INTERNALS
OF
APPLICATION SERVER

GROUP – 3
TEAM - 3
HACKATHON 2

AYUSH KHASGIWALA (2020201088)
NAMAN JUNEJA (2020201072)
SOMYA LALWANI (2020201092)

Introduction

It is an end-to-end IoT platform applicable for enterprise IoT projects of any scale. It provides a range of features that enable developers to build advanced applications for smart products and flexibly manage their connected devices. With the IoT features provided by this platform, you can create and deploy your IOT applications.

Modules Covered

1. **Application Manager**
2. **Platform Manager**
3. **Monitoring Module**
4. **Bootstrap Module**

BASIC WORKFLOW OF OUR MODULE :

- **Running applicationManager.py file :**
  - In this project there are 2 users, one is **Application Admin** and other is **End User.**

  - Initially the users/admin will be provided a menu to login/register.
  - Then the user/admin can login or register using that menu on the terminal.
  - If the user/admin is not registered then the user/admin will first create a user and then login in the platform and then can proceed with its work.

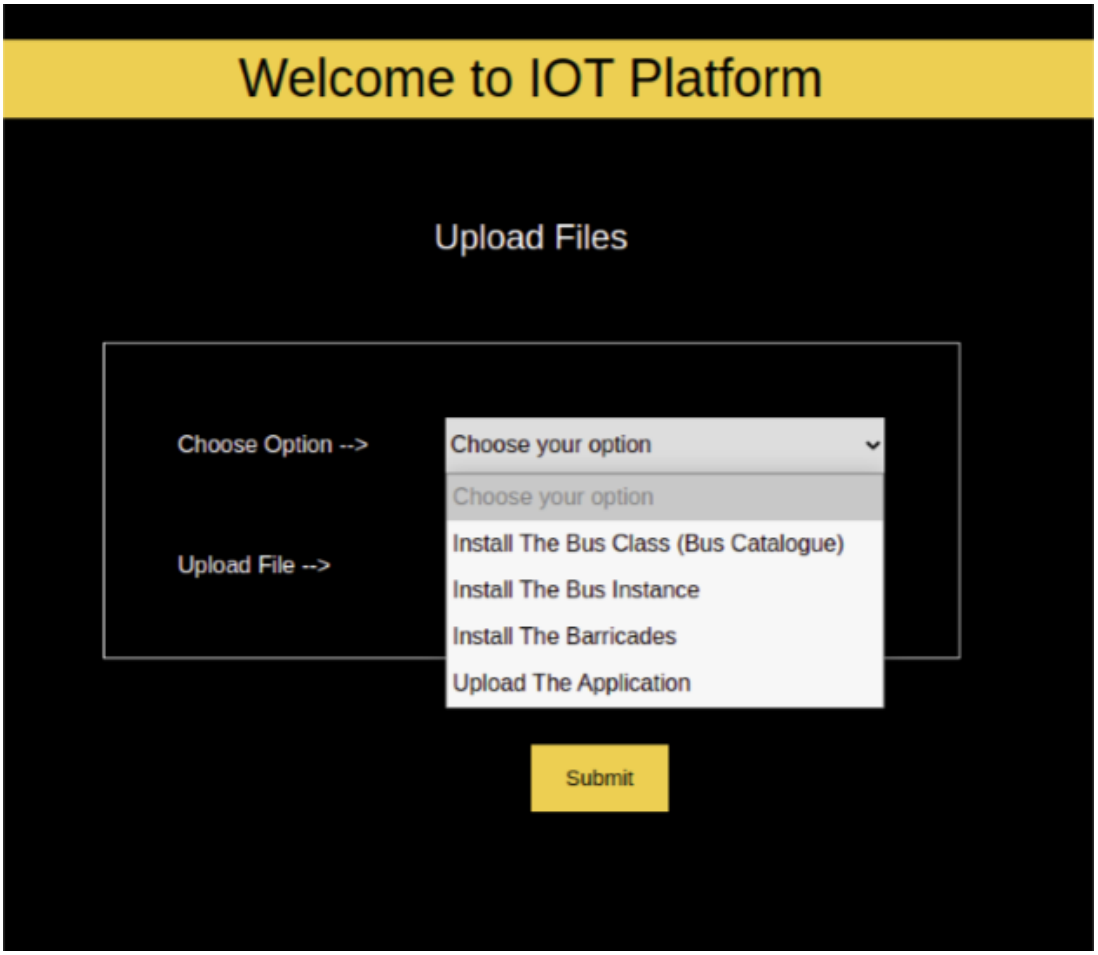
a. Application Admin

- The **Application Admin** is the one who will deploy the application and will ask the platform to install the sensor class(type) if it is not available on the platform. This will be done by providing a config file(sensorClass.json) to the platform
- The Application Admin will also install the instances of the sensor class(type) and this will be done by providing the the config file(sensorInstance.json) to the platform
- And then the application admin will give the platform the application file(code file) and the config file(named -req.json)(to run the application file)

- On sending the file for deploying to the platform manager, then the platform manager will communicate to the sensor manager to validate the config file given to deploy the application. If any of the instances are not available then it will tell the platform manager and then the platform manager will then send the msg to the application manager .
- Then the application manager will show the message to the application admin and by seeing that message the application admin can then install the sensor class(type) and the instances and then again deploy the application. In this way the application will be deployed by the application admin

Designed according to the sample application:

1. Install the Bus Class
2. Install the Bus Instance
3. Install the Barricades
4. Upload the Application



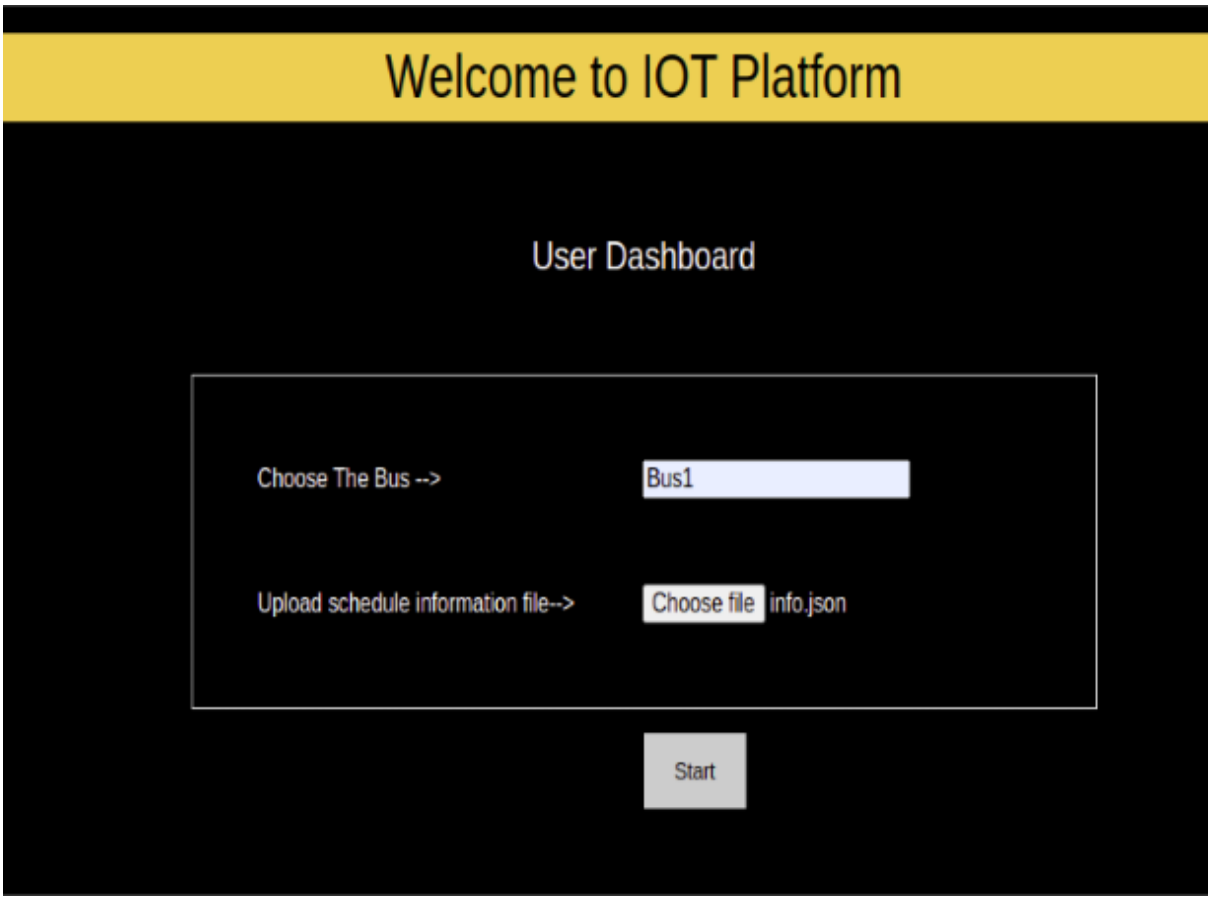
b. End User

- The End User is the one which will run the application deployed on the platform.

- The user will be given a list of the applications (received from Platform Manager module) which have been already deployed on the platform.
- Based on this list, will be asked to enter the following details: application name,user config file, start time & end time. These details are then sent to the Platform Manager module.
- In the Platform_Manager module, data will be read from the user config.json file and based on the location, we use the getSensorIdByLocation to get all the sensor IDs located at that location.
- In continuation with this connection (between Platform Manager and User), the Platform Manager will create new threads and send the following details to the deployer module : list of all the sensor IDs present at that location, start time and the end time.
- On the User terminal, the progress of the running application is shown and the final output of the query asked by the user will be presented in another terminal which is communicated via Scheduler & Deployment Module.

The following options are given to the end user:

1. Choose the Bus
2. Upload the schedule information file (consists of start time, end time, recurring bit, etc. Details – to be sent to the scheduler)



