SAN JOSÉ STATE UNIVERSITY

Air Quality Index using Mobile Sensor Cloud

Presented by: Group 25

Aditya Dhende

Anuvrat Tiku

Hema Begur

Mohammad Awaise

Presented to:

Dr. Jerry Zeyu Gao

Professor – CMPE281

Cloud Technologies

Demo Link : https://vimeo.com/167075786

ABSTRACT

Sensor-Cloud is a new paradigm for cloud computing that uses the physical sensors to accumulate its data and transmit all sensor data into a cloud computing infrastructure. Sensor-Cloud handles sensor data efficiently, which is used for many monitoring applications.

The concept of Mobile Sensor Cloud Computing (MSCC) is to extend Sensor cloud-computing ecosystem to the world of future sensor enabled mobile applications and Internet clouds. Also it introduces new technologies, hardware, software, communication protocols, etc., which together forms ecosystem of mobile cloud. Smart world environment that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network. Wireless Sensor Networks (WSN) solutions, a growing green movement, demand for public safety solutions. WSN offer a powerful combination of distributed sensing, computing and communication. WSN have very high resource constraints in terms of memory, processing, and transmission power. They lend themselves to countless applications and, at the same time, offer numerous challenges due to their peculiarities. This paper includes an overview of wireless sensor networks, integration of sensor cloud computing services in smart city environment and accessible the same through sensor intended mobile. Also, it covers a comprehensive study regarding the requirements, different kind of well-known attacks and some of the proposed solution and model to counter the security attacks on WSN.

Table of Contents

1. Introduction

In various fields like medical, educational, environmental, and industrial we have seen the advent of use of wireless sensor network applications. These applications are inevitable for all these fields and many more. An environmental application consists of a typical sensor network, in which the sensors collect data from a particular region and the application working on this data. A wireless sensor network in an environmental set up, sensors can collect data related pollution of air, sound, temperature, earthquakes, forest fires, pressure, motion etc. Every node in these kind of networks is fitted with radio trans receiver or any other wireless communication device and is almost always powered by battery. Each node has cooperative abilities and they are present in random manner. Each node consists of three functionalities, first being fetching data, second processing this data and last communicating this data. Most commonly used sensors for example are camera sensors, traffic signal sensors, sensors related to microphone and heat detecting sensors.

Presently, wireless sensor networks are in use in varied areas like defence systems, medical fields, and military systems and patrolling services. They are used for seismic sensing analysis systems to detect earthquakes. They are used in many government and environmental systems like traffic control systems. Many problems can be solved when these sensors are connected together and when they send data together in a cooperative fashion. Nevertheless, these wireless sensor networks face lot of issues and problems with regards to their communication for example like short communication range, secure communication and privacy and reliability of data communicated, etc.) and resources (like power/battery, memory capacity, processing capacity, bandwidth, etc.). Further, wireless sensor networks have its own design limitations. These design issues are specific to the application and they are dependent on the environmental conditions. Depending on the environment, size of the network varies. If a small area has to be monitored, smaller number of nodes are needed to form a complete network but to cover a bigger area, large number of nodes are required. To monitor huge environment, there is a narrow communication between these sensor nodes, these communication challenges are caused to the obstacles in to the environment, these in turn affect the overall communication layer of network architecture. These challenges in wireless networks cause a dip of quality and performance of networks, it is in these situations that sensor cloud computing comes in to picture and provides a solution to these problems.

2. Project scope and description

Following diagram shows a gist of the application.

