

EMERTXE TRAINING PROJECT DOCUMENTATION FRAMEWORK
REQUIREMENTS & DESIGN DOCUMENT

Module – IoT Cloud

**Street Light Control and
Monitoring**

Contents

1 Abstract.....	1
2 Requirements.....	2
2.1 Device Requirements.....	2
2.2 Gateway Requirements.....	3
2.3 Cloud Requirements.....	4
3 Prerequisites.....	6
4 Design.....	7
5 Sample Output.....	11
5.1 Station.....	11
6 Artifacts.....	13
6.1 Skeleton Code.....	13
6.2 References.....	13

1 Abstract

As time passes there an increasing demand of electricity, leading to many environmental issues like Increase in carbon footprint, climate changes, light pollution and also its over usage lead to increase in energy tariff due to demand and supply issues

One possibility is reduce the power usage in all the sectors like Housing, Industrial and Corporate which seems to be bit difficult because of our life style. This requires some awareness programs within us.

The other possibility is to identify various area in public sectors, where we can conserve energy to solve the above concerns. One such this is to avoid the wastage of power in street lights. Well this could be done manually, but the efficiency matters!!

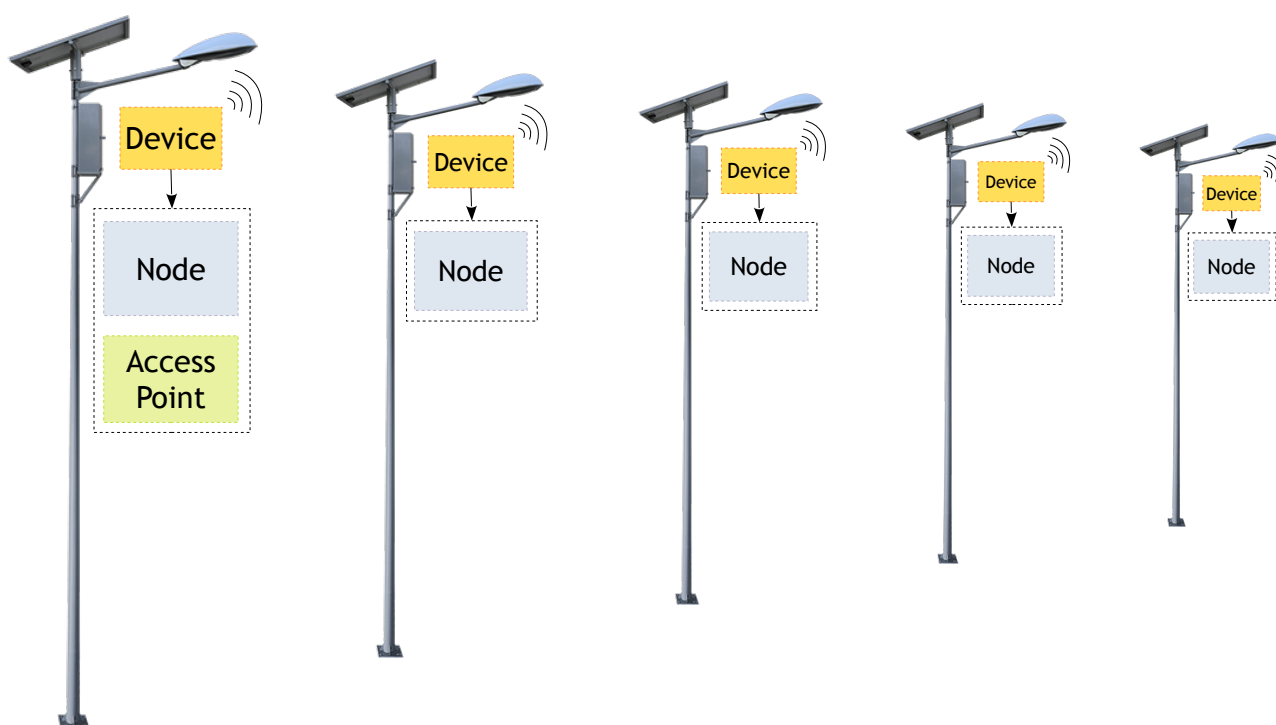
So the idea is to fix a device on every pole which would monitor the lighting activity and send it to the cloud for analytic purpose. There should be a option to control the lights as needed from the cloud.

