MICRO SERVICES ARCHITECTURE FOR DISASTER MANAGEMENT SYSTEMS

Makkia Abbasi

Enrollment No: 01-244152-053

A thesis submitted to Department of Software Engineering, Bahria University, Islamabad in the partial fulfillment for the requirements of a Masters degree in Software Engineering

October 2018

Approval Sheet

Thesis Completion Certificate

Scholar's Name:	Makkia Abbasi	Registration No:	43393
Programme of Study:	MS Software Engineering		
Thesis Title:	Micro Services Architecture for Disaster Management Systems		

It is to certify that the above student's thesis has been completed to my satisfaction and, to my belief, its standard is appropriate for submission for Evaluation. I have also conducted plagiarism test of this thesis using HEC prescribed software and found similarity index at 9% that is within the permissible limit set by the HEC for the MS/MPhil degree thesis. I have also found the thesis in a format recognized by the BU for the MS/MPhil thesis.

Principal Supervisor's Signature: _____

Date: _____ Name: Dr. Awais Majeed

CERTIFICATE OF ORIGINALITY

This is certify that the intellectual contents of the thesis Micro services Architecture for disaster Management Systems, are the product of my own research work except, as cited property and accurately in the acknowledgements and references, the material taken from such sources as research journals, books, internet, etc. solely to support, elaborate, compare and extend the earlier work. Further, this work has not been submitted by me previously for any degree, nor it shall be submitted by me in the future for obtaining any degree from this University, or any other university or institution. The incorrectness of this information, if proved at any stage, shall authorities the University to cancel my degree.

Signature: _____ Date: _____

Name of the Research Student: Makkia Abbasi

ABSTRACT

Disaster is a big challenge that can occurs at any time anywhere in the world. Disaster management various activities involving disaster monitoring, early disaster detection, generate alert signals etc. Collaboration among various stakeholders is the key to successful relief and rescue operations which can be enhanced with the use of Information & Communication Technologies (ICT). Information needs to be gathered from multiple sources like sensors, multiple IoT devices, smart mobile devices and Information Systems. Often such information gathering and dissemination needs to be carried out spontaneously and on ad-hoc bases due to volatility and availability of information sources and information targets. Thus scalable and easily configurable information architecture is required. We propose micro-services architecture to overcome this problem that is more scalable, less expensive, loosely coupled and language independent. Micro service is a new term that is most widely using in multiple applications that is a subset of SOA (Service Oriented Architecture). We have developed a prototype implementation for disaster management using a micro services, this prototype is used to develop an application based on micro services architecture for disaster management. We have proposed architecture to build such system. After that, we have measured the performance of both architectures to find out which one is a best way calling micro-services through mobile interface. There is not much difference but API gateway is important for decoupling.

DEDICATION

To my whole family for their love and support

ACKNOWLEDGMENT

All praise belongs to **Allah** Almighty who gave me the strength; courage and knowledge in completing this research .I owe my deepest gratitude to my supervisor **Dr. Awais Majeed**. I am very fortunate to have his outstanding guidance, generous support and constant encouragement that has enlightened me at every step of my research work.

Table of Contents

ABSTRACT 1
DEDICATION 2
ACKNOWLEDGMENT 3
List of Figures 7
List of Tables 9
Chapter 1 10
Introduction 10
1.1. Overview: 10
1.2. Micro-services overview: 11
1.3. Motivation: 11
1.4. Scope: 12
1.5. Problem Statement: 12
1.6. Research Questions: 12
1.7. Research Objectives: 13
1.8. Structure of Thesis: 13
Chapter 2 15
Literature Review 15
2.1. Introduction: 15
2.2. Issues with monolithic: 20
2.3. Comparison of Existing approaches: 21
2.4. Conclusion: 23
Chapter 3 24
Proposed Methodology 24
3.1. Research Methodology: 24
3.2. Proposed Conceptual Model: 25
3.3. Disaster Scenario: 27
3.3.1. Transportation Micro-service: 27
3.3.2. Emergency Notification Service: 27
3.3.3. Sensory Information Service: 28
3.3.4. Food and shelter service: 28

3.3.5. Evacuation plan service: 29
3.3.6. Nearest Rescue Center services: 29
3.3.7. Emergency Contacts Numbers: 29
3.3.8. Send Response Micro-service: 29
3.4. Conclusion: 30
Chapter 4 31
Proof of Concept and Results 31
4.1. Overview: 31
4.2. A Case Study of the disaster management, mobile application and validation process: 31
4.2.1. A typical Disaster Management Scenario: 31
4.2.2. Micro-service Development Lifecycle: 32
4.3. SOAP vs. REST: 34
4.4. How Micro-services work: 34
4.4.1 Service Interface Diagram: 35
4.5. Implementation of Proposed Micro-service Architecture: 37
4.5.1. Implementation Technologies: 38
4.5.2. MYSQL: 39
4.5.3. Internet of Things Platform: 39
4.6. Implementation/Simulation of Micro-services on Node-Red: 39
4.6.1. Transportation Micro-service Flow: 39
4.6.1.1. Display of UI-Template Node: 40
4.6.2. Evacuation Plan Micro-Service Flow: 41
4.6.2.1. Display of UI Template Node: 42
4.6.3. Emergency Notification Micro-service Flow: 42
4.6.4. Emergency Contact Numbers Micro-service Flow: 43
4.6.4.1. Display the Emergency Contacts: 43
4.6.5. Food and Shelter Micro-Service Flow: 44
4.6.5.1. Display of UI-template: 45
4.6.6. Nearest Rescue Center Micro-service Flow: 45
4.6.6.1. Display on Map: 46
4.6.7. Sensory Information Micro-service Flow: 46
4.6.7.1. Payload Results of the Temperature Sensors: 47

4.6.8. Main Flow for Mobile Application without Gateway: 48
4.6.8.1. Interface of Mobile Application: 48
4.7. Flow of Calling Micro-services through API Gateway: 49
4.8. Main Flow for Mobile Application Interface with Gateway: 49
4.8.1. Mobile Interface Using API gateway: 49
4.9. Comparison between the Architectures using Gateway and without Gateway: 50
4.9.1. Flow with API Gateway: 50
4.9.2. Performance Results with Gateway: 51
4.9.3. Flow without API Gateway: 51
4.9.4. Performance Results without Gateway: 52
4.10. Comparison between performances of both flows: 52
4.11. Comparison between Alternate proposed Disasters based architectures and our propose micro-services based architecture: 53
4.12. Conclusion: 54
CHAPTER 6 56
Conclusion 56
6.1: Contributions: 56
5.2. Reflections: 57
5.3. Future Work: 57
References: 58