# Software Design Document

## DXC Technology Managed Services For Workloads On Public Cloud


SE 4485.001

Dr. Weichen Wong

***Group 2:***

*Ihfaz Tajwar*

*Zunayed Siddiqui*

*Po-Yu Liu*

*Yeswanth Bogireddy*

*Zachary Tarell*

# Abstract

Service providers are constantly challenged to accept customers' workload as-is and manage it on behalf of the customer. The management can range from monitoring, patching, and security aspects of the workload.

- The security could be shut off certain ports or setup security scans for Denial of Service attack, etc.
- The monitoring could leverage the cloud provider monitoring but would alert based on certain rules
- The patching could use cloud providers specific to patch from vulnerabilities, routine service packs, etc.

Initially, the workloads may be untouched as in refactored (7 R's) to reduce the timelines but over time it might get refactored over time to reduce costs or increase stability/performance

# Table of Contents

# List of Figures



# List of Tables



# Introduction

Our team has been tasked with implementing a makeManaged script for DXC Technology to monitor, make patchwork, and manage security aspects of their workload. This project management plan will include all of the aspects from beginning to end on how our team will create and implement DXC Technology's goal from our script. Even though the requirements list out three different phases of this project, our sponsor has made clear that the minimum viable product will only need to contain the monitoring aspect of our project and if time permits move onto patchwork as well as security, load balancing, and firewall monitoring.

## Purpose and Scope

The purpose of monitoring software through Infrastructure as a System (IaaS) is to benefit many different companies that require help desk support, patchwork, and security. Since SaaS and PaaS are already managed by cloud providers, we will only be using IaaS. The scope will range from a working monitoring script able to communicate with a cloud service and relay information, notifications, and patchwork to DXC Technology. This will lighten the workload of their employees by automating a lot of tasks and opening up free time to allocate resources elsewhere.

## Overview

A brief overview of the makeManaged program will be to use a virtual machine (VM) to communicate to AWS, Azure, and/or GCP cloud services over IaaS to monitor and the help desk and send notifications and email alerts or patch any necessary fixes.

# Detailed Design Architecture

## GUI (Graphical User Interface) Design

When talking to our sponsor, Chandra, and as we monitored our process and established a better understanding of our goals in this project; we and Chandra decided a GUI was not needed for this project.