2 Requirements

This project should have 3 main components

- The Device
- The Local Gateway
- The Cloud

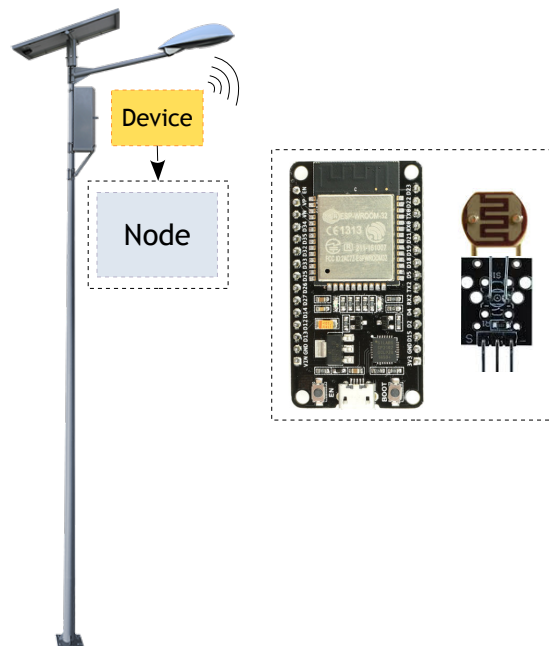
The below are the network requirements needed in this project



2.1 Device Requirements

- Should scan for the available networks
- Show all possible networks including the your access point so that the user can select it
- Make sure the password entered are represented with asterisk
- Once the connection is established show all the network details like
 - Server Information
 - The SSID
 - Server IP

- Station Information
 - Name (Generate unique name with the help of MAC id, consider lower 2 bytes)
 - IP Address
 - Signal Strength
 - Connection Status



2.2 Gateway Requirements

- The gateway should support both
 - Station Mode
 - Access Point Mode

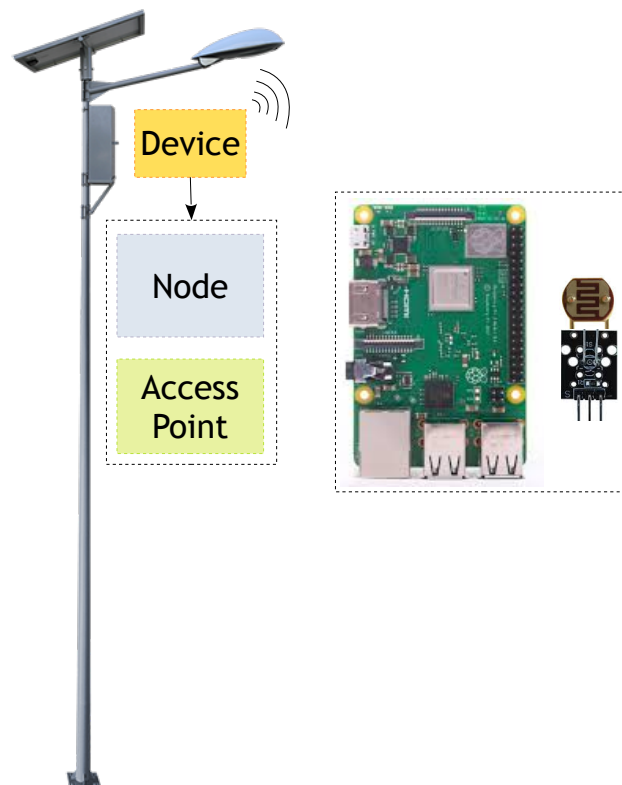
Station Mode

- Should scan for the available networks
- Show all possible networks so that the user can select a network to connected
- Make sure the password entered are represented with asterisk
- Once the connection is established show all the network details like
 - The SSID it is connected to
 - Stations IP address
 - Signal Strength

Once the your gateway as station connected to internet gateway it should configure itself as access point too

Access Point Mode

- It should generate an SSID with the help of its MAC address (consider 2 lower bytes)
- Allocate a static IP so that the local stations can connect
- Display the server status and wait for the nodes to get connected
- Display the connected node information

