IoT based Application Platform

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.

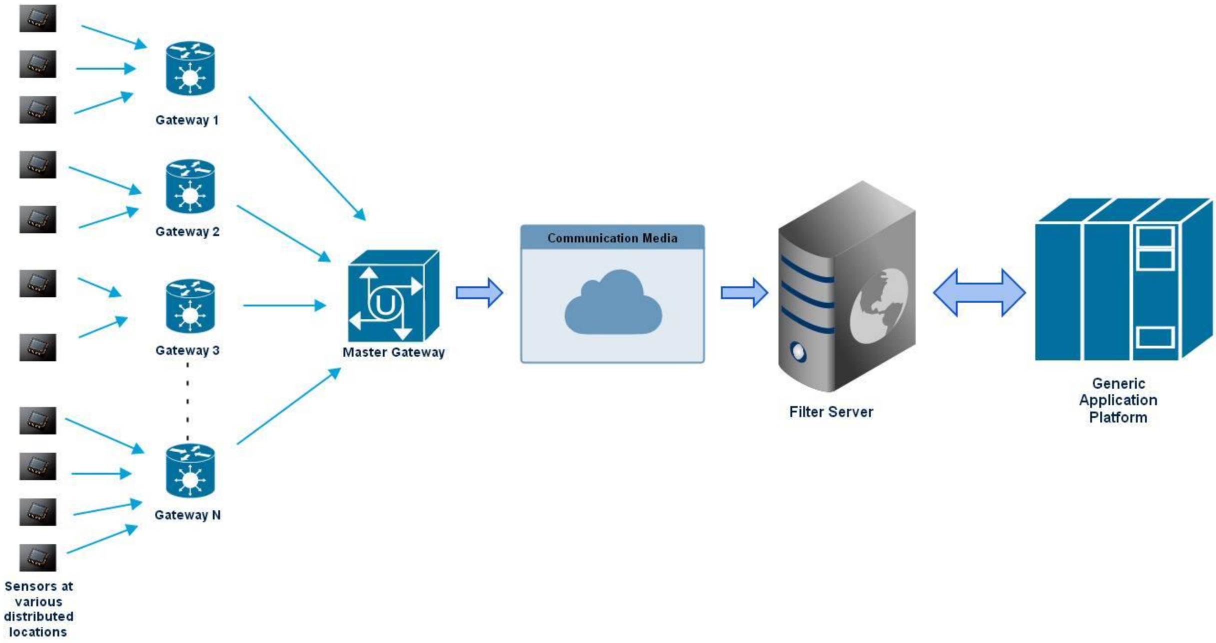
The platform abstracts the middleware of the system and the APIs are exposed to the developers. They don't need to worry about the data format (as data from different sensors will have different meanings), the communication protocol etc., as all these are taken care by the platform. Rather, they just 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. **Indoor Air Quality** – Air Quality sensors that can measure the contents of the air can be used in chemical industries to measure the amount of poisonous content in the air. The sensors can give current real times status of the contents of the air and then update the employees in case of Gas Leakage or hazardous locations so that incidents can be avoided.
2. **Solar Energy Measurement System** – Sensors can be placed at different locations, across the state or country, and now based on the data received from the sensors it could be predicted/analysed, which region has received the optimum solar energy, so that the solar panels be installed in the concerned areas only for maximum efficiency.
3. **Advanced Wireless Automatic Digital Pumping System for Agriculture using Soil Sensor** – Soil Sensors can be used to detect areas where the soil quality content has been optimum for agriculture. Different attributes like water content, minerals, potassium content etc. could be sensed by the SOIL SENSORS and based on the data received, the water pumps which are connected with the system can be activated automatically thus helping in managing huge farms, helping in optimum water usage.
4. **Advanced Wireless Earth Quake Alarm System for Early Warning** – This might include installing sensors deep beneath the earth's surface. The sensors can detect the behaviour of the surface change and also the pressure above and below the sensor location, and based on such data from various sensors the earthquake probability can be estimated and an early warning could be generated.
5. **Street Light that Glows on Detecting Vehicle Movement** – The usage of motion sensors in combination with thermal sensors would make this system functional. If a vehicle's presence is detected then the motion sensor would help in detecting the vehicle movement and street lights could be automated accordingly. This helps saving energy during late nights when there are no vehicles, the street lights could be temporarily switched off.
6. **Design & Implementation of Railway Automation System with Sensors Network -** This system includes heterogeneous sensors like there would be a sensor which detects when train enters a platform, and the sensor installed for signal generation for the trains. Now based on the data the signals could be generated especially during foggy conditions, a thermal sensor would be a boon if installed nearby the tracks, thus train operation would also be possible in foggy conditions.