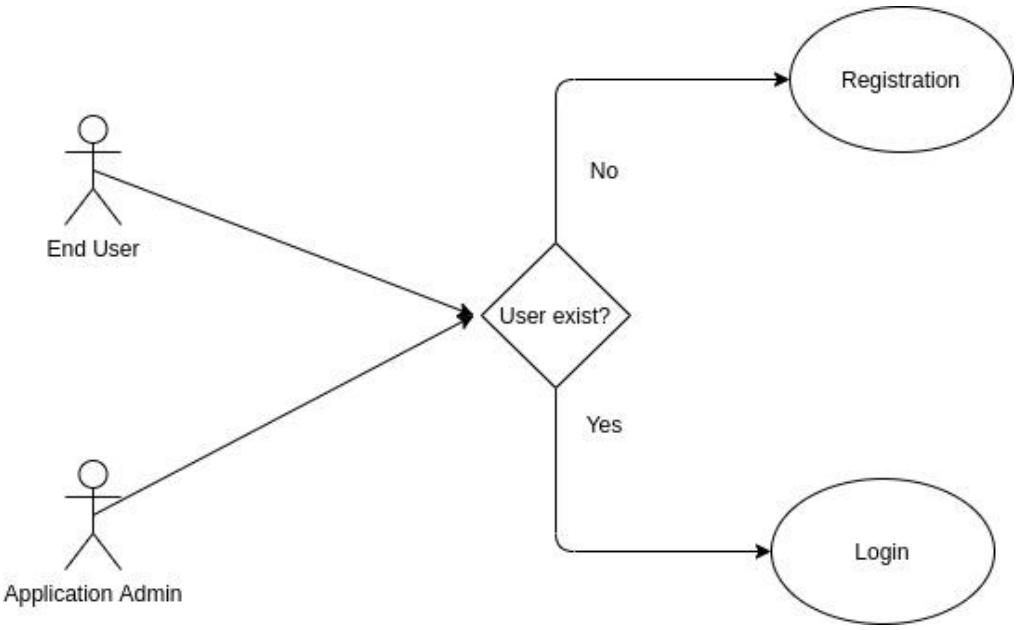


Diagram 1 - Application Module - Registration Submodule

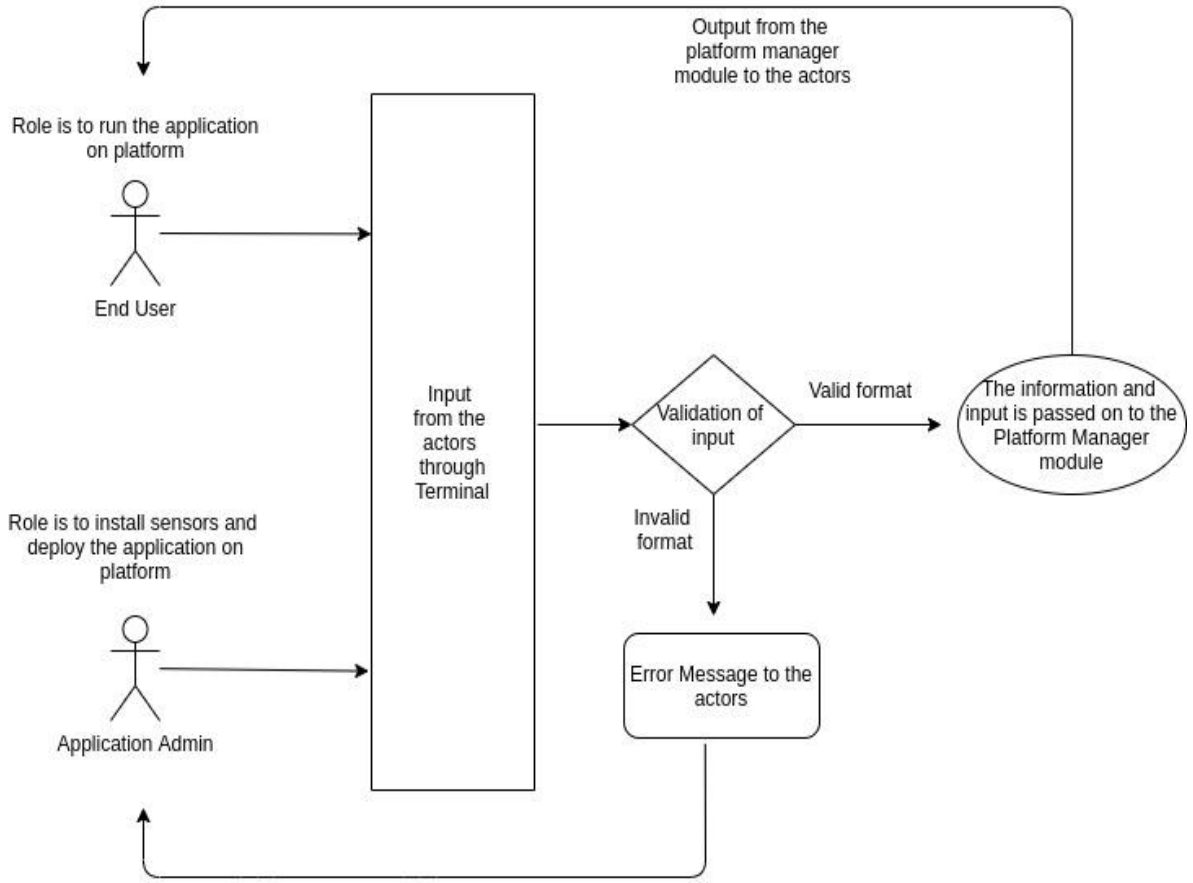


Diagram 2 - Application Manager - Processing Input from different actors

- **Running Platform Manager Module**
  - In this module, we will be communicating with the Application Manager, Sensor Manager and the Scheduler module.
  - As soon as the module is run and the server is established, we create new threads to communicate with different modules mentioned above.
  - The input from the user/admin will be received at the platform manager and then the module will first identify the user i.e. the user will be either Application admin or the End user.
  - For the **Application Admin**, it will first receive the different config files i.e the sensorDetails.json, sensorInstance.json or programSensorConfig.json from the application manager.
  - 3 types of tasks can be done : to install a new sensor class, to install a new sensor instance, to deploy the application.
  - The module will send the data from that config files to the sensor manager module for verification or for the installation of the sensor type and sensor instances.
  - Then the message received from the sensor manager module will be passed on to the application manager module.
  - Once, the application admin thinks that all the sensors that their application needs are installed, then he can send the application to get it deployed on the platform.
  - Then the platform manager will send it for validation to the sensor management module and the validation response is received at the platform manager side
  - If all the conditions are met then the platform manager will deploy it to the platform and it will be then available to the user.
  - For the **End User**, the module returns a list of all the applications already deployed on the platform to the end user.
  - If no module has been deployed yet, it will navigate back to the end user page with the status of “No application deployed yet!”.
  - In the case, where the end user receives a list of deployed application’s name, the end user will send the following information back to the module : appname, start_time, end_time and location.
  - Data will be read from the user config file(.json) file and based on the location, we use the getSensorIdByLocation to get all the sensor IDs located at that location.
  - In continuation with this connection (between Platform Manager and User), the module creates new threads to communicate with the Scheduler Module.
  - In these threads, it sends the following details to the deployer module: list of all the sensor IDs present at that location, location, start time and the end time. The output of this call is the status of the application.
  - On the User terminal, the progress of the running application is returned from Platform