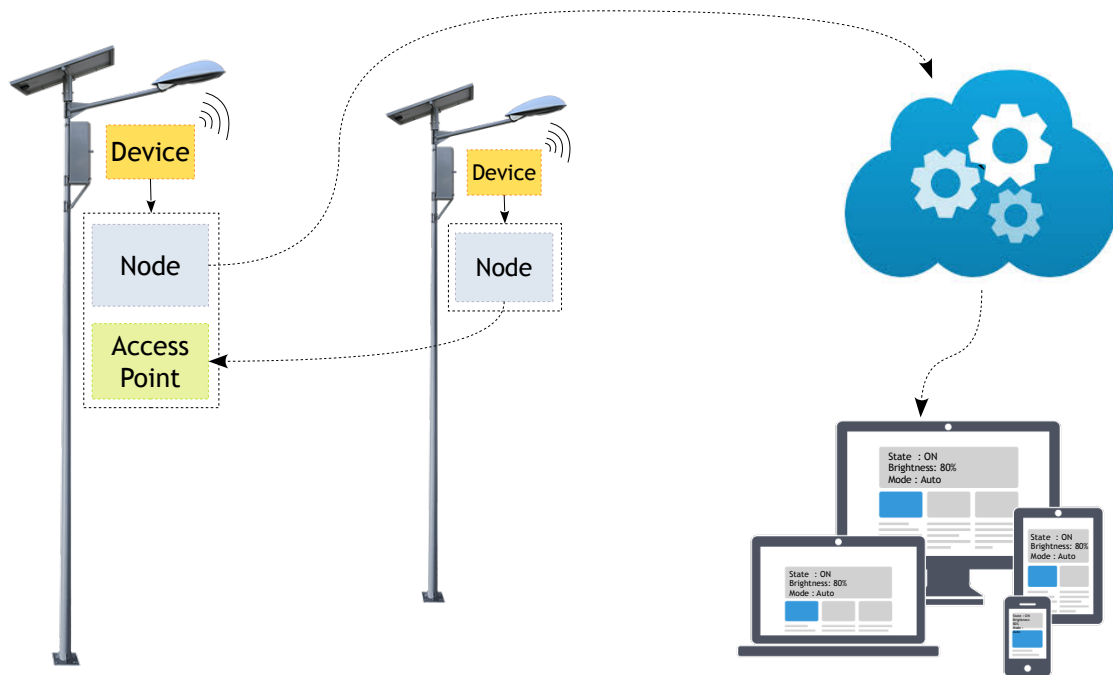


as said the above was about the network configuration, apart form all these the device is responsible to sense street light activity and send it to the cloud for analysis

2.3 Cloud Requirements

- The cloud should be configured to interact with 2 entities
 - The local gateway
 - Should provide interface for triggers
 - Store all the data received
 - Web application

- The cloud should provide an API so that the application can fetch the data for presentation as well as control



3 Prerequisites

- Embedded C, Python programming
- Understanding of Wifi Modes
- Working knowledge on Raspberry PI boards
- AWS functions from IoT Core like Lambda, Dynamo DB, API Gateway