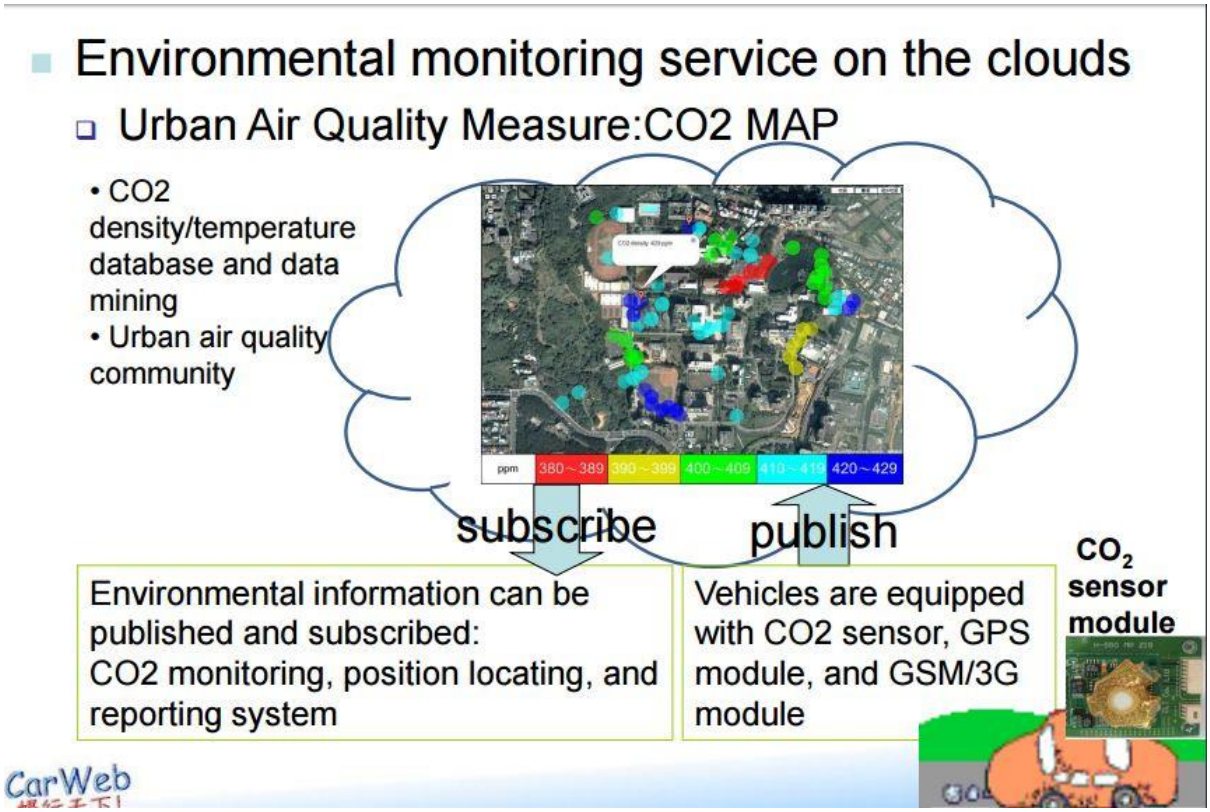


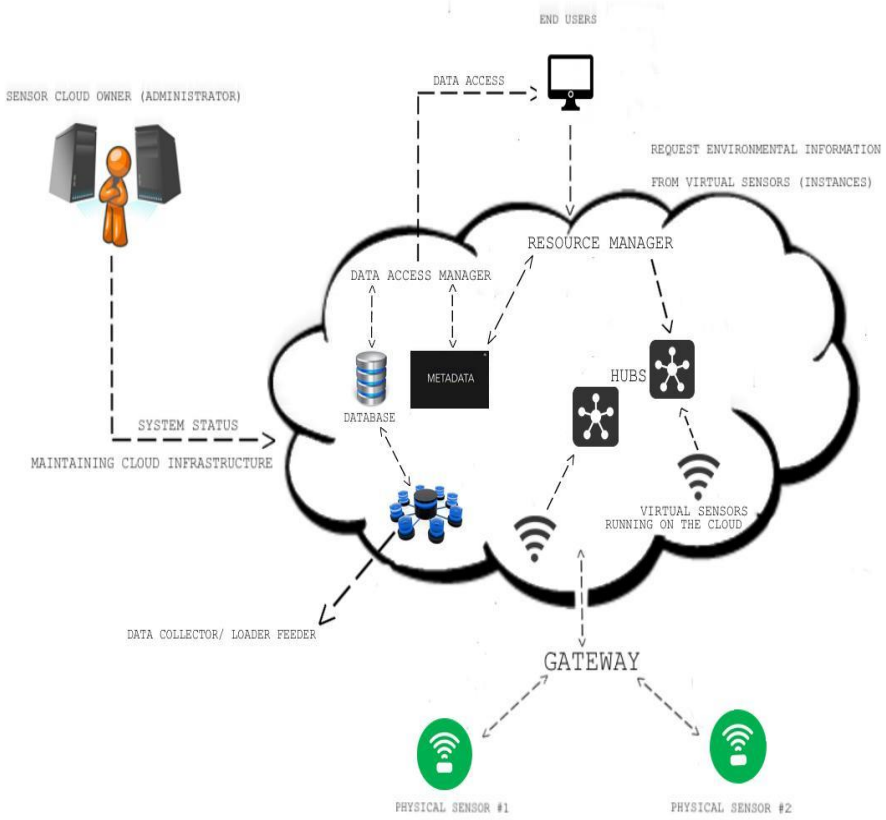
Figure 1 : Introduction to the application

Mobile sensor cloud is used in many applications such as disaster prediction systems, environmental awareness systems, seismic analysis systems, healthcare industries, educational institutions, etc. For examples let's talk about a sensor cloud infrastructure that is deployed in a health care industry like a system that monitors patients with diabetes or high blood pressure. This system is supposed to monitor a person's sleep activity, sugar levels and pressure levels of the patient. In the normal approach these details of every patient are reported to the doctors in an interface of telemedicine. Information related to the patient is stored in a server so that it can be used by doctors or nursing staff at a later point of time to diagnose any other diseases of that patient. Consider this system if a patient travels to a different location from time to time. In this case, we need a more mobile approach to handle all the data and to keep all the data available. Data that is collected from different several sensor nodes of a wireless sensor nodes could be processed in pipelines and parallel way, and hence to making the system easier to scale and be economic in terms of resources present. Pipelining of instructions can be possible or data need not be pipelined if processing of data in that application does not require so. In any case, mobile sensor cloud can be integrated to wireless networks as a solution.

In this document, we first talk about the system infrastructure and design, where we first show how the infrastructure of this mobile sensor cloud appears, then we talk about each and every component that is present in the system, next we talk about the same system in perspective of deployment. We next discuss the UML diagrams of the project. We list a complete set of services that are provided by this system and develop a plan to achieve complete implementation of the system.

