A interaction carried out between user's control devices and the sensors installed to exchange information. The end points of the system are the sensors and user's device and a CRM. The whole process can be broken down into three parts.

Event receiving and Control

1. Data from the sensor enters our Cloud server and into a CMS/SDK where it gets stored in the form of log files.
2. We then use a custom library to listen to the SDK ports and load the data onto kafka Brokers so that it gets distributed and carried onto the next phase.

Spark Processing and Streaming

1. The second phase of the process is about managing the data recieved from kafka Brokers by using a consumer/zookeeper server from our spark engine.
2. The data is recieved onto a Ruling Engine.
3. The unstructurred data gets stored in our NoSQL database, for future analysis.

Access and Control Management

In this user's end phase, we will be using two different platforms for recieving and displaying information. All the requests-response process will be carried out through an API written in Scala using Swagger which will fetch/create/update data from the Ruling engine and display the response onto the respective platform upon getting a valid HTTP request.

* Mobile Application- The Mobile application will be able to accomplish the following set of tasks.

1. Login/Logout of an existing account.
2. Make a valid GET request to get all relevant data of the device sensor from database. This request will be included in a runnablle thread as soon as the devices' state are set to armed.
3. Contact the emergency contacts via SMS/Call, listed at the time of installation in case of a securtiy breach.
4. Make a valid PUT request to change the state of the devices to armed or disarmed in the database.
5. Make a POST request to admin for creating a new subscription or renewing an existing subscription
6. Make a POST request to admin to discontinue an ongoing subscription.
7. Report any issue with the sensor/application.

* CRM- The CRM will be used as a store-front with its own MySQL database having an admin and customer section. The admin would be able to accomplish the following set of tasks.

1. Login/Logout
2. Send messages to customer regarding their subscription status.
3. Manage all customers using valid GET(view)/POST(insert)/PUT(update) requests.
4. Manage the devices available using valid GET(view)/POST(insert)/PUT(update) requests.
5. Manage all the payments made using valid GET(view status) request

The user module of the CRM would be able to accomplish all the tasks listed in the mobile applications with an additional functionality of making payments through a POST method and exception of voice call upon a security breach.