8/21/2020

Patrick Kelly – x14533687

National College of ireland

Distributed Systems

TBAA Report

For my project, I was tasked with creating application that simulates a distributed systems environment based on the industry of Centralised Log Management.

3 Sufficiently Complex Services:

The 3 services that I have designed for this project are:

**System Logs:** This service allows the user to send multiple request which will allow the user to turn on, turn off, get status and print values from the printer.

**Event Logs:** This service allows the user to send multiple request which will allow the user to log in, log out, and open local applications on the computers system. When opening any application, the user must be logged in first for any functionality to be executed.

**Network Logs:** This service allows the user to send multiple request which will allow the user to check current temperature of the fridge, change temperature of the fridge, display a list of items current stored in the fridge and create a shopping list of items needed for the fridge

Use Of jmDNS:

When the server class starts up, it creates an instance of jmDNS is in which all the services available on it are then registered and then added to the server with the server’s information such as ServerType, ServerName and port also.

A screenshot of a social media post

Description automatically generated

Use of gRPC:

Each service was implemented using a proto file which contained all necessary fields such as messages and info on the types of rpc’s that would be called which would then generate the necessary stubs that would be then used to on the application.

A screenshot of a social media post

Description automatically generated

Remote Error Handling:

A series of try catch statements was used to look for all NullPointExceptions that might have occurred when any requests were being sent from the client to the server.

GUI:

A GUI for the application could not be made in time sadly by the project is visibly functional on BloomRPC where you can see a visual representation of the different rpc’s that are being called on the system and returning the requested information.

Github:

The final project is pushed to a public repository with the following link: <https://github.com/PaddyK95/DSTBAAproject>. The repo has a history of commits also.

Section (B) Related Questions:

Q1- The chosen service I have decided to implement as a publish-subscriber MQTT messaging protocol is System Logs as seen below:

The publisher which publishes a new topic to the server in which any subscribers can then view.

A screenshot of a cell phone

Description automatically generated

The subscriber which listens for any new publishes made on the server after it is connected.

A screenshot of a computer

Description automatically generated

There are 3 different types of QOS’s(Quality of Service) which specify the delivery quality of messages being sent

0 – Messages delivered at most once.

1. Messages may be delivered buy may be duplicated.
2. Once and once only delivery

Q3- Below is an implementation of one of my services with IDL(Interface Definition Language).

A screen shot of a social media post

Description automatically generated

An service that is created using IDLs consist of 3 main classes (Client, Server, Servant).

Client Class:

A screenshot of a cell phone

Description automatically generated

Server class:

A screenshot of a cell phone

Description automatically generated

And finally, the Servant Class:

A screenshot of a cell phone

Description automatically generated

Result:

A picture containing drawing

Description automatically generated

A picture containing bird

Description automatically generated

1. The purpose of an IDL is to describe the softwares API’s that are going to be used. It describes the interface as language independent which allows the communications of different components that are in different programming languages. It also specifies the different operations and fields that will be passed between the server and client.