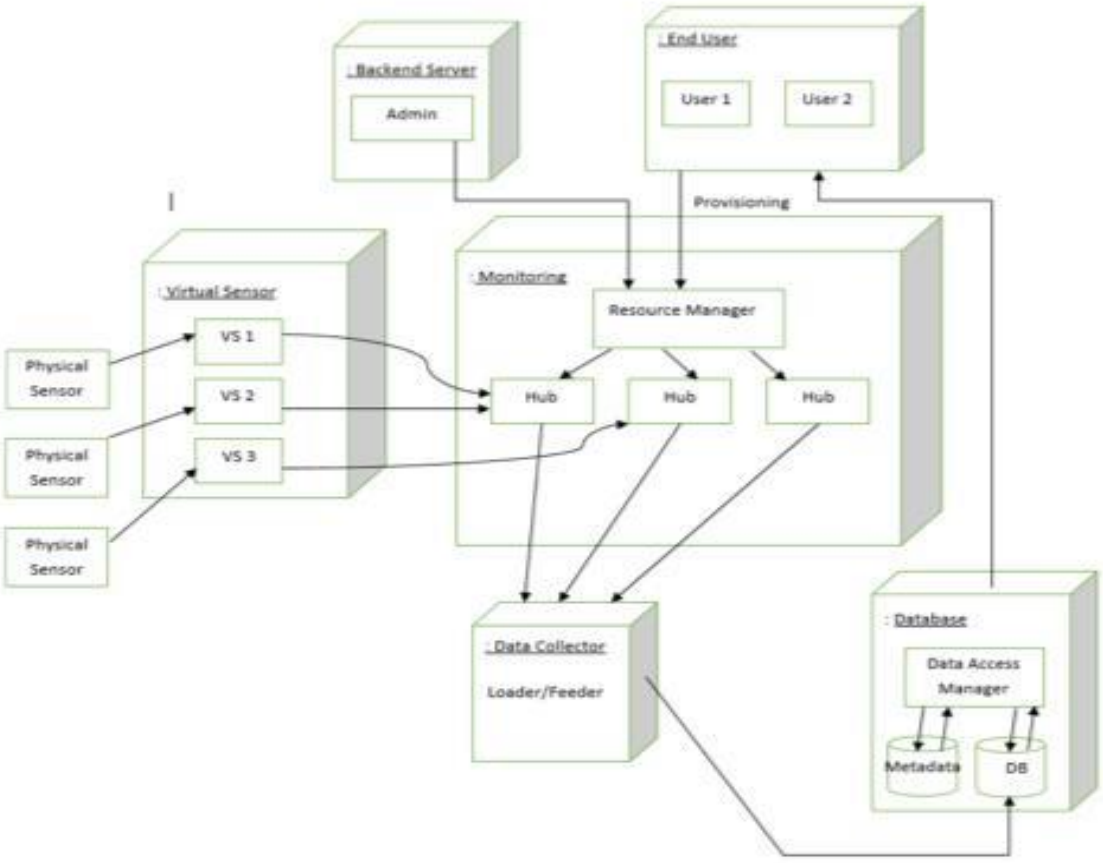
3. System Design

3.1 Infrastructure Design



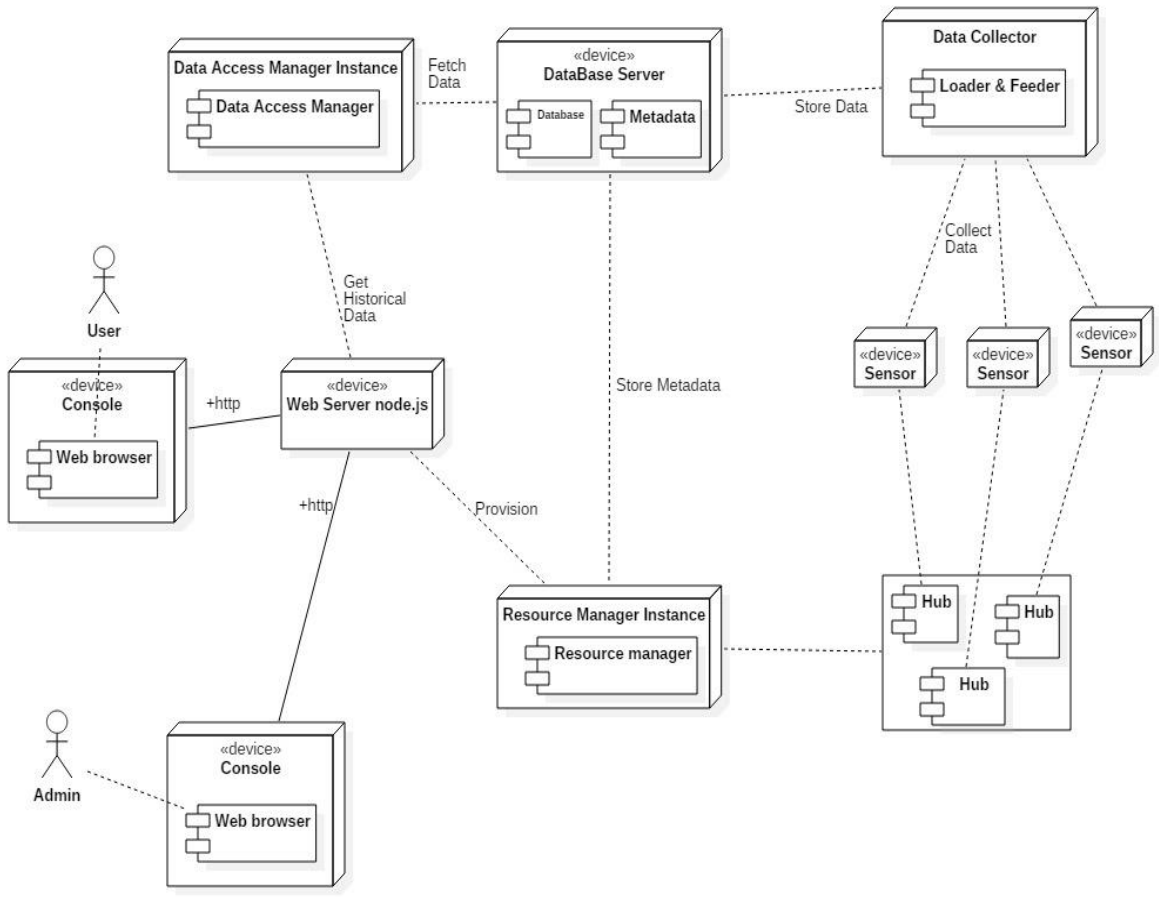
- Data obtained from Air Composition sensors using network (internet)
- The Hubs manage these sensors and fetch the data from the sensors.
- Data collector running on an EC2 instance will periodically collect data from the hubs and Load it into the database.
- MySQL and MongoDB used for the databases. The database servers will be running on a separate EC2 instance.
- Metadata DB will store details about the sensors allocated to the user while the MySQL RDBMS will store data fetched from the sensor.
- The Data Access Manager running on a separate ec2 instance will fetch the data requested from the user.
- Another EC2 instance runs the resource manage that will provision Hubs and sensors for the user.

