

Distributed System

Smart Building  
  
Sebastian Konefal  
 Student ID: 21197458

Contents

[1 Introduction 4](#_Toc108383375)

[2 Service 1: Climate control 4](#_Toc108383376)

[2.1 Methods 4](#_Toc108383377)

[2.1.1 RPC Method 1 - DesiredHVAC 4](#_Toc108383378)

[2.1.2 RPC Method 2 - HVACstatus 4](#_Toc108383379)

[2.1.3 RPC Method 3 - RoomStatus 4](#_Toc108383380)

[3 Service 2: Desk Usage 5](#_Toc108383381)

[3.1 Methods 5](#_Toc108383382)

[3.1.1 RPC Method 1 - DeskStatusInquiry 5](#_Toc108383383)

[3.1.2 RPC Method 2 - DeskBooking 5](#_Toc108383384)

[4 Service 3: Cleaning heatmap 5](#_Toc108383385)

[4.1 Methods 5](#_Toc108383386)

[4.1.1 RPC Method 1 - EntersToToilet 5](#_Toc108383387)

[4.1.2 RPC Method 2 - ToiletStatusInquiry 5](#_Toc108383388)

[4.1.3 RPC Method 3 – UpdateToiletStatus 5](#_Toc108383389)

# Introduction

This proposal outlines gRPC system for a smart building solution in an office setting. The purpose of the system is to provide energy and cost efficient solution in management of the building resources while ensuring safe and comfortable environment for occupants.

Wide range of existing technologies can be implemented in a smart building and it is feasible to achieve automated control of the real-time functioning of a building thanks to the capability acquired by Internet of Things devices to gather, process, and analyse data from various elements and locations.

The below services may be offered as standalone packages and consist of services regarding climate control, desk occupancy and cleaning heatmap.

* Climate control service works with existing heating, ventilation, and air conditioning (HVAC) system to ensure that temperature and humidity are set to a desired level to ensure low operation costs in areas that are not utilised.
* Desk usage service ensures that the management is informed about the current usage of office desks in the building. This service is particularly important in the light of the working from home movement and it is estimated that approx. 30% of desks are not utilised throughout the day.
* Cleaning heatmap service provides monitoring of high traffic areas ensuring that cleaning of toilets is scheduled depending on the actual usage.

# Service 1: Climate control

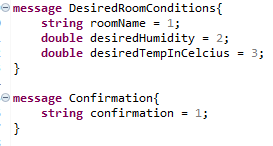
This service deals with adjusting heating, ventilation, and air conditioning system according to the desired settings, and current room conditions including temperature and humidity.

## Methods

### RPC Method 1 - DesiredHVAC

**Unary -** This method accepts **desired** values of temperature and humidity for each room in the building set by the system admin and saves the requested adjustment for that particular room. This value is later used as a check to calculate value in method 3. This method returns confirmation for the room that the conditions were set or informs that it was not found.

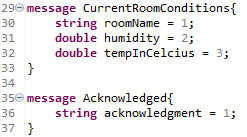




### RPC Method 2 - HVACstatus

**Client-Side Stream** - This method accepts stream of values of **current** temperature and humidity for a particular room in the building provided in real-time by sensors and stores the data for that particular room. This value is later used as a check to calculate value in method 3. This method returns confirmation for the room that the conditions were set or informs that it was not found.

****

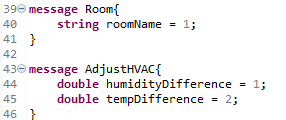
****

**rpc** HVACstatus(**stream** CurrentRoomConditions) **returns** (Acknowledged){};

### RPC Method 3 - RoomStatus

**Server -Side Stream** - This method accepts of name of a particular room in the building and calculates difference between values provided by method 1 (desired values) and 2 (real time values) and returns rounded to second decimal difference in temperature and humidity as a stream of 5 responses over 5 seconds (this might be utilised with a multithreading on the server so that it will accept up-to-date values while the stream is returned). This method returns values for the room or informs that it was not found.





# Service 2: Desk & Chair Usage

This service deals with adjusting desk and chair height to enable faciality of standing desk.

## Methods

### RPC Method 1 - DeskStatusHeight

**Unary Stream** - This method accepts value to adjust desk height (increment or decrement by 1) to enable a user transformation from sitting to standing desk and returns adjusted height. The method also performs a check whether the requested height is within the given range for that desk object (throws custom exception) or informs that the desk was not found.

### 

### 

### RPC Method 2 - ChairStatusHeight

**Unary** - This method accepts value to adjust chair height (increment or decrement by 1) to enable a user transformation from sitting to standing desk and returns adjusted height. The method also performs a check whether the requested height is within the given range for that chair object (throws custom exception) or informs that the desk was not found. The method uses Enums to check operation and custom message object in the request and response.





# Service 3: Cleaning heatmap

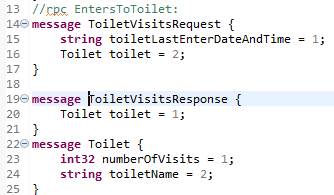
This service provides tracing service of toilets usage in real-time to enable more cost-efficient cleaning service. A cleaner will be able to ascertain, using a tablet, which toilets will require cleaning due to heavy traffic.

## Methods

### RPC Method 1 - EntersToToilet

**Unary** - This method sends date of last entry to a particular toilet in the building provided in real-time by sensors placed at the toilet entrances together with number of visits and name of the toilet (using custom object message). The method stores that data for that particular toilet or informs if the desk was not found in the database.





**rpc** EntersToToilet(ToiletVisitsRequest) **returns** (ToiletVisitsResponse){};

### RPC Method 3 – UpdateToiletStatus

**Bidirectional stream** - This method accepts stream of data regarding a toilet that was cleaned (custom object message (name and number of visits – i.e. set by the cleaner) and date), updates the database and value for that particular toilet requested and return all toilets that needs cleaning (custom object, date and Boolean to confirm that is was not updated and needs cleaning).



