IOT based sensor system

Introduction

IOT based sensor monitoring system will serve many purposes. Sensors can be of different types, whether it be temperature sensors, humidity sensors, pressure sensors etc. Since all the different types of sensors have different type of data they transmit, it is very difficult to make use of the data in its raw format. Hence, we provide a platform that allows application developers to build and deploy the apps that make use of data that we receive from the sensors and provide essential information to the users. For e.g. temperature sensors can be used for central ac system. The temperature from different sensors can be read and analysed to find out whether the temperature in the corresponding area should be increased or decreased.

The platform abstracts the middleware of the system and the APIs are exposed to the developers. They don’t need to worry about the data format (as data from different sensors will have different meanings), the communication protocol etc., as all these are taken care by the platform. Rather, they just need to focus on what app they want to make and how will the app deliver what it promises to the target audience using the api’s provided. An efficient level of abstraction is implemented so the user only sees what’s important to him and all the other secondary things are hidden. Since, the platform caters to heterogeneous class of sensors, the data from the sensors are read and classified into different categories based on the requirement.

Use Cases

1. **Central Air Conditioning System** – Temperature sensors can be used to identify temperature in low/high temperature areas and the data can be analysed to increase/decrease temperature accordingly. Since, the location of sensors can be identified on the central server, it is easy to identify, which sensor is transmitting the corresponding temperature information. Hence, this data can be used to control temperature automatically at the specified location.
2. **Humidity Sensors** – Chemicals and other household stuff are prone to moisture damage. Humidity sensors can be used to track the amount of moisture present in the area. The sensors continuously reads the data, transmits it to the central server and the server can notify the application user whenever the humidity percentage crosses a certain threshold so as to alarm the user.
3. **Car Parking** – Car Parking is a serious issue in today’s crowded world. In malls, where the space for car parking is huge, it is very difficult find out an empty slot. We might need to roam a lot before we find space. To avoid this hassle, what we can do is, install infrared/light sensors at each parking slot. If the slot is occupied, sensor will identify the object and notify to the central server which can then push this information to the application that the user has installed. The app directly shows the available/occupied slots and hence the user can select the nearest spot for parking.
4. **Fitness Tracking** – GPS sensors are very common in mobile devices. These sensors can be used to record the path travelled by the runner. Later on this data can be used to identify the distance travelled which in turn can be used to calculate the amount of calories burnt during the run based on the time taken by the user to complete the distance. Other information like average speed, pace and heart rate can be calculated and displayed to the user.
5. **Fire Detection** – Smoke sensors can be used to detect the presence of smoke inside the building. The sensors can transmit data to the central server and the owner of the house can be alarmed about the incident at the earliest.

Key Functionalities

1. Collection of data from various heterogeneous sensor and gather the details of each sensors dynamically.

2. Security check is applied to maintain the confidentiality of information.

3. Real-time data processing and providing the valuable information in lesser time.

4. System is dynamic in nature so minor changes in sensor reading are tracked and taken care of.

5. It provides a layer of abstraction to a developer who is building an app using our platform.

Descriptions of components

1. **Sensors:** A device which detects or measures a physical property and records, indicates, or otherwise responds to it. In our project, we have heterogeneous types of sensors, for example, temperature sensors for sensing temperature, humidity sensors for measuring humidity of a place at a particular time, light sensors for measuring ambient lighting conditions, etc.
2. **Gateway:** Gateway is a device which is used to connect two different networks, especially a connection to the internet. In our project, these connect sensors to the filter server (or sensor server) through a communication medium. Gateway, after getting the data encapsulates it into a message format that contains type, id, location, etc., so that the sensor server can identify what kind of data it is receiving.
3. **Communication Medium:** A medium that helps to communicate Gateway and sensor (filter) server. It uses RESTful APIs to transfer information from gateway to sensor server.
4. **Sensor Server (Filter Server):** As the name indicates, it filters the information it gathers from the gateway via the communication medium and make it available to the developer who will be using it in his application. It hides all the irrelevant information which is of no use to a particular developer who intends to build application on our platform.
5. **Application Platform:** It allows developers to build secure, data-driven application using our API, which can be accessed from mobile devices anywhere, anytime. This application platform provides the ambient information gathered by different real-time sensors like temperature sensor, weight sensor, light sensor, humidity sensor, smoke sensor etc., through the APIs. The API only exposes the required functionality hiding all the other things that may be irrelevant to the user.

Primary Use Case: Central AC System

The idea is to provide real time temperature notifications to the user. The sensor data can be displayed to the user through a mobile application which will pop notifications whenever the room temperature crosses a certain threshold, giving the user an option to control the temperature accordingly.