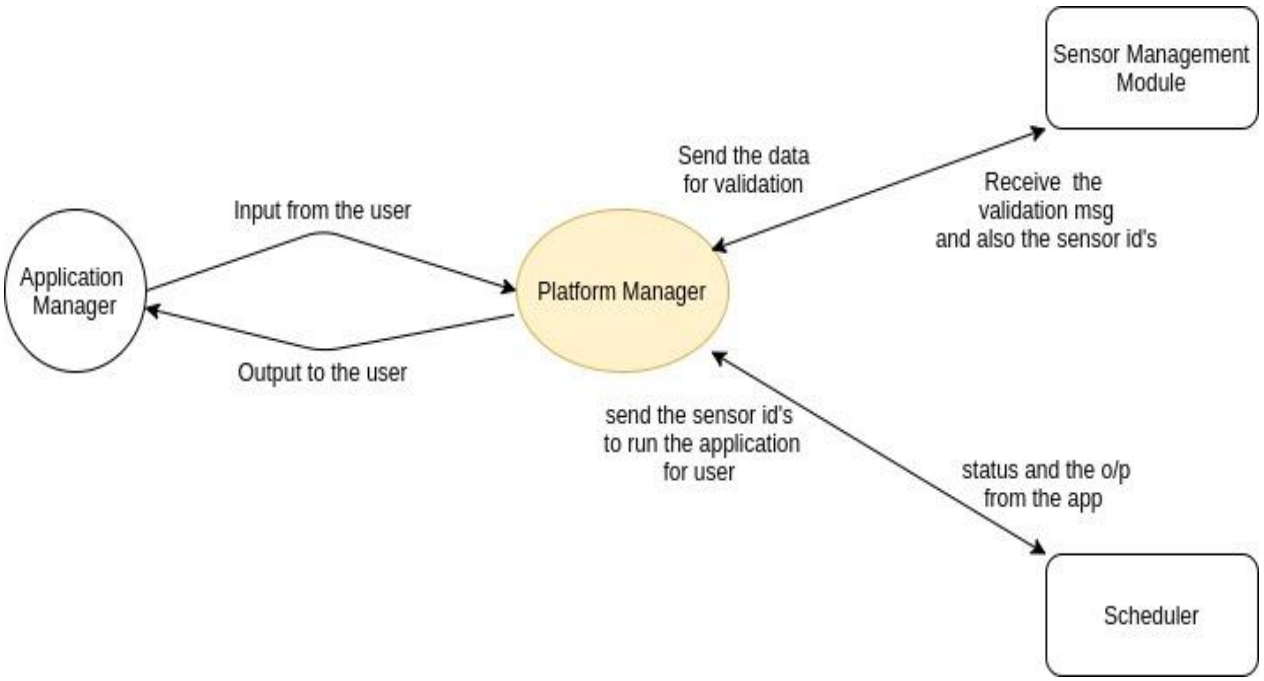
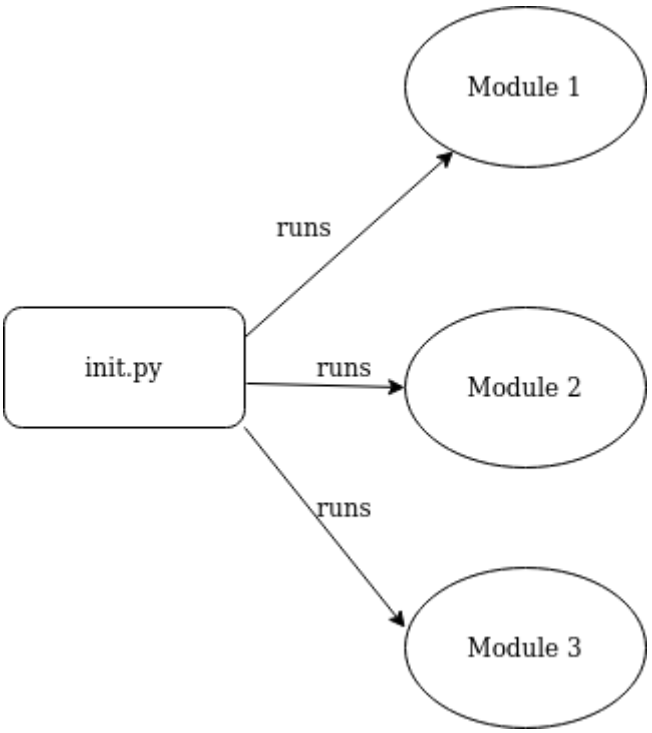


Diagram 3 - Platform Manager

- **Running Monitoring Module**
  - The module will keep polling to check the current status of all the modules.
  - In case no response is received during polling from a specific module, the last consistent state of the respective module will be loaded again from the log files created periodically.
  - The modules uses socket programming to connect to different modules present on different devices.

- **Running Bootstrap Module**
  - The init.py file is run which in turn , initializes all other modules located on same or different modules.
  - It also opens other terminal screens for different modules as and when required.
  - It uses the concept of multithreading and runs all the modules parallely.



Test Cases

1. Application Manager

A] Test Case 1 – User Not Registered

If the user/admin is not registered, then first register it.

Input – User details, as well as user role

Output- Login the user

B] Test Case 2 – User Authentication

If the user/admin is registered, then he can define its role first by choosing the option from the terminal and then login into the platform using the id and password. The platform will authenticate the user from its database. If the id

and password matches, then the user is logged into the platform otherwise an error message is printed.

