

# Submarine-Exploration Application

## 1. Scenario

The Application is about exploring a special type of plant whose flower contains some medicinal properties. Usually these plants are rare and found underwater. Submarines have gone on an expedition to find them but as the plants are rare and other countries have deployed mines around the areas where they are found. We have two type of special sensors one which send the sonar data and other is a sensor that records the properties of flowers that come in its range. Two more type of sensors have been deployed to help in the submarines navigation and connectivity.

## 2. Sensors

### A. Distance Sensor

**Description :** This sensor is a distance sensor and is deployed on the base of the submarine. This captures the distance of the submarine from the land below.

**Type :** Two-way should be able to take in a command and display it on the sensor itself.

**Data Generated :** This sensor would be giving a number between 1 and 1000 which would suggest the distance of the submarine from the ground in ft.

**Rate :** The sensor would be sending values every second.

**Simulation :** 90 % (Approx) values should be within the range 201 to 1000. 10 % (Approx) values should be between 0 and 200.

### B. Sonar Sensor

**Description :** This is a Naval mine sensor taking in Sonar Data.

**Type :** One way.

**Data Generated :** vector format  $[60 * 1]$  per input.

**Rate :** Every 10 seconds.

**Simulation :** Simulate a sensor using the Sonar Dataset that has been previously provided.

### C. Flower Analysis sensor

**Description :** This is an iris-flower classification sensor.

**Type :** One way.

**Data Generated :** vector format  $[4 * 1]$  per input.

**Rate :** Every 0.5 seconds.

**Simulation** : Simulate a sensor using the Iris-Flower Dataset that has been previously provided.

### 3. Artefacts

#### Naval Mine Model

This is the model that you had made in Assignment 4. The output should be relayed on a UI. This service should run on an exclusive environment.

#### Iris Model

This is a pre trained model which on receiving an input from sensor D classifies it and returns an output :

1. Iris setosa (0)
2. Iris versicolor (1)
3. Iris virginica (2)

For more info on Dataset : [https://en.wikipedia.org/wiki/Iris\\_flower\\_data\\_set](https://en.wikipedia.org/wiki/Iris_flower_data_set)

For more info on model : [https://www.tensorflow.org/guide/premade\\_estimators](https://www.tensorflow.org/guide/premade_estimators)

This is critical and should run on the gateway.

#### Other Services / Algorithms

##### 1. Distance Alarm Service

This service accepts the output of Distance Sensor and checks its range. If the range is between 0 and 100 then it sends a message "EMERGENCY STOP" + "Value of Sensor". If the range is between 101 and 200 the it sends a message "CRITICAL" + "Value of Sensor". Else it sends the message "ALL FINE" + "Value of Sensor". This service should run on an exclusive environment.

##### 2. Emergency Notification Service

This should send emails to a list of people whenever the distance sensor reaches a value below 200.

##### 3. The Counter Service

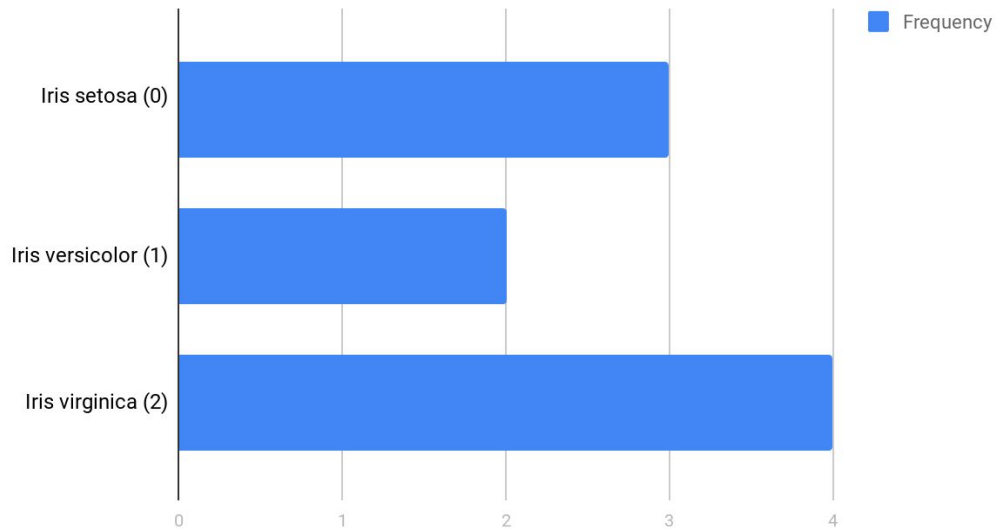
This service is simple. It reads from a file an integer adds one to it and sends it output to an output stream visible on a separate UI. This is triggered in every 30 seconds. This is also triggered when a mine is detected.

#### 4. Helper Service

This is a helper service to the iris model which reads the output stream of the Iris model and updates a UI for it. This would just be making a histogram according to what value it has received on the UI and updating it.

Eg. if 3 (0) and 2 (1) and 4 (2) have been received then UI should be

IRIS HISTOGRAM



#### 4. User Interfaces

- One UI to display a danger signal in case the Naval Rock-Mine Modal detects a mine or normal green signal in case only a rock is detected.
  - One UI to display the output of Counter Service.
  - One UI to display the output of the Helper service as requested.
- The output of The Distance Service should be visible on the sensor only.

#### 5. Flow

- Build and package the application.
- Deploy the Application.
- Check that all servers as per bootstrap Config are up.
- Check if all app services and gateways are up
- Have a name based mechanism to deploy sensor and services on Gateways.
- A UI should be capable of initiating the following services of the Application :
  - Naval Mine detection service
  - Iris Flower Service along with its helper service
  - Distance Alarm Service
- The UI should also be capable of disabling the above services also.

- h. The Emergency Notification service should keep running once the Application Starts. There should be a mechanism to disable it from a UI.
- i. The counter service should start according to a schedule and should be triggered in every 30 seconds.

## 6. Testing

- a. **Action** : Turn the server with the Naval Mine model which is accepting data from the sonar sensor stream.  
**Test** : This model should be re-instantiated at some other location within one minute and should be showing output on the UI. The UI should also tell that the model had failed at one location.
- b. **Action** : Stop the container containing the “Iris Model” in the gateway that it is running.  
**Response** : A new container is up within one minute and the UI reflects the same.
- c. **Action** : The shared server that contains the Emergency Notification and the Counter Service is turned off.  
**Response** : A replica server comes up within 1 minute 30 seconds and platforms starts functioning normally. The UI shows the failure and also reports that the services have restarted.