**Module3: Filter Server to Application**

**Life Cycle of the Module**

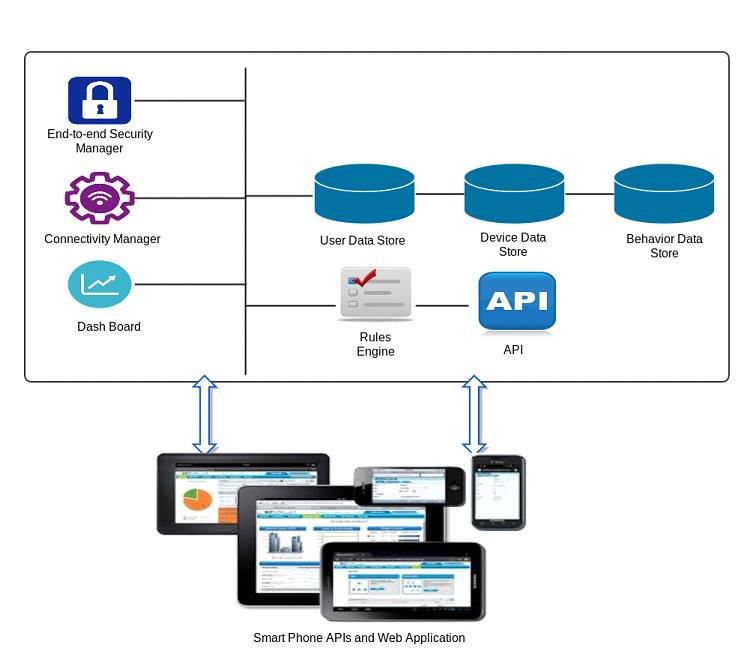
**Initiation, Monitoring and Stopping of module**

* **Initiation of Sub-Component**: An application platform is made available to the developer by the system. This setup is done by the system admin itself at the time when system is up and starts all the modules and makes them serviceable. So the filter server will always active and wait for any data request.
* **Monitoring:** Monitoring of all the sub-module is made automated and managed by system automatically. Separate modules are written for managing the devices and ensuring error free data transmission. In case of any physical fault system admin is responsible for recovery of physical devices.
* **Stopped:** Although the system is made available all the time throughout but in case if system has to be stopped it is done by system-admin. Before the system would stop it will serve all the active connection and make safe connection termination. So that data transfer will not results in error and user’s service will not interrupted.

**List the sub modules**

1. **Connectivity Manager:** This component will provide the connectivity between various components.
2. **Security Manager:** This component will ensure that each application will receive authenticated data.
3. **Rules Engine:** This sub-module will contain information about the active applications.
4. **Filter Module:** This sub-module will filter the data that is coming from gateway.

**Block Diagram**

****

**Brief overview of each sub-module/ component**

1. **Connectivity Manager:** This component will provide the connectivity between
   1. Filter Server and Rules Manager
   2. Filter Server and Central Database
   3. Filter Server and Application Server
2. **Security Manager:** This component will ensure that each application will receive authenticated data. For Example, application that works on Humidity readings should not receive temperature readings.
3. **Rules Engine:** This sub-module will contain information about the active applications. Information should contain sensor id, type, status, location, time stamp, checksum etc.
4. **Filter Module:** This will filter the data that is coming from gateway. The type of data that application requests, will be given by filter server to application as a response. The Filter Module will only provide that type of data that application request.

**Functionality of sub-module**

Functionality of system, when it is in phase three, i.e., from filter server to application platform, is as follows:

1. Filter module/server will be running continuously. Application will make a request to the filter server.
2. Filter server will acknowledge the application, by providing it some credentials.
3. Application then requests for the data it wants from the filter server.
4. Filter server then provides the data to the application. If the application that requests the data is active, then it will send data directly from gateway, otherwise, it fetches the data from the database.
5. Filter server will also ensure that only that type of data is send, which the application needs.
6. Application platform will provide API to the end users that help them to build applications for their purposes.

**APIs and Classes**

Following APIs and Classes are used:

* **Developer Class:** Class containing information of each developer.
* **Rule Class:** Class containing list of active applications
* **Sensor Class:** Class containing sensor information, such as id, type, location etc.
* **Filter Class:** Class that filters the required sensors information from all the sensors information.
* **UI Class:** Provide methods and tools to create UI of applications.
* **Query API:** API used by logic server to query the registry server about the active devices on the basis of type, gps coordinates and range. The registry server returns the list of all the valid devices that are currently active.
* **Command API:** The command API is basically used to send commands by logic server. The commands may specify a device type and location requiring the data of only that sensor/device present at the specified location of specified type.
* **Callback API:**  The API is used to register callbacks on some particular event. For e.g. remind the user after the temperature reaches 35 degree or above. This kind of request gets registered at the filter server and whenever it is fulfilled an event is triggered and data is sent to the logic server.
* **Security API:** API that ensures the authenticity of data received by the application server.
* **Rules API:** API that provides access to active applications with the current data received from the gateway and archived data from the database to the other applications.

**Interactions between sub modules**

**Connectivity Manager:** Connectivity manager interacts with the different modules of the system and verifies whether the connectivity between all the modules is being maintained or not.

**Security Manager:** Security manager ensures security of the system by verifying that the correct type of data must be sent to the respective requesting applications. This means that the malicious applications must be recognized and no data must be sent to them. This is done by communicating with the rules engine by and getting back the information about the application and the type of data it requires.

**Rules Engine:** It interacts with the security manager for maintaining the security of the system. It sends all the details of the application such as (its running status and the type of data it requires).

**Filter module:** This module gets the data from either the gateway or the central database and filters the data based on the type of data requested by the application and then forwards the respective data to the application.

**Logic Server:** Logic server can use the Query API, the Callback API and the Command API. Logic server first consults the registry server about the active devices and possibly the type and location is specified in the query. The registry server returns the list of valid devices and sends it to the logic server. After this, the logic server may use command API or callback API as needed to fetch the sensor data. The command API may be used if it requires a data of some particular sensor and the callback API may be used to register an event for retrieval of data.

**Interactions between APIs and classes**

**Developer class** will interact with the **Filter API** to retrieve the data of the sensor of particular type(s). **Filter API** will interact with the **Rules API** to check the type of application, whether it is active application or not. If it is active application, then Filter API will contact to the gateway to fetch the results, otherwise it will pick results from the database (archived data). It will then send the data back to the application to the developer. The developer will use the methods and classes to check the authenticity of data. It then provides the API to users to build applications using that information.

**Interactions between other modules**

* The **Application Platform** module will contain all the applications. The applications will request for a particular type of data through the application platform. This request is being received by the Filter server module.
* The **Filter Server** module will receive the request by the applications. Logic server sends request to filter server to get the data by consulting the gateways that receive the data from the sensors. It will interact with the rules engine to maintain the active applications list. It will also interact with the central database for the archived data. Filter server then responses back to the authorised application in the form of filtered data.

**Other Design Considerations:** The current design can further be made efficient by using virtualization at hardware level so that the performance of the system can further be enhanced.