4 Design

- Block Diagrams - Stations

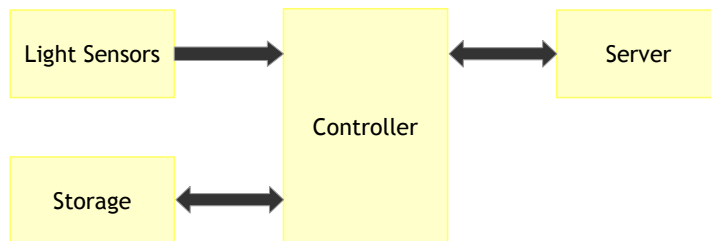


Fig 4 1: Block Diagram - Level 0

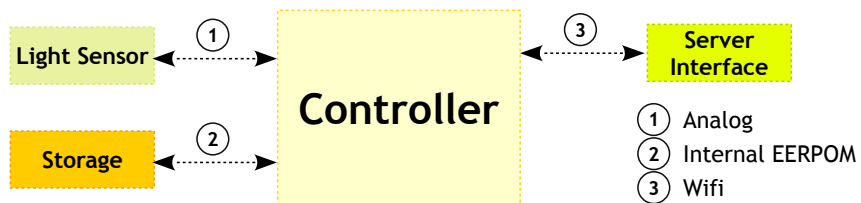


Fig 4 2: Block Diagram - Level 1

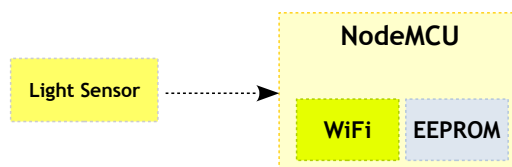


Fig 4 3: Block Diagram - Level 2

- Block Diagrams - Gateway

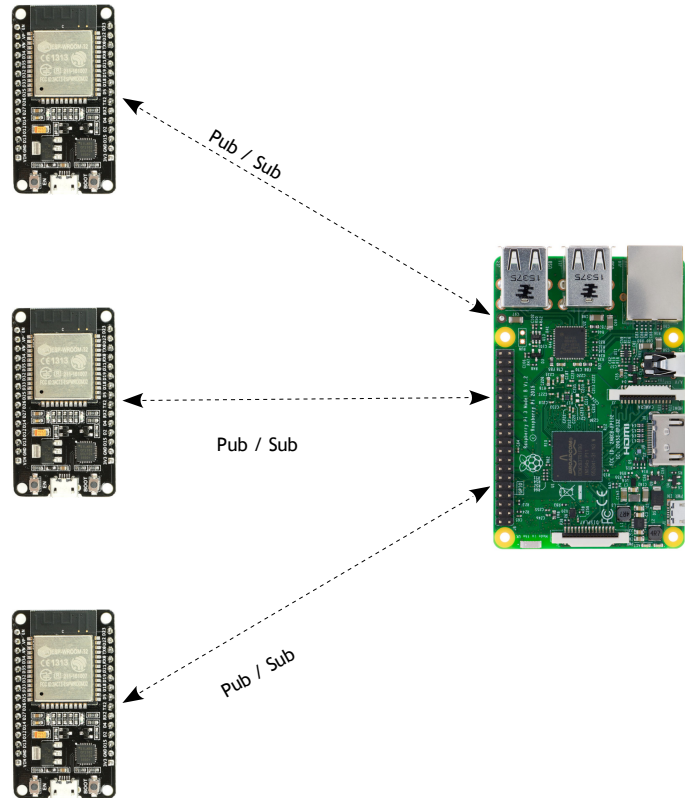


Fig 4 4: Device interaction with local gateway

- Block Diagrams – Cloud

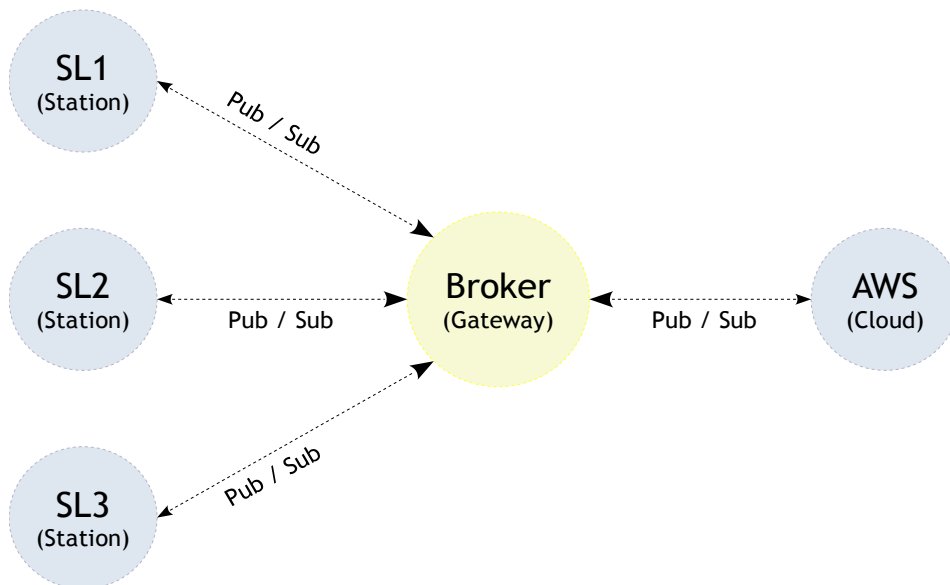


Fig 4 5: Device to Gateway to Cloud

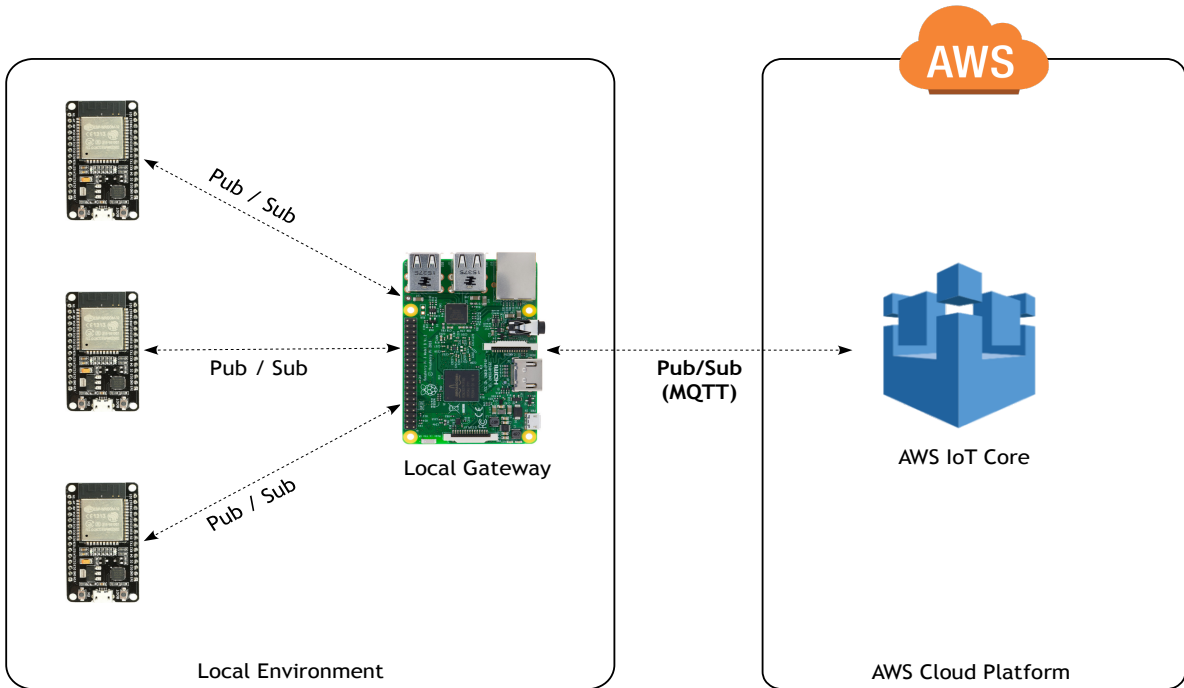


Fig 4 6: Local Gateway to Cloud

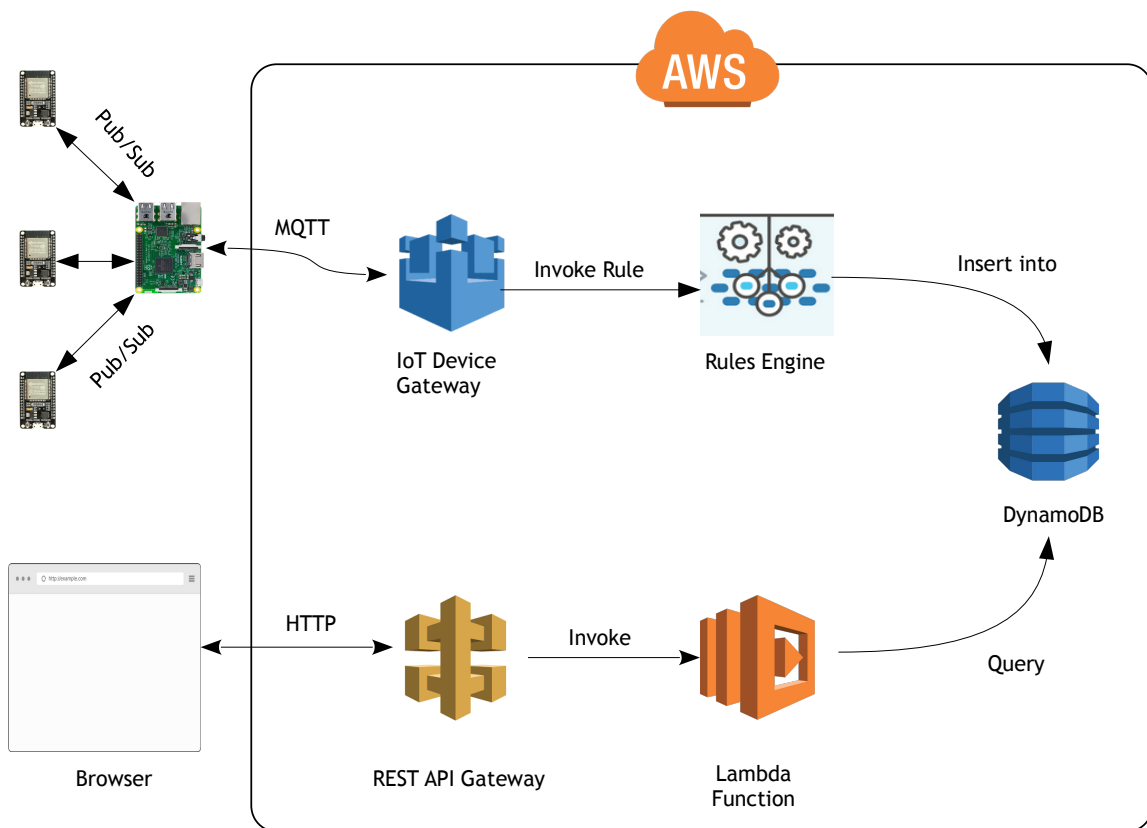


Fig 4 7: Complete Architecture

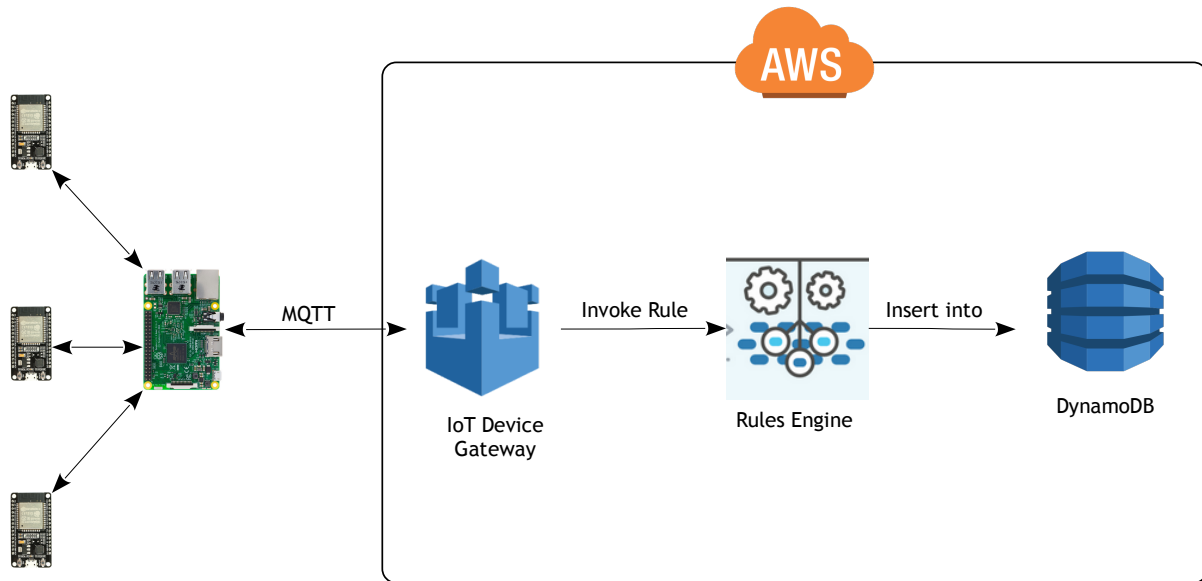


Fig 4 8: Handling Incoming device data

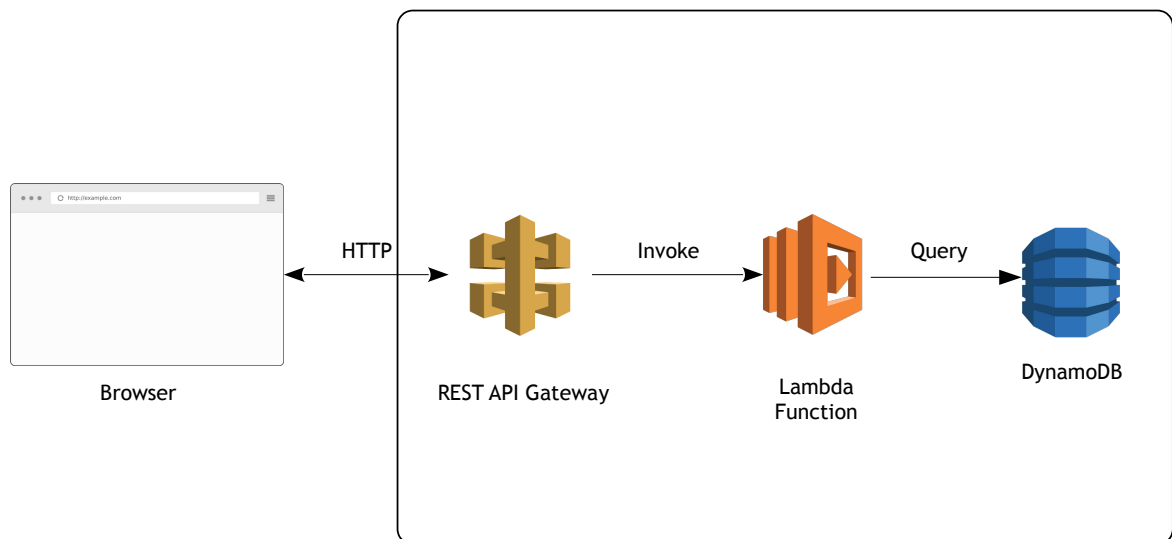


