



National
College^{of}
Ireland

Distributed Systems July 2024

Laura Carvalho Santana
Student ID - 23169931

Contents

1	Introduction	4
2	Service 1: MachinesControl.....	4
2.1	Methods.....	4
2.1.1	RPC Method 1 – MachineStatus	4
2.1.2	RPC Method 2 - TurnMachineON.....	4
2.1.3	RPC Method 3 – TurnALLMachinesOFF	5
3	Service 2: StaffStatus.....	5
3.1	Methods.....	5
3.1.1	RPC Method 1 – StaffStatus	5
3.1.2	RPC Method 2 - StaffHours	5
4	Service 3: DoorLocker.....	5
4.1.1	RPC Method 1 - DoorHistory	6
4.1.2	RPC Method 2 - DoorStatus	6

1 Introduction

This proposal outlines the guide for a Smart Factory, trying to optimize the control of services and daily activities inside the desire environment. Seeking an optimization of the activities and time.

2 Service 1: MachinesControl

With this service, the idea is to have the control of the machines. The client would be able to ask the machine status – if the machine are ON or OFF - and in both cases, the client would be able to turn the machine ON or OFF, changing the MachineStatus. The most important feature of this service is the EmergencyStop method when the client can ask for all the machines to be turned off in case an emergency is found, instead of having to do it manually with each machine individually.

The attributes are:

- MachineID;
- MachineType;
- MachineStatus (it can be on/off);
- MachineON;
- MachineOFF;

2.1 Methods

2.1.1 RPC Method 1 – MachineStatus ()

Using a Unary RPC Method. The client makes a call to this service asking for the MachineStatus, specifying the MachineID in the call.

The service gets back telling the MachineStatus returning the value in the real-time between the two options: MachineON or MachineOFF.

rpc SayStatus (MachineStatus) returns (MachineON/ MachineOFF) { }

2.1.2 RPC Method 2 – TurnMachineON ()

Using a Unary RPC Method. The client makes a call to the service asking to turn the MachineStatus to ON, specifying the MachineID in the call.

The service return MachineON.

2.1.3 RPC Method 3 – TurnALLMachinesOFF ()

Similar with the Method 2, Using a Unary RPC Method. The client makes a call to the service asking to turn the MachineStatus to OFF, selecting all machinesIDs.

The service return AllMachineOFF.

3 Service 2: StaffStatus

Whis this service, it would be able to keep a track on the StaffStatus. With the functions of ClockIn and ClockOut and keeping a track of the time for both functions.

It will also be able to keep a track of how many hours each Staff have done

The attributes are:

- StaffID;
- StaffStatus; (if the staff is “clock-in” or “clock-out”)
- time; (currently time for each activity)
- StaffHours; (the number of hours done by the day)
- StaffTotalHours (the number of hours done by the week)

3.1 Methods

3.1.1 RPC Method 1 – StaffStatus ()

Using a Unary RPC Method. The client makes a call to this service asking for the StaffStatus, specifying the StaffID in the call.

The service gets back telling the StaffStatus returning the value in the real-time between the two options: StaffIn or StaffOut.

3.1.2 RPC Method 2 - StaffHours ()

Using a Server streaming, the client makes a call to the server asking for StaffHours, specifying the StaffID.

The services get back telling the number of hours done by the staff in the day, calculating by the difference of time from the Clock-outa and Clock-in.

4 Service 3: DoorLocker

With this service, it will be possible to have a control of the access for different parts of the factory, controlling the doors locks. Each door would have a DoorID that will have different styles of controls, depending on the StaffLevel and the DoorLevel,

keeping the access restrict for different areas and also been able to lock or unlock any door remotely.

The attributes are:

- DoorID;
- DoorLevel; (will keep the control for each staff and who can get access to)
- StaffID;
- DoorStatus;
- DoorOpen;
- DoorClose; (even close, anyone can open)
- DoorLock; (only the staff with the right level/authorization can open)

4.1 [RPC Method 1 – DoorHistory \(\)](#)

Using a Server streaming, the client makes a call to the server asking for the DoorHistory, specifying the DoorID in the call.

The server gets back telling all the access that the specify door has in the certain amount of time, including the last one, showing the DoorLevel and the StaffIDs.

4.1.1 [RPC Method 2 – DoorStatus \(\)](#)

Using a Unary RPC Method. The client makes a call to this service asking for the DoorStatus, specifying the DoorID in the call.

The service gets back telling the DoorStatus returning the value in the real-time between the two options: DoorOpen, DoorClose or DoorLock.