Telecommunication companies have already identified the importance of having common API sets in their systems. Due to the complexity of those APIs, it will be difficult in catering existing requirements.

Several studies have been carried out on similar areas such as blockchain management, e-government portals and event driven architectures on designing an interoperable architecture for dynamic business models. However, most of them are analyzed in architectural level but not in microservice or API level with compatible data model and none of them have shown promising results.

This research attempts to explore the importance of an interoperable transaction architecture for microservices by analyzing TM Forum open API specification architecture along with its data model. It lends to propose a microservice based reference implementation and a reference architecture for community wide domains.

Key words: TMF, Microservices, API

TABLE OF CONTENTS

Declaration	i
Acknowledgement	ii
Abstract	iii
Table of Contents	iv
List of Figures	vii
List of Tables	ix
List of Abbreviations	x
List of Appendices	xi
1. Introduction	1
1.1 Research Problem and Motivation	2
1.2 Research Gap	4
1.3 Research Objectives	5
1.4 Outline	5
2. Literature review	6
2.1 Interoperability	6
2.1.1 Importance of Interoperability	7
2.2 Complications in telecommunication industry	7
2.3 Importance of APIs in the telecommunication industry	8
2.4 Existing Telco Core Systems Which Use APIs	9
2.4.1 CRM systems	9
2.4.2 Order Management Systems	9
2.4.3 Billing systems	10
2.4.4 VAS systems	11
2.4.5 SMSC systems	11

2.5	Why interoperable common model is required for telecom industry.....	12
2.6	TMF Open APIs	12
2.7	Studies carried out on designing interoperable system architectures.....	14
2.8	Studies carried out on TMF Open APIs	17
2.8.1	TMF Open APIs with Database models	18
2.9	Relational Databases	20
2.10	NoSQL Databases.....	20
2.11	Hybrid Databases.....	21
3.	Methodology	23
3.1	Software Requirement Specification.....	24
3.1.1	Overall Architecture.....	24
3.1.2	Stakeholders	25
3.1.3	Finding from requirement elicitation	27
3.2	Context diagram	29
3.3	Use cases	30
3.3.1	Use Case Descriptions	31
3.4	Functional requirements	35
3.5	Non-Functional requirements.....	36
3.6	System Architecture and Design	37
3.6.1	Design Goals	37
3.6.2	Layered Architecture/Tiered Architecture	37
3.6.3	Component Diagram	39
3.6.4	Class Diagram	40
3.6.5	Sequence Diagrams.....	41
3.6.6	Process flow chart	44
3.7	How the new data model speeds up the development?.....	47

3.8	Query data with flexibility	48
4.	Evaluation	50
4.1	System testing	50
4.1.1	Unit Testing	50
4.1.2	Integration Testing	50
4.1.3	Functional testing	51
4.1.4	Non-functional testing	52
4.1.5	User interfaces.....	53
4.1.6	Limitations in testing	55
4.2	System design and architecture of the solution.....	55
4.3	Query customers by any combination of features.....	56
4.4	Create a customer profile	56
4.5	Update a customer profile	57
4.6	Delete a customer profile	57
4.7	Non-functional requirements.....	57
5.	Conclusions.....	65
5.1	Summary	65
5.2	Research limitations	66
5.3	Future work	66
	References.....	68

LIST OF FIGURES

Figure 1-1: Loosely coupled components in microservices [1].....	2
Figure 1-2 Rich picture of proposed solution	5
Figure 2-1: Basic Modules of a telecom operator system (Fig. 1, (Olga, 2020))	10
Figure 2-2: Scope of TM forum APIs.....	13
Figure 2-3: Scope of API re-use and contribution	14
Figure 2-4: TM Forum Open Digital Architecture [21].....	14
Figure 2-5: The extended LCIM [24]	15
Figure 2-6: Tecnotree digital architecture.....	16
Figure 2-7: Concept Map	22
Figure 3-1: Overall view of solution.....	24
Figure 3-2: Stakeholder over	27
Figure 3-3: Context diagram	29
Figure 3-4: Use case diagram.....	30
Figure 3-5: Architecture Diagram.....	39
Figure 3-6: Component diagram of implemented solution.....	40
Figure 3-7: Class diagram.....	41
Figure 3-8: Sequence diagram of creating new customer.....	42
Figure 3-9: Sequence diagram of updating an existing customer.....	42
Figure 3-10: Sequence diagram of querying customers.....	43
Figure 3-11: Sequence diagram of delete a customer	44
Figure 3-12: Process flow chart of delete customer.....	44
Figure 3-13: Process flow chart of create customer.....	45
Figure 3-14: Process flow chart of update customer.....	45
Figure 3-15: Process flow chart of query all customers.....	46
Figure 3-16: Relational Representation	47
Figure 3-17: MongoDB Representation.....	48
Figure 3-18: GET operation implementation.....	49
Figure 4-1: application.property file of tmfapi-customermanagement module.....	50
Figure 4-2: Docker file and docker image of tmfapi-customermanagement module.....	51
Figure 4-3: View all customers user interface	53