3.2 Component design



• Sensor Resource Manager allocates and manages sensors according to the user's request.
• Sensor Data Collector periodically collects data from the sensors and loads it into the Database.
• Sensor Data Repository consists of a Database (DB) that stores the data collected by the Data Collector.
• Sensor Data Access Manager supports the access of existing sensor data from the database.
• Mobile hub can track and report sensor status, report, and collect the live sensor data based on predefined schedule.

3.3 Deployment diagram



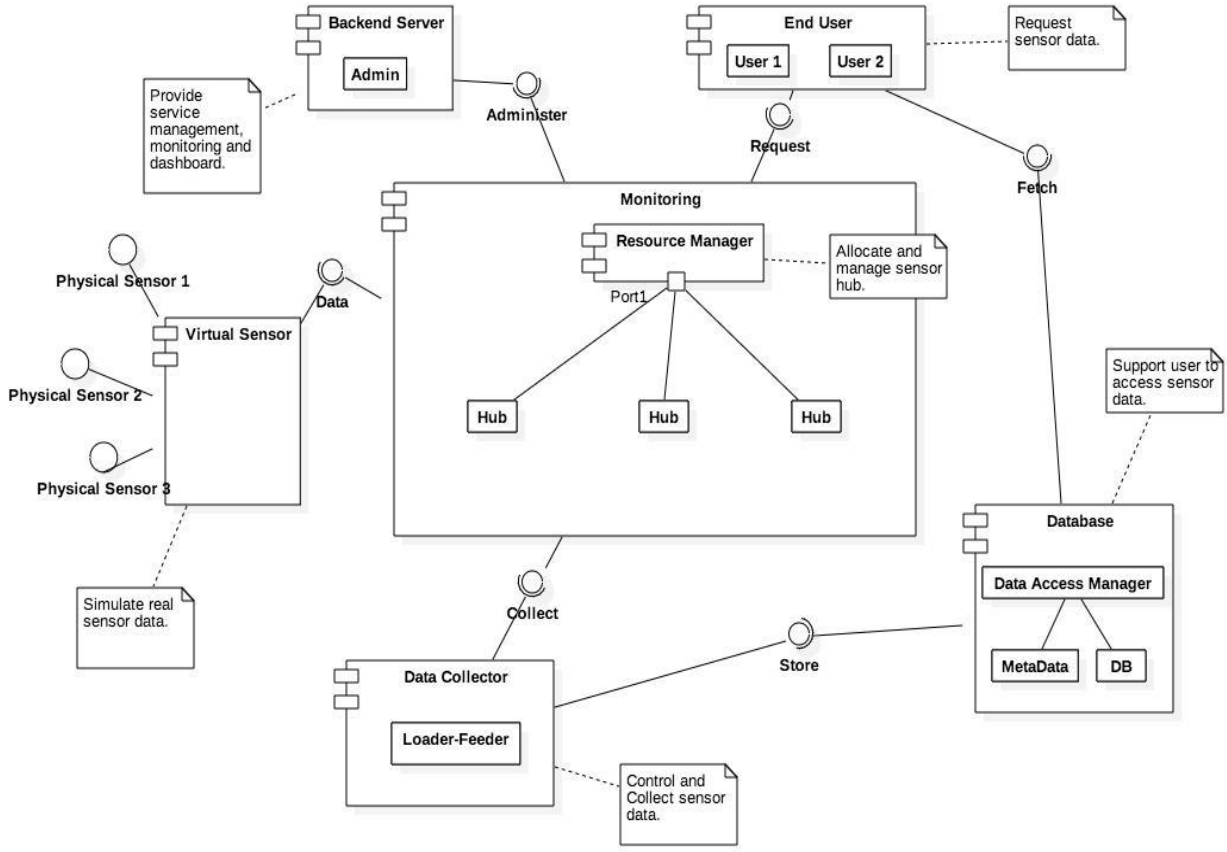
- Sensor devices are individual programs that fetch data from the sensors periodically.
- Hubs are components programs allocated on a per user basis which manage collection of data from the sensors provisioned for the users. These hubs can run on separate ec2 instances to support

scalability as the number of users increases.

- Data Loaders can also run on multiple instances in a load balancing configuration to redirect requests to free loaders in case of huge data loads (for large number of users and sensors).
- Users can view their provisioned sensors from a console dashboard and also request for new sensors.
- To manage large number of users, a load balanced configuration can also be used where multiple instances are automatically created for the node.js webserver to distribute the load.
- Admin can use the Admin console to monitor the cloud system status, check system health, configure and manage sensors, etc.