Input - Login Details

Output- If the details are correct then – User is logged in , otherwise, the error message is printed on the terminal

C] Test Case 3 – If the User is Application Admin

The admin is provided with 3 options

1. To install the sensor class(type)
2. To install the sensor instance
3. To Deploy the application

Input - option 1,2,3 from the admin

Output - validation message

D] Test Case 4– If the Admin chooses option 1.

The admin will provide a file to install a sensor class(type) on the platform.

Input - sensorClass.json file

Output - Validation message

E] Test Case 5 – If the Admin chooses option 2.

The admin will provide a file to install the sensor instances on the platform.

Input - sensorInstance.json

Output - validation message

F] Test Case 6 – If the Admin chooses option 3.

The admin will provide the application zip file to deploy the application

Input - zip file of the application

Output - validation message

G] Test Case 7 – If the user chooses any wrong input at any time.

Input - Wrong Input

Output- Validation message.

H] Test Case 8 – If the User is End User

The user will provide the input to run the application. And the output will be displayed on the terminal.

Input - application name, start time, end time, location.

Output - the output that the application will return on its execution.

2. Platform Manager

A] Test Case 1 – If the message received from the application manager is for application admin and for installing sensor class(type)

The platform manager will receive the file from the application manager and then will parse and send it to the sensor manager

Input - sensorClass.json file

Output - send it to sensor manager

Input 2 - receive msg from the sensor manager

Output 2 - send that message to the application manager

B] Test Case 2 – If the message received from the application manager is for application admin and for installing sensor instance

The platform manager will receive the file from the application manager and then will parse and send it to the sensor manager

Input - sensorInstance.json file

Output - send it to sensor manager

Input 2 - receive msg from the sensor manager

Output 2 - send that message to the application manager

C] Test Case 3 – If the message received from the application manager is for application admin and for deploying the application

The platform manager will receive the zip file from the application manager and then will use it to deploy the application on the application

Input - application name

Output - send it to sensor manager to validate the sensors.

Input 2 - receive msg from the sensor manager

Output 2 - process on the msg and reply to the application manager that the application is deployed or not.

D] Test Case 4 – When the sensor instance is not available during running of the application on the platform

The user requests for running of a specified application with other details like start time, end time, location. This information is sent to the Scheduler, If a sensor is not found at the given location, the Scheduler sends us an error message, and no further action is taken. The user is displayed with the error message that the application failed to run.

Input - application name, start time, end time, location

Output - Error message received

3. Monitoring Module

A] Test Case 1 - When the module is working

The monitoring module sends a connection request to the module. If the respective module accepts the connection and receives the data sent, the monitoring module considers the module alive.

B] Test Case 1 - When the module is not working/responding/started

The monitoring module sends a connection request to the module. If the respective module has to accept the connection and receive the data sent, then the monitoring module will consider the module alive. Else, if the code crashes in any of the above steps, or is not able to complete the process - the monitoring module takes 2 possible actions :

1. If the module to be re-initiated is present on the given system, then the command to open the terminal in the same machine is given.
2. If the module to be re-initiated is present on another system, then the monitoring module tries to communicate with the initializing module present on that respective system, and a command is given to re-initialise the module.

4. Bootstrap Module

Sub Modules Present :-

Application Manager

1. Registration Module
2. Processing input from different actors

Platform Manager

1. Validation of input from Sensor Manager
2. Deploying
3. Sending the application to run on a scheduler for the user.

Monitoring Module

1. Connection with various modules
2. Re-initialisation of module
3. Connection with different system on which module is present

Interactions of the various modules :-

- ➔ The application manager will interact with platform manager for installing sensors on the platform
- ➔ The application manager will interact with platform manager to deploy application
- ➔ The application manager will interact with platform manager to run the application for the user
- ➔ The platform manager will interact with the sensor manager for validations
- ➔ The platform manager will interact with the scheduler to pass the application and the inputs required to run the application.
- ➔ The Monitoring module will interact with all the modules present in the system time to time.