IIIT, HYDERABAD

Internals Of Application Servers

Team No : 1

Team Members:

Abhishek Saban (201405601)

Ankur Shrivastava (201405551)

Chinmay Patel (201405627)

Dharak Kharod (201405583)

Mrugani Kurtadikar (201405621)

Pawankumar P (201405637)

Ruchir Sharma (201405558)

Shwetha G (201405525)

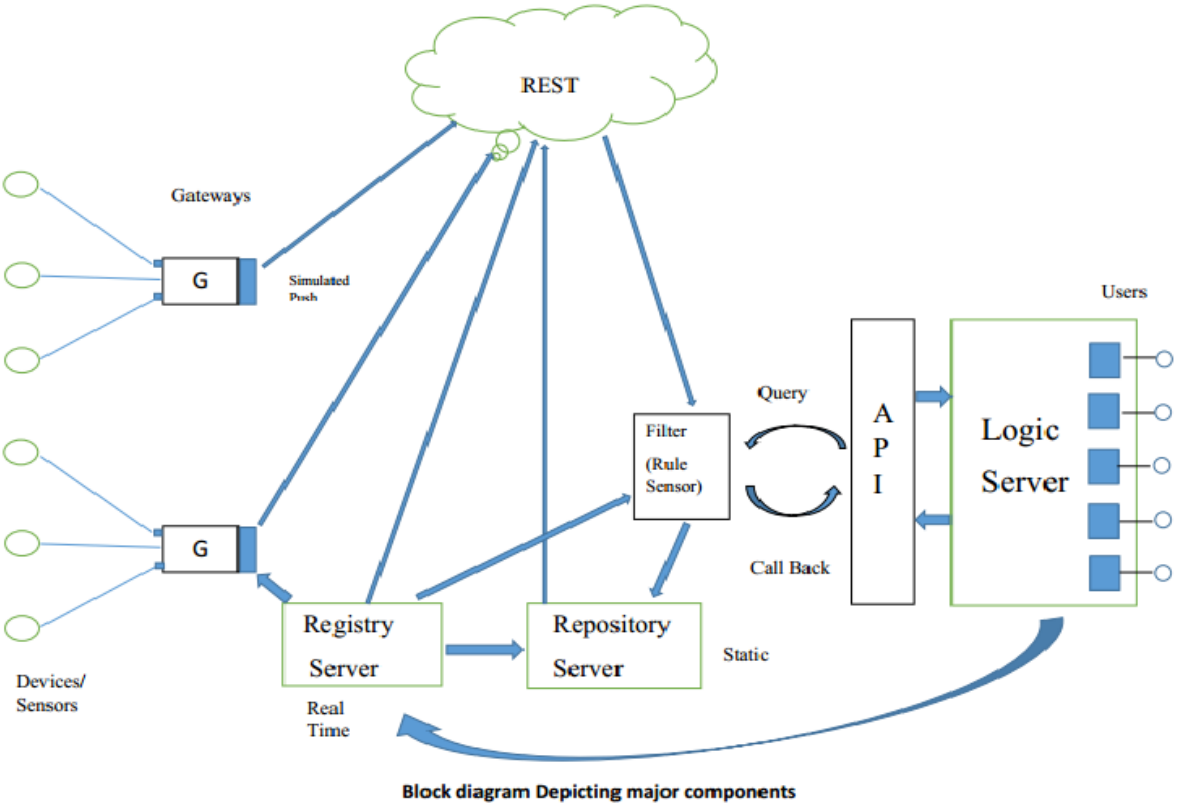
Suhas K S (201405524)

Project: Design Outline

1. A. Introduction:

This project mainly deals with the development of platform which builds a generic platform between heterogeneous sensory devices and user application software in IOT genre. Here since it is a simulation project sensory data will be simulated and sent to the clustered gateway from where it is pulled by the server. From user's end the required data will be extracted from the server using the defined rules engine. The project will be developed using simulated gateway from android devices, mongoDB, node.js. This document is the brief description of the above components with real world use cases of the same project.