3.4 UML Design

This shows your system in terms of components. (Client side and Sever side)
Section #3 - Sensor cloud system services



4.System Implementation

Back-end mobile cloud server

1. Service management, dashboard features are provided.
2. Through an interface of dashboard, users are able add /delete/ modify sensors, this is running on node.js
3. Server communicated with resource manager to process user requests. User can request provision or modify a sensor.
4. Server also functions with data access manager to fetch data that is requested by the user

Mobile sensor hub

1. Reporting and tracking of sensor status is done by sensor hub.
2. Collection of live data and reporting this dta is performed by sensor hub.
3. This collection of data is performed based on a scheduling algorithm and it is also passed on the data collector for data to be stored in database

Sensor resource manager

1. Each use is allocated one hub by resource manager.
2. Each hub is mapped to different virtual sensors based on user requests.

Sensor data access manager

1. Existing data related to sensors can be accessed by using this module.

Sensor data collector

1. Main job of this module is to collect data from hubs and put it in to the database.
2. This also performs load balancing, that is to distribute load across many aws instances.
3. New instances of loader is created to provide scalability to the application.

Sensor data repository

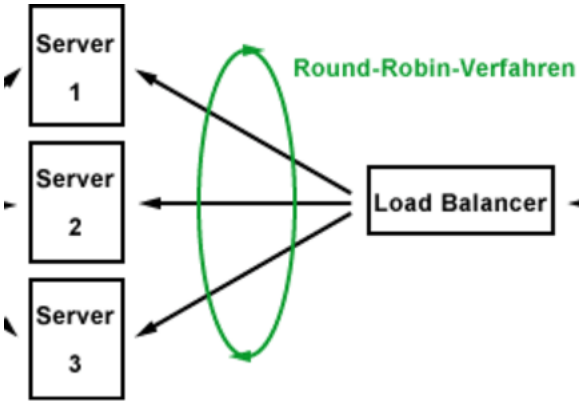
1. This contains a collection of sensor data, that can be used to perform useful data mining.

Sensor Metadata repository

1. This module is just a database repository of data of sensors, user information, hubs etc.

Algorithms Used

Round Robin algorithm for load balancing:
Round robin algorithm is used for load balancing.



In round robin algorithm, requests are distributed to servers in turns, once it reaches the end of servers, requests are looped back to the first server. This is the simplest and most effective algorithm.

5.Development plan

Our team plans to perform the project development as per the below schedule (with respect to the deadlines for the deliverables).

Deliverable #1: (3/15/2016) (Team deliverable) – [Design Document]

- Research sensor networks and cloud technologies.
- Plan the system architecture
- Finalize the language(s), frameworks and cloud technologies to use for developing the system.
- Distribute components to be developed individually by team members.

Deliverable #2: (4/04/2016) (Individual deliverable) – [Project Component Program & Demo]

- Develop the Hub, data collector and sensor modules.
- Develop the database repository and data access manager.
- Develop the resource manager to manage and allocate sensors. Store sensor metadata in the repository.

Deliverable #3: (5/02/2016) (Team deliverable) – [Project demo and program with PPTs]

- Create API's for the individual components
- Create Admin dashboard & consume the API for monitoring, configuring and testing the components of the system

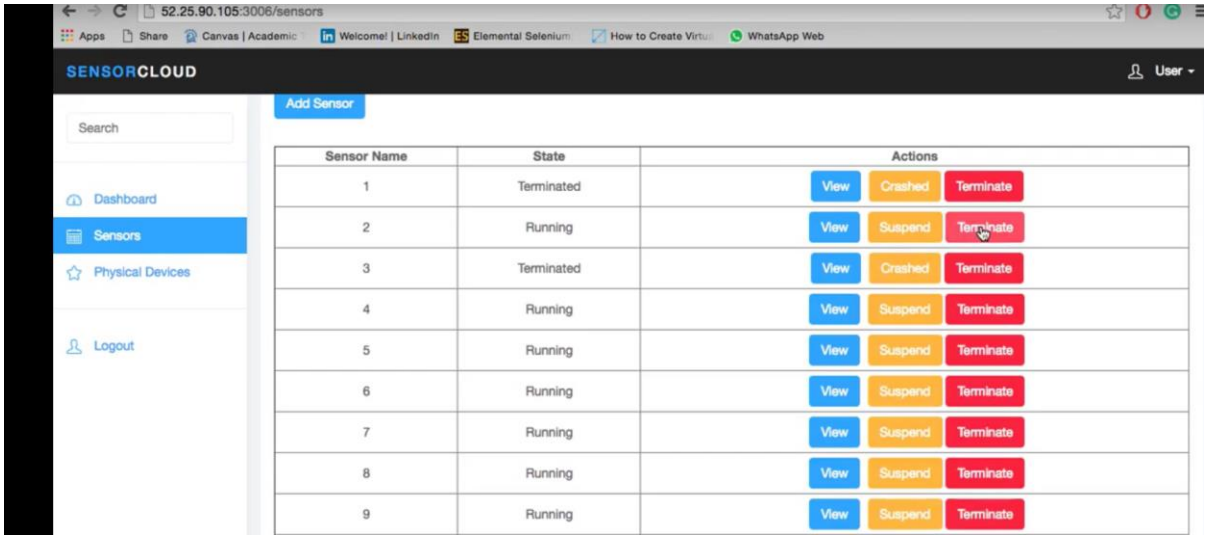
- Create the load balancing configuration for scalability. Develop API's that can be used from the admin console to test the scalability of the system. Show statistics such as number of users, number of instances, etc. - Create user dashboard that consumes the API's for provisioning Hubs and sensors and for accessing data from the sensors and historical data from the repository. Use charts API, etc. to display the statistics to the user.