## Static Model - Class Diagrams

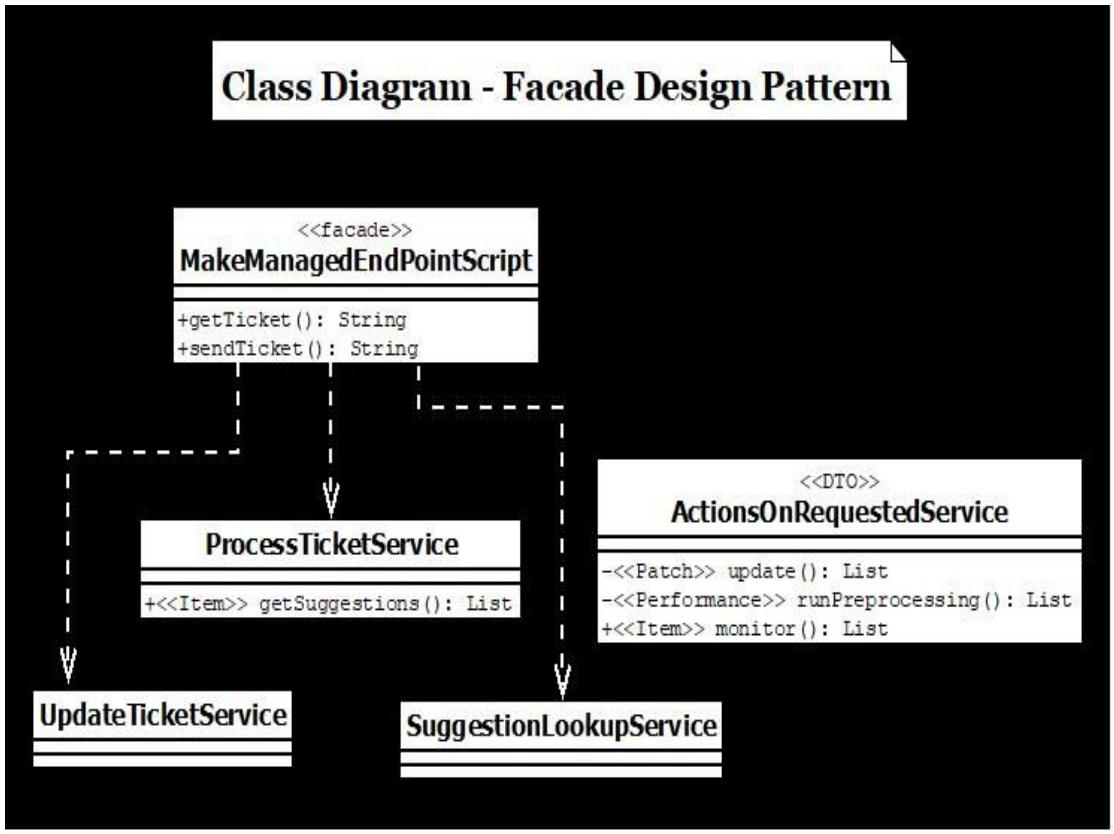


*Figure 1*

## Dynamic Model - Sequence Diagrams

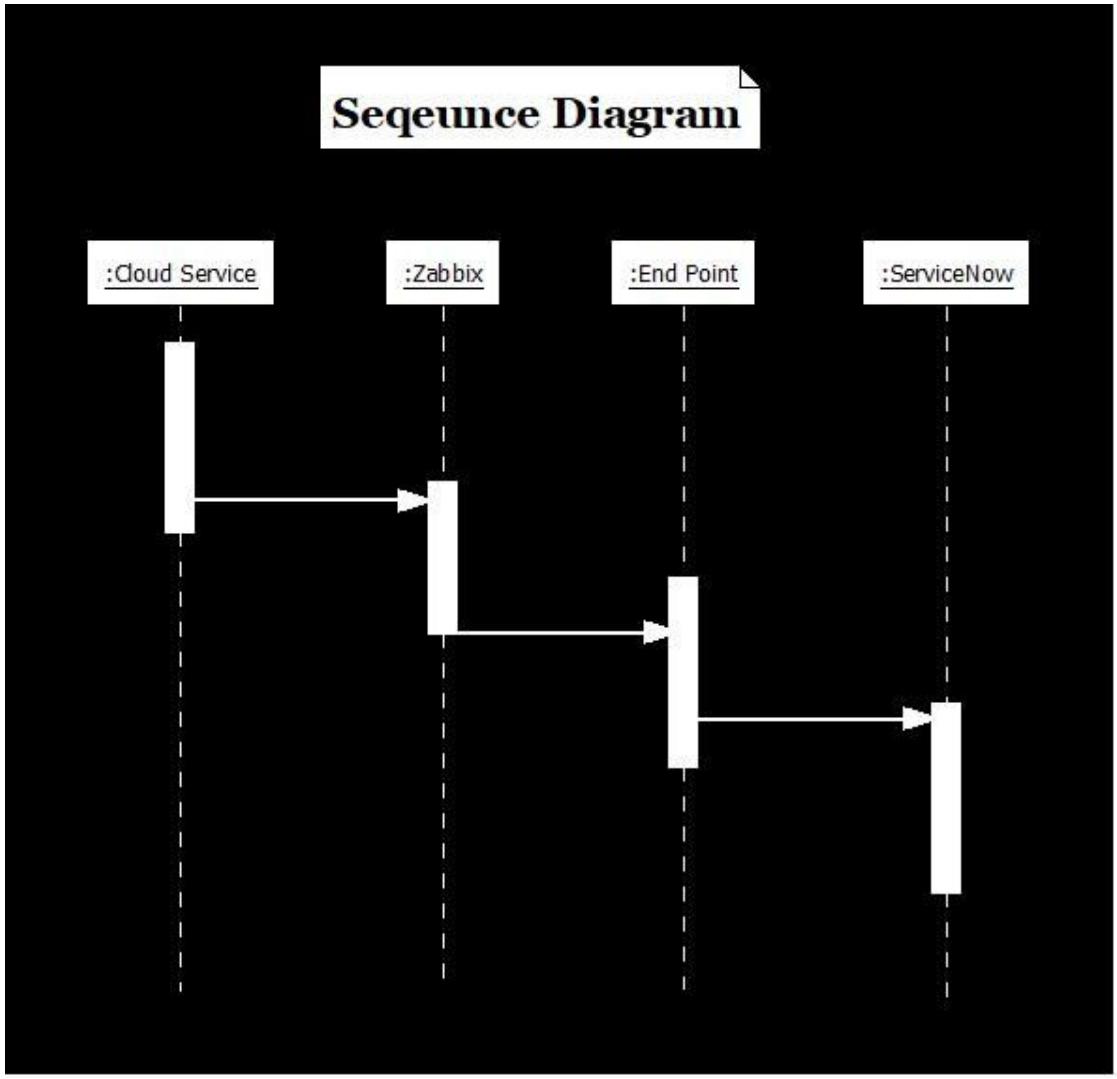


*Figure 2*

# Rationale for Design Model

Based on our conversation with our sponsor, we determined that this model would best fit our needs in monitoring the cloud providers. The client needs to be able to access the monitoring system which comes from our end point back to the cloud provider which sends out a call for a problem or issue that needs to be addressed. After more understanding and programming, we'll be able to update as needed to comply with our final design and architecture.

# Traceability from Requirements to Design

| Req ID | Req Description | Rationale | Type |
|---|---|---|---|
| 1 | The system should not have a response delay of more than 500 ms | Performance | NF |
| 2 | The system should be up for 99.5% of the time | Performance | NF |
| 3 | The system should be able to handle up to 5 triggers per minute | Performance | NF |
| 4 | The system should notify users within 100 ms of the trigger | Performance | NF |
| 5 | The system should create up to at least 5 tickets per second | Performance | NF |
| 6 | The system will be robust to fault tolerance and have a backup if there are crashes | Maintainability | NF |
| 7 | The system should make sure not to compromise the security of the users | Security | NF |
| 8 | The system should run within a certain operational cost threshold | Usability | NF |
| 9 | Client can use monitoring tool to monitor cloud provider | Client doesn't need to waste time on monitoring the cloud | F |
| 10 | Alert customer service with notifications (or patches) with a servicenow integration | Easier to free up other services instead of help desk | F |
| 11 | Client can use monitoring tool to monitor LAMP stack | Can write a script to send back to client | F |

***Table 1***

# Evidence Under Configuration Mgmt.

# References

"Software Architecture and Design" *Tutorials Point.* Web. 2020 https://www.tutorialspoint.com/software_architecture_design/introduction.htm

Sharma, Nishant. "Software Architectural Structures and Design Patterns". *Medium*. Web. July 25, 2019. https://medium.com/ios-expert-series-or-interview-series/software-architectural-patterns-design-structures-c5692fe8affc