Figure 4-4: Find one customers user interface.....	53
Figure 4-5: Update successful message user interface.....	54
Figure 4-6: Delete confirmation message user interface	54
Figure 4-7: Create customer user interface	55
Figure 4-8 : Summary of Request Hits	57
Figure 4-9 : Latency Report	57
Figure 4-10 : Summary of Error Results.....	58
Figure 4-11 : Percentage of Time Spent	58
Figure 4-12 : Network bytes sent vs Network bytes received	58
Figure 4-13: Summary of Network Throughput	58
Figure 4-14: Query Customer API (Simple	59
Figure 4-15: Query Customer API (Simple	59
Figure 4-16: Query Customer API (Complex.....	60
Figure 4-17: Query Customer API (Complex Query to get all fields) - Throughput vs Number of Request	60
Figure 4-18: Query Customer API (Simple Query to get few fields) - Average Response time vs number of requests.....	61
Figure 4-19: Query Customer API (Simple Query to get all fields) - Average Response time vs number of requests.	61
Figure 4-20: Query Customer API(Complex Query to get few fields) - Average Response Time vs Number of Request.....	62
Figure 4-21: Query Customer API (Complex Query to get all fields) - Average Response Time vs Number of Request.....	62
Figure 4-22: Create Customer API - Throughput vs Number of Request	63
Figure 4-23: Create Customer API - Avg Response Time vs Number of Request ...	63
Figure 4-24: Update Customer API - Throughput vs Number of Request	64
Figure 4-25: Update Customer API - Average Response Time against Number of Request.....	64
Figure 5-1: Component diagram of solution that can be implemented to cater all TMF APIs.....	67

LIST OF TABLES

Table 3-1: Selected methodologies list	23
Table 3-2: Selected development strategies.....	24
Table 3-3: Stakeholders and their viewpoint	25
Table 3-4: Functional requirements priority and mapping with use cases	35
Table 3-5: Nonfunctional requirements	36
Table 3-6: Design goals for the system.....	37
Table 4-1: Functional Testing results	51

LIST OF ABBREVIATIONS

Abbreviation	Description
3GPP	3rd Generation Partnership Project
API	Application Programming Interface
BSS	Business Support Systems
CBS	Customer Billing System
CCRM	Customer - Centric Relationship Management
CDR	Call Detail Record
CRM	Customer Relationship Management
CRUD	Create, Read, Update, Delete
DBMS	Database Management System
GCP	Google Cloud Platform
HTTP	HyperText Transfer Protocol
JSON	JavaScript Object Notation
JVM	Java Virtual Machine
KPI	Key Performance Indicator
LTE	Long Term Evaluation
OCCI	Open Cloud Computing Interface
ONAP	Open Network Automation Platform
PHP	Hypertext Preprocessor
PID	Project Initiation Document
REST	Representational State Transfer
SOAP	Simple Object Access Protocol
SMSC	SMS Centre
SNMP	Simple Network Management Protocol
TRC	Telecommunications Regulatory Commission
XML	Extensible Markup Language
WSDL	Web Service Definition Language
URI	Uniform Resource Identifier

LIST OF APPENDICES

Appendix	Description	Page
Appendix - A	Unit tests written for the Service Layer Functions	72
Appendix - B	Unit Test Report	74
Appendix - C	Integration Test Results	75