**Module1: Sensor to Gateways**

***Internal design overview***

This would be the main input serving module for the platform to succeed. All the Sensor based operations, domain specifying constraints (for example as in temperature sensors in air conditioning systems) need to be handled here only.

The sensors functioning, addition, deletion would purely be dynamic in nature. Thus, the module focuses on that only.

***Lifecycle of the module***

* First of all every sensor needs to be registered with the gateway
* Then the gateway will assign a unique identification number (id) to each sensor, now that identification number (ID) would be valid every time the platform serves the particular application.
* The user needs to only specify the type of sensor which he wishes to incorporate, to the gateway and then the sensors could be installed at the desired places and the data generated by the sensors could now be received and processed further.
* User has to take care that the transfer technology used would be Bluetooth, for transferring the data generated by sensors to the gateway.

***SUB-MODULES***

***Registry Module***

* This module handles the registration of new sensors to the gateway.
* All the registration based methods are invoked in this module.
* It will store the sensor details along with its ID in the database.

For example- If a new temperature sensor is in range and needs to register then this module would handle such case.

***Control Module***

* This module is basically responsible for analyzing and processing the data received from various sensors which are registered with the gateway.
* It is responsible for forwarding the received data to the filter server.

***Kernel Module***

* This module loads in the communication device kernel modules required to use various sensors.
* It is invoked whenever the gateway needs to interact with the sensors.

***Interactions***

***Wireless Communication between Sensor and the Gateway***

This would be the communication medium between the sensors and the gateway.

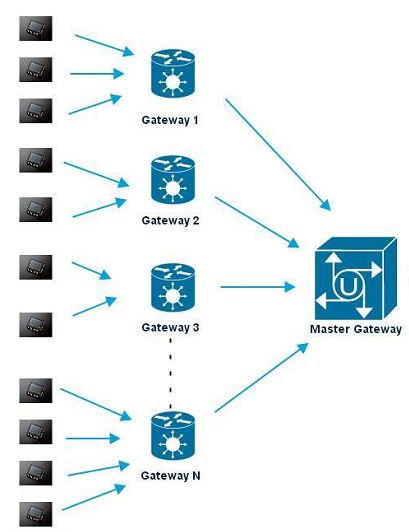
All the sensor metadata would be stored in the database. MongoDB will be used for this purpose.

The sensors would generate data and send it to the gateway for further processing.

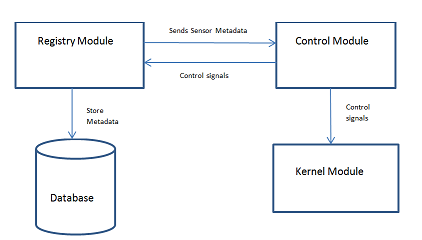
***Gateway Interaction with Filter server***

The gateway would assign the IDs to the sensors and store them in a database dump. This will then be forwarded to the filter server.

***Block Diagram***

**

***Inter-Module Interaction Block Diagram***



***APIs and Classes***

**Registry APIs**: API that is used to register new sensor in the system. All the registration based methods are exposed to the developer.

**Control Class**: Class that is responsible for analyzing and processing the data received from various sensors which are registered with the gateway.

**Kernel Class:** Class that is used to load the communication device kernel module required to use various sensors.

**Interaction between sub modules**

**Registry module & Control Module:** Registry module sends sensor metadata to the control module. Control Module sends control signals to the registry module. The metadata includes the information about the sensor such as sensor id, type, data received etc.

**Control Module & Kernel Module:** Control Module sends control signals to the kernel module. The kernel module loads the communication device kernel module required to use various sensors.

**Registry API & database**: Registry API sends the sensor data to the database for persistence. The database would store the sensor ID & various data related to the sensor in the system.

**Interaction between other modules**

**Control Module** will forward the data received from the sensor to the filter server. Wireless Communication will be used between the gateway and the filter server. Control Module will also forward the data to the database for storing the data received from the sensor for persistence.