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.
6. Usage of RESTful web services for communication over TCP/IP protocols.
7. Storage of all the sensor types. The database used in MongoDB.
8. Classification of sensed data on the basis of their type and then sending it to the respective application that requests it.
9. Creation of protocol header so as to send it across the communication media to the filter server with ease.
10. Maintenance of Sensor Type Repository so as to identify the type of data the sensor sends.



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. **Master Gateway:** All the individual gateways send their sensed data to the Master Gateway which then forwards it to the filter server using the communication medium.
4. **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.
5. **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.

6. **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: It 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.

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.

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

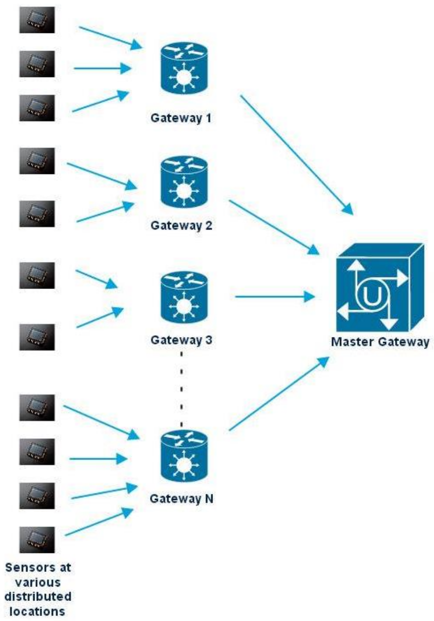
Mobile interface will be provided through an android application which receives data from the application programming interface (API) via a filter server.

Interactions

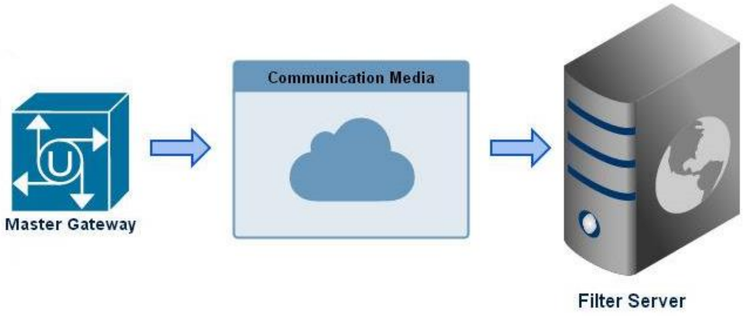
First of all, the data sensed by sensors are sent to the respective gateways which collect the information and then cumulatively send all the data to the master gateway. The master gateway identifies the type of sensor and data by consulting the Sensor Type Repository. Since, raw data cannot be sent across the communication medium, master gateway encloses the data, type, location and other properties concerning the sensor into a protocol header which is then transferred across the communication medium using RESTful web services on TCP/IP protocol. The filter server then filters out only those sensor data that are being requested by the apps that are built using the APIs provided by the platform. The Sensor type repository uses MongoDB as the database and the Gateways are instances running on a RaspberryPi system to simulate the functionality of an actual gateway.

Interactions involved across subsystems

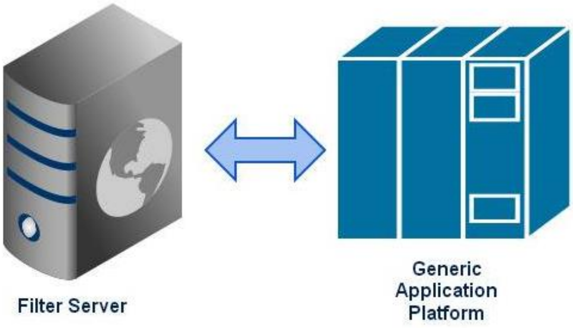
Sensor-Gateway Interaction - Socket programming is used for interaction between sensor and gateway. There is a collection of gateways which are simulated using a RaspberryPi system. The sensors send their sensed data to the gateways they are connected and all these gateways then send the collected data to the Master Gateway which does further processing and sends it across.



Gateway-Filter Server Interaction - RESTful services are used for interaction between Gateways and filter server. The Master Gateway creates a protocol header containing the required information to transfer across the communication medium. The communication medium is the Internet and the data is sent over the TCP/IP protocol. 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. Since filter server receives data of all the types of sensors that are installed in the area, it must filter out only those that are requested by the application, i.e. an application requesting temperature information must not be shown humidity information and vice versa. For this, the filter server consults the sensor type repository and then sends only valid data to the application. Also, the filter server validates the source of generation of data. This is primarily required from the security standpoint otherwise malicious data might be shown to the user.


Filter Server
Generic
Application
Platform