B. Functional Overview:



Block diagram Depicting major components

List the key functions

- Sensors broadcasts data over BLE to connected gateway
- Gateway formats the received data using predefined protocol and stores it temporarily

- Filter server pulls data from all gateways configured to it at regular interval of time which forms *simulated push* of data.
- Filter server stores the sensor data in its database
- Rules engine exposes set of API's for the *Logic server* to interact with our platform.
- APIs include getting current data/data within a range/data from a particular location/registering for callbacks etc.
- Data received by logic server is further processed according to user's requirement.

2. Test cases

a. Application use cases

Following are some of the use case that describes how the platform can be integrated with the application.

1. Detection of movement in a room with motion sensors by registering for callbacks in filter server
2. Getting current traffic status at a particular location based on GPS co-ordinates
3. Weather monitoring by collecting range of data of past 3 months using range queries.

b. Test Scenarios:

1. Testing the validity of sensory data:
In the user application built on IOT platform get the data of the sensor which actual value is known to user. Check the data obtained from this application is fine or not.

2. Testing the status of a sensor:
The registry server pings gateway which in turn pings sensors to get the status of sensors attached to it. Tested by switching off one or more sensors and checking their status in registry server

3. Testing the registration of gateways and sensors:
   - Check whether newly added gateways and sensors in admin UI is able to connect or not.
   - Check the entry in repository server and status in registry server

4. Testing the connection between logical server and filter server.
   - Verify initial registration using username
   - Verify the token sent as response to user application registration
   - Verify invalid token sent by logic server are not processed and error 401 is sent back in response
   - Verify the presence of token in every request sent from logic server

5. Testing the database.
   - Verify the database collections are created properly or not.
   - Verify the sensor and gateway registry collections data.
   - Verification of communication between filter server and database.
   - Verification of the storing and retrieving of data sent by sensor is working fine or not.
   - Verification of the request service of database.

6. Testing the real time request of data by user application.
   - Testing the validity of data in real time request.
   - Testing the response time for real time request.

7. Load testing.
   - Test the application using many sensors
   - Test application using many applications
   - Test application by sending enormous request at the same time.

8. Smoke testing.
   - Test the end to end flow from sensor registration to data retrieval from user application.

9. Cross platform testing
   - Keep the IOT platform in different OS and test.
   - Request the data from different type of user application
   - Test the platform for the support for heterogeneous sensors.

3. Solution design considerations

3.1 Description of components

Sensors

A sensor is a transducer whose purpose is to sense (that is, to detect) some characteristic of its environment. It detects events or changes in quantities and provides a corresponding output, generally as an electrical or optical signal. There can be multiple types of sensors.

Gateway

Gateway acts as an interface between sensors and the server. It collects the data from various types of sensors with various kinds of interfaces and forwards it.

Communication Channel

It acts as a medium for transmitting the data from gateway to the filter server. Various gateway's communicates to the server through this channel. Protocols like REST which is used for communication between the modules are implemented over this channel.

Filter Server
It receives raw data from various gateways through the communication channel. It then eliminates the noise and filter out useful data into system recognizable structured format.

Repository Server
It is a static list of all the resources available in the system. It would load details of each and every component of the system from config file on startup and make it available for every authentic component of the system.

Registry Server
It is a dynamic copy of repository server which continuously checks the status of every components of the system and maintains a log for any query or transaction.

API
It provides an interface between end user and server. It implements various methods and functions to expose the data through the query engine.

End User Application (Logical server)
This application acts as the consumer of data and performs suitable action depending on the result of data generated through the API.

Major Modules

Part ONE:
It manages the interaction between **sensors and gateways**. A gateway collects data from various sensors and passes it to the next level. Also includes the repository server which is a static record of the metadata of the system's components and registry server which is a dynamic log of system which includes status of various components

Part TWO:
Filter server collects the data sent by various gateways through the communication channel(REST). It parses the data and stores it in system recognizable structured format into the database.

Part THREE:

End user applications requests the data stored into the database through API and performs suitable actions on the basis of results obtained on processing of that data.

3.2 Technologies to be used:

a. REST: It is a lightweight, stateless, platform independent protocol.

b. Node.js: It is faster, event driven.

c. Android: used to develop user app, ease to develop various types of app.

d. Mongo DB: stores in no sql format which is good to store huge data.

4. User's view of system

What is an application- for each type.

- The logic server acts as the consumer of data and performs suitable action depending on the result of data extracted from the database by the filter server communicated through the API. The developer at the end will use the exposed API to develop applications according to given scenarios (which can be static or real time).

- **What are the files (config, etc)**
- **Config fil**e: It contains connection related information such as IP address of the server, port numbers with which it interacts and other protocol related information. It also contains the general information about the devices.
- **Session file:** It contains the session id, start time and end time of the particular session, dependent on when the user access the application.
- **Log file**: Details of history of the application (operations performed), it contains the information about the previous sessions.

- **Structure of the files**

Repository Addition File:

```
<Gateway>
        <Gateway id>
        <Gateway location>
        <Gateway name>
 </Gateway>
<Gateway id>
        <sensor>
                <sensor id>
                <type>
                <location>
        </sensor>
```

</Gateway>

- Registry **File:**
  <Timestamp>
  <Device ID>
  <Type of query>
  <Status>

- **Session File:**
  <Device/Server ID>
  <Start Time>
  <Stop Time>

- **How is it deployed/setup. And where (on which module/process/location)**

The setup consist of an server which needs to be started by admin. Suppose a new device needs to be added then a ui is provided to add. The new user will sign up and use the same for further request.

- **How to start & stop**

The application would consist of an option which would initialize the process and connect the application with the server whenever the user uses that option. The process will stop as soon as the user will exit the application.

- **How to access/invoke**

The application would be accessed through an android device, initially the user would be required to register his application. The user would be given an API key, later the application can be invoked through that API key and user Id.

<u>User interactions</u>

- **How will the user use this system**

User application need to create a logic server using node.js to interact with the platform.
User can use appropriate apis from the list of apis exposed from filter server

- **What & how to configure**

During installation the user would be required to allow access to GPS and other network services to configure the application. After registration the user will get his particular login token which would be used to configure the application whenever user opens it.

5. Interactions & Interfaces

- User interactions

On user's demand, the application will request the data through the API. After receiving the data from the server the application will perform the processing on that data and based on that processing, the user will get the desired results as per the query.

- Module to Module interactions

- Module ONE:

It manages the interaction between **sensors and gateways**. A gateway collects data from various sensors and passes it to the next level.

- Module TWO:

Filter server collects the data sent by various clusters of gateways through the communication channel. It parses the data and stores it in system recognizable structured format into the database.

- Module THREE:

End user applications requests the data stored into the database through API and performs suitable actions on the basis of results obtained on processing of that data.

- File/Wire format definitions

1. Between Sensors and Gateway: Since the sensors will be the android devices, the interaction between them will be using Bluetooth. The sensor will transmit UDP data packet containing the information such as:

Sensor Id, Timestamp, Register domain Id, type of sensor, Actual data

2. Between Gateway and Filter Server: The communication between them will be basically TCP/IP and REST calls will be made to communicate. The gateway will process the data received from sensors and removes noise (filters the data) and forwards that packet to the server. The data transmitted would consist of the following fields:
Gateway Id,Sensor Id, Timestamp, Register domain Id, type of sensor, Actual data

3. Between Server and Android Application: Socket programming will be used to interact between these. The application would request required data through an API

which would act as an interface between the app and the server. The API would expose JSON object which would consist of the following fields:

Transaction ID, Timestamp, Formatted Data, Session information,

6. Technical Details:

- The sensors will continuously collect and broadcast data of the required type(temperature, motion, images etc).
- The sensors will prepare data in the specific pre – decided format.
- Gateways will ask for data from sensors using REST in specified intervals.
- At the time of initialization, gateways will collect information about their connected sensors by querying repository server. After that gateways will communicate with their sensors directly.
- Whenever new resource is introduced into the system, admin needs to make an entry in xml file present in repository server.
- Registry server will continuously check the status of every components in the system at specific intervals.
- Filter server will get data from all gateways using REST and store data in predetermined format in the database. MongoDB will be used as a database .
- The filter server will then prepare data in the database to answer the API queries issued by the user.
- The filter server can expose the raw data after processing it through a versatile API.