



# EFR32 IOT GATEWAY CLOUD

Design Document