6.Technologies Used

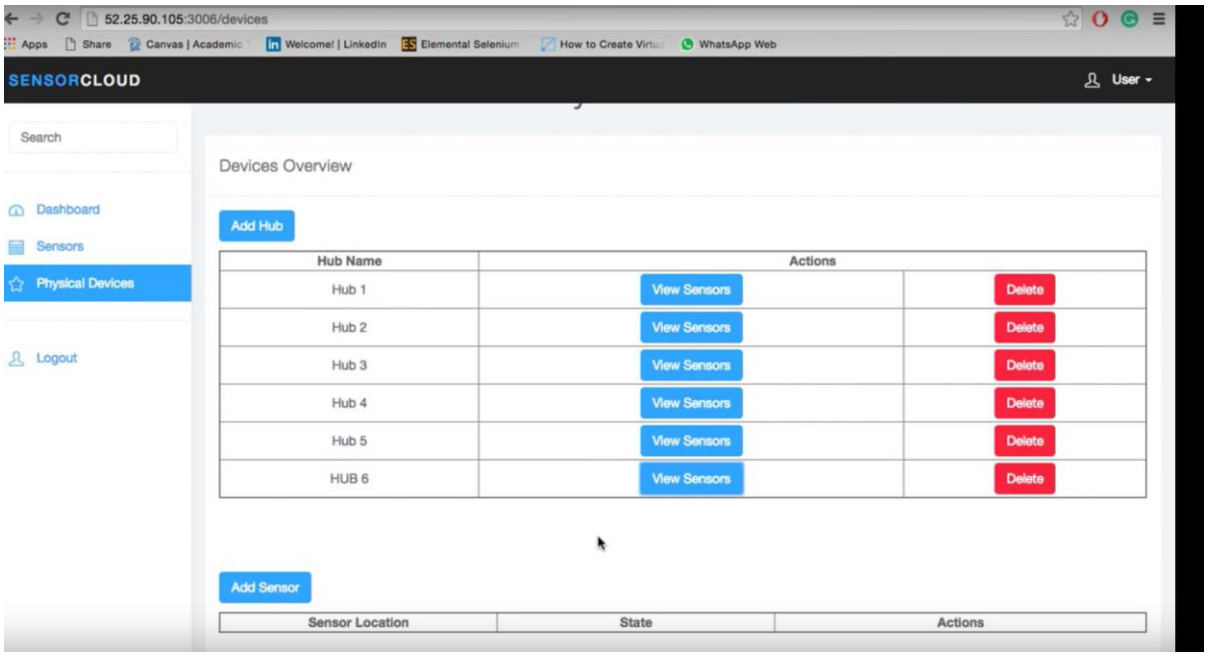
FUNCTIONAL COMPONENTS	**TECHNOLOGY/TOOLS**
Web Server	Node.js
Front End	HTML5, CSS, JavaScript, Bootstrap, AngularJS
Individual Components (Hub, Collector, Data Access Manager, Resource Manager)	Node.js
Database Server	MySQL, Mongo DB (Metadata)
Cloud Deployment	Amazon Web Services (AWS) Elastic Compute Cloud 2(EC2)
Requests and data transfer	RESTful services
Tests & validation environment*	Node.js

7. Screenshots

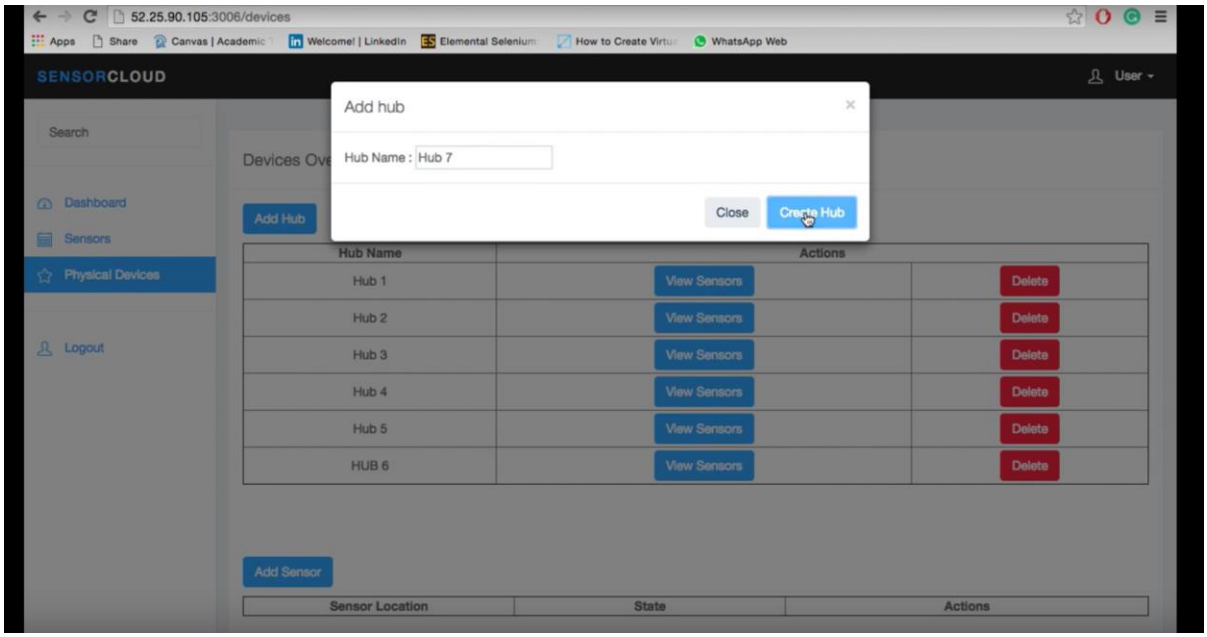
Add a Sensor, view a Sensor, Terminate a sensor