Fig 4 9: Web Application Interaction

5 Sample Output

5.1 Station

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
confsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:808
load:0x40078000,len:6084
load:0x40080000,len:6696
entry 0x400802e4

|--- Starting the Network Scan ...
```

Fig 5.1 1: Scan for available network

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
confsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:808
load:0x40078000,len:6084
load:0x40080000,len:6696
entry 0x400802e4

|--- Starting the Network Scan ... Done

|--- Found 3 Networks

1. StreetLightGateway
2. Wisdom
3. Emertxe

Please select the required Network:
```

Fig 5.1 2: Display available networks and prompt for selection so that user can select ESP Server

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
confsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:808
load:0x40078000,len:6084
load:0x40080000,len:6696
entry 0x400802e4

|--- Starting the Network Scan ... Done

|--- Found 3 Networks

1. StreetLightGateway
2. Wisdom
3. Emertxe

Please select the required Network: 1
Please enter authentication key:
```

Fig 5.1 3: Selecting the server, prompt for password

```
rst:0x1 (POWERON_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:808
load:0x40078000,len:6084
load:0x40080000,len:6696
entry 0x400802e4

|--- Starting the Network Scan ... Done

|--- Found 3 Networks

1. StreetLightGateway
2. Wisdom
3. Emertxe

Please select the required Network: 1
Please enter authentication key: *****
|--- Attempting to connect StreetLightGateway
...
|--- Connected to StreetLightGateway

|--- Connection Information
Server      : StreetLightGateway
Server IP Address : 192.168.1.1
Client     : Node C21A
Client IP Address : 192.168.1.2
Signal Strength : -50 dBm
State      : Connected
|--- Done
```

Fig 5.1 4: Connection information with the server

6 Artifacts

6.1 Skeleton Code

The skeleton code is a very interesting concept used in Emertxe. By looking into the skeleton code, you will get a clear picture into converting the given requirement into a working solution. This will also take care of important aspects like modularity, clean coding practices, re-usability etc.

- TBD

6.2 References

- <https://ieeexplore.ieee.org/document/7489455>