Smart Room

09.02.2019

Shoaib Ahmed
Rahul Korthiwada
Sai Kumar Immadi

IIIT Guwahati

Vision

Smart Room System is designed with the objective of providing remote control of electrical appliances based on real-time sensing of room conditions. The users can interact with this cloud-based system through a web-based, real-time interface, which also shows the room conditions in real-time. The system uses Wi-fi for communication with the cloud.

Behaviour

The system can be operated in two modes :

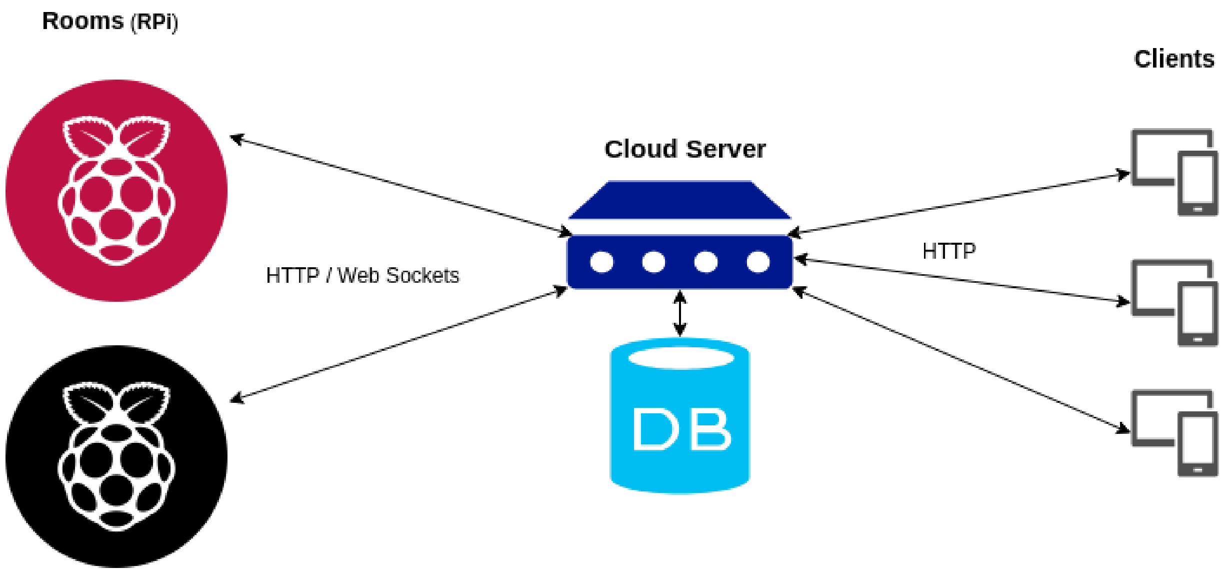
1. **Auto** : Feeling too lazy to toggle those switches everytime you enter or leave the room? We feel you. This mode allows True smart-management of your room wherein the sensors present in the room sense and redirect real-time sensor data like the light intensity, temperature, humidity, human presence etc. to the cloud and the cloud-based server does all the work for you; from controlling your fan speed to automatically switching off your appliances when nobody's around.

2. **Manual** : When you're bored of being taken care by the gizmos, show them who's the boss and take control of your home from your mobile. In this mode, the system does not take any control decisions and works solely on the input given by the user on the web-interface.

Management and Control

The application uses sensors and a Raspberry Pi to collect and send real-time room conditions to the cloud server. This is where the magic happens. The cloud server looks at the incoming data and accordingly manages the appliances in the room by sending back control information to the Raspberry Pi device. If the manual mode is chosen, whatever you choose is conveyed to the Raspberry Pi device.

The web-interface provided to the user can be used to control the appliances or can be used to monitor the room conditions.



Requirements

➔ Software

- ➔ **Node.js** : To run the whole system. Used on the Pi device as well as the cloud server and is the backbone of the whole system.
- ➔ **Socket.io** : To enable real-time Websockets-like data transfer to and from the Pi device. Powered by Node.js.
- ➔ **Cloud infrastructure** : To get the cloud server up and running with the database, client web application, HTTP server and Business Logic for Auto mode.

➔ Hardware

- ➔ **Raspberry Pi** : To collect and send sensor data to the server and to receive and execute control information from the cloud.
- ➔ **Sensors** : To sense room conditions like temperature, light intensity, humidity, human presence etc.
- ➔ **Wi-fi** : For the Pi to communicate with the server.