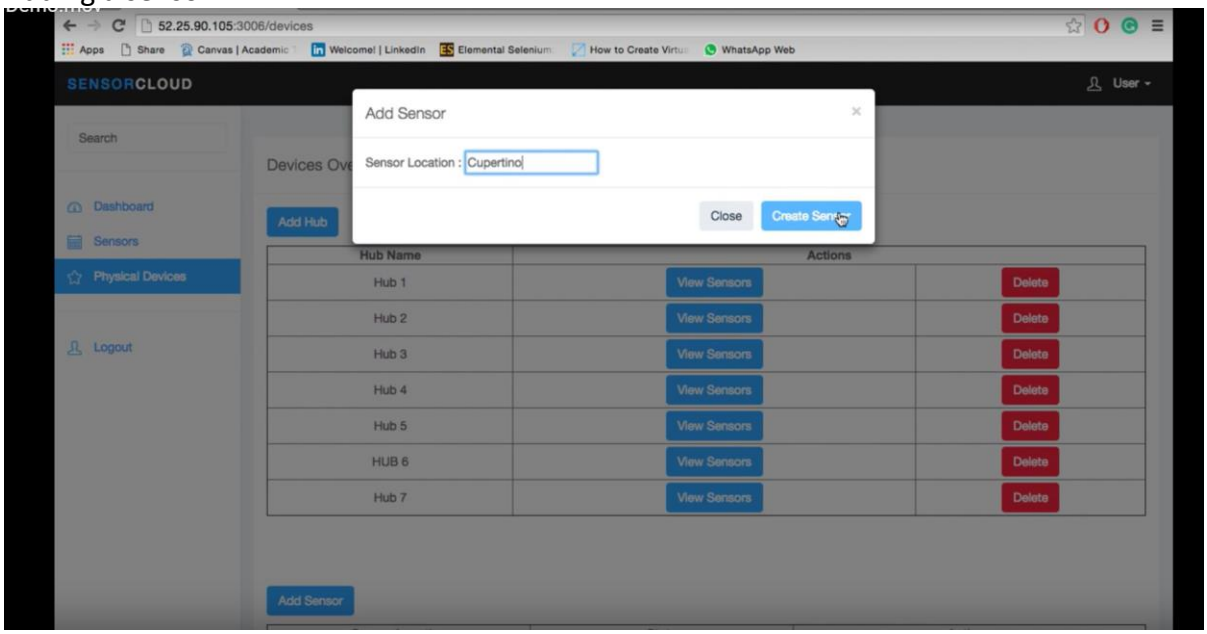
Viewing sensor and hubs



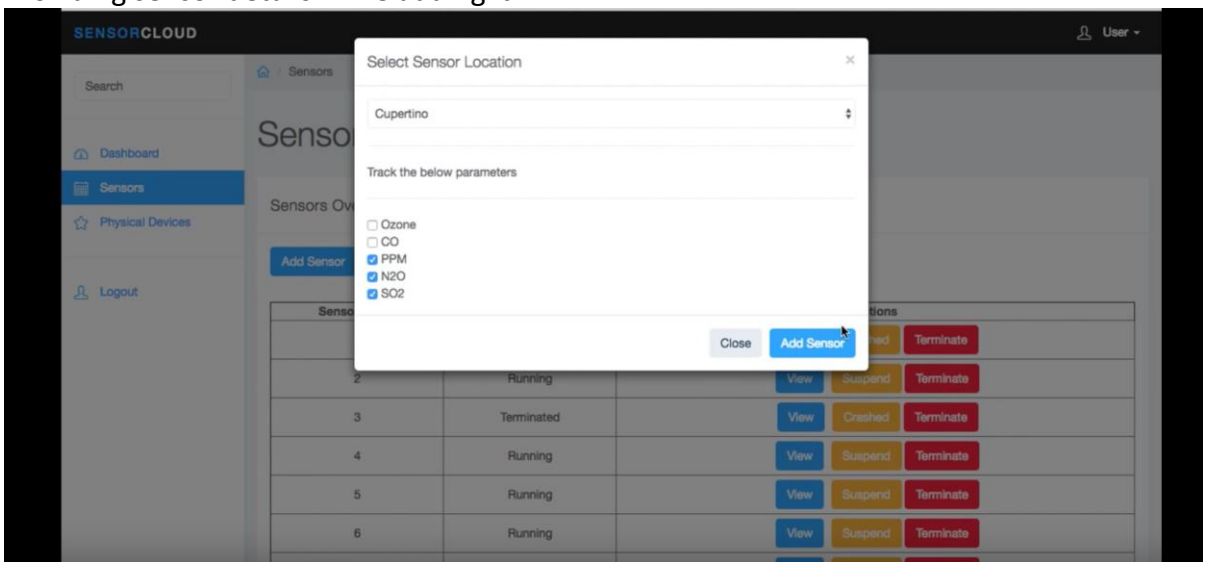
Adding a hub



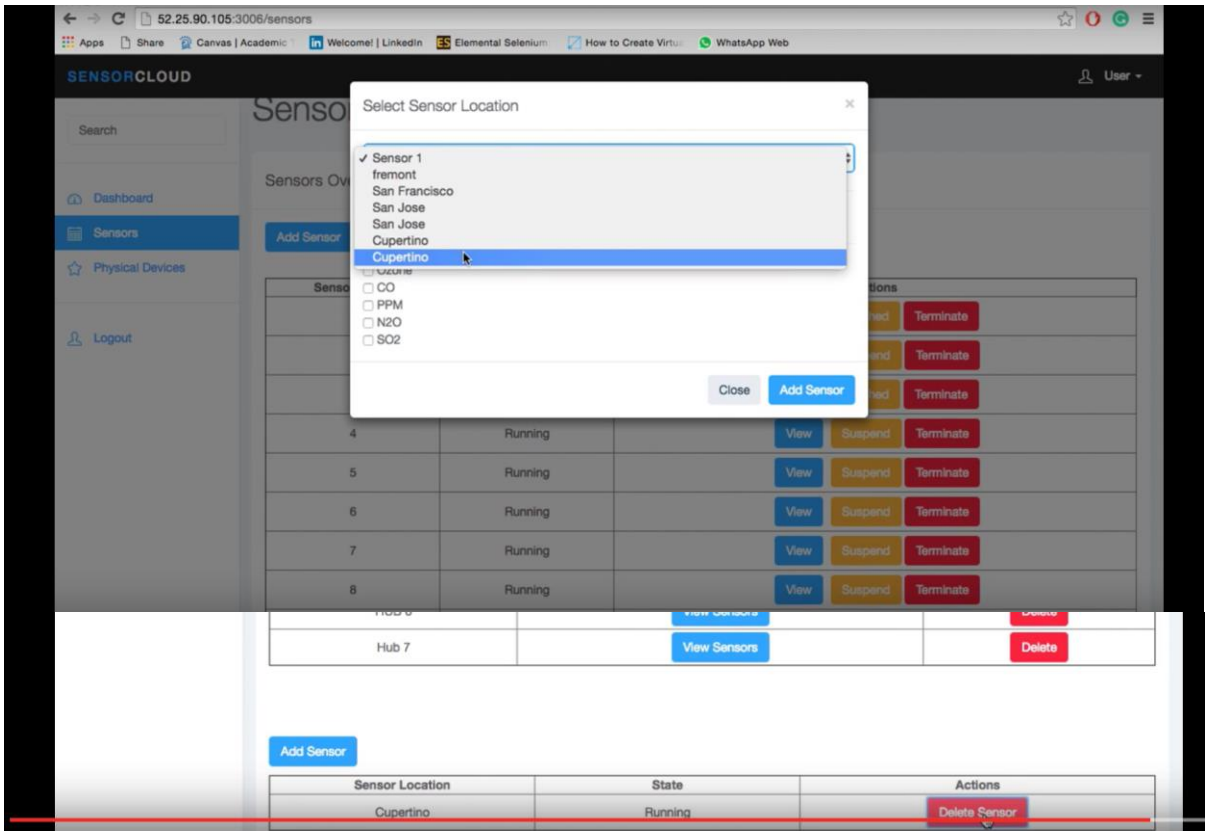
Adding a sensor



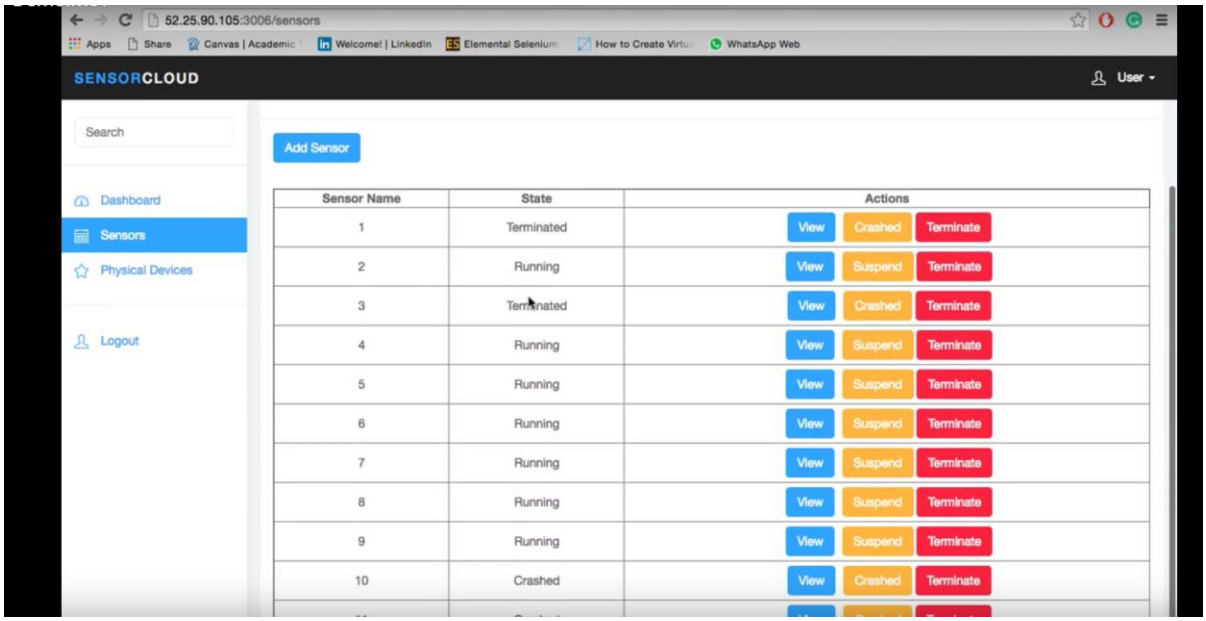
Providing sensor details while adding it



Adding sensor location



View different Sensors



Deleting a sensor

SYSTEM DEMO

Link : https://vimeo.com/167075786

8.REFERENCES

- Mobile Sensor Cloud Computing: Controlling and Securing Data Processing over Smart Environment through Mobile Sensor Cloud Computing (MSCC) published in Published in: Computer Sciences and Applications (CSA), 2013 International Conference on
- A Survey on Sensor-Cloud: Architecture, Applications, and Approaches Atif Alamri, Wasai Shadab Ansari, Mohammad Mehedi Hassan, M. Shamim Hossain, Abdulhameed Alelaiwi, and M. Anwar Hossain