Temperature sensors can be used to measure the temperature in any desired area. The sensors will analyse the change in temperature and will constantly send the data to the central server. At the central server each sensor is bounded to a unique id which is used to identify which particular area or location the sensor belongs to. After analysing the data, central server forwards the information to the application installed in user’s mobile phone .The application further provides user with an option to increase/decrease the temperature.

User can monitor the temperature from anywhere across the world. Consider a case where a user is out for some work and has his children alone at home, the user can control the temperature of the house/room from his work place making use of the application and internet connectivity.

Components:

**Temperature Sensors:** Temperature sensors gather temperature data and send it to the central server. The sensors must be installed in such a way such that no sensor is present in the range of some other sensor to avoid data inconsistency.

**Central Server:** Central server receives all the data from the sensor. It maintains a table at its end which contains id-location mapping, type etc. After gathering the data, it then sends it to filter server via communication channel.

**Internet: I**t is the most important communication component in any such model. All the information, request or response generated by various components are transmitted via internet. It makes possible for the user to control the things from anywhere around the globe.

**Information Model:**

(will insert diagram)

**Sensor and Location Information:** Sensor basically sends the reading that it captures from the location it is installed in. For e.g. temperature sensor reads temperature data of a specified location. Location Information is required because there can be a lot of sensors placed in a small area, so we need to get the precise location before we act on the reading that we get from the sensor. For e.g. there can be many temperature sensors placed inside a building but you must know the exact place where you want to increase/decrease temperature.

**Backend Processing:** The sensors are connected to the gateway which will forward the data to the filter server via TCP/IP protocols. The filter server will check for the validity of the data and any duplicate or invalid data is neglected. The Sensor registry table will also be accessed and information will be mapped to the corresponding sensor id. Finally the filtered information is forwarded to the application using Internet.

**User View:**

* **Notification bar:** Whenever any information is fetched form the central server it can be notified to the user using notification bar.
* **Content Area**: This area is used to display the information sent by the central server.
* **Response field:** User can give the instructions increase/decrease in temperature through the response field in the screen
* **Button:** After giving the response user can click in “**Submit**” button which will save the user’s response and will forward the request to the central server.

Definition and Scope

With the ever-increasing amount of data that is inherent in an IoT world, the key to gaining real business value is effective communication among all elements of the architecture. **The** platform delivers an integrated, secure, comprehensive platform for the entire IoT architecture across all vertical markets. The platform allows application developers to build and deploy the apps that make use of data received from the sensors and provide essential information to the users. The platform abstracts the middleware of the system and the APIs are exposed to the developers.

Following factors need to be analysed for efficiently developing a platform: -

* Real-time response capabilities for millions of device endpoints
* Faster time to market
* End-to-end security
* Integration with IT systems
* A worldwide, coordinated ecosystem of partners
* End-to-end compatibility and lifecycle solution management

Subsystems of the Project

* **Input Subsystem –** It consists of a collection of sensors which could be heterogeneous.
* **Control Subsystem-** The gateway, communication channel and the filter server together constitute a control subsystem.
* **Output/Application Subsystem-** It consists of the application programming interface.

Interactions involved across subsystems

**Sensor-Gateway Interaction -** Socket programming is used for interaction between sensor and gateway.

**Gateway-Filter Server Interaction** - RESTful services are used for interaction between Gateways and filter server. The message header containing information like id, type, location etc. is transferred to filter server for further processing.

**Filter Server-Application-** RESTful services are used for interaction between Gateways and filter server. The filter server identifies valid data, removes unwanted information and then sends the information to the application requesting it.

Device Interface with system

Device interface is provided through Application Programming Interface (API). API provides the functions to the developer and to the administrator to view the details of the sensors registered in the system and receive the data sensed by these sensors. Developers can make use of this data in order to develop useful applications. Administrators can add or delete sensors currently in the system.

Registry and Repository

Sensor registry stores the data sensed by the sensors like temperature, humidity, heat, location etc.

Repository contains the details of each sensor and the type of data sensed by it. For example heat sensor, temperature sensor etc.

Logic server (aPaaS)

**Application platform as a service (aPaaS)** is a cloud service that offers development and deployment environments for application services.

Location Services

Location Services will provide developer the location of the sensors from which developer intends to receive the data. Accordingly developer can make use of this information so as to develop domain specific applications.

Mobile Interface

Mobile interface will be provided through an android application which receives data from the application programming interface (API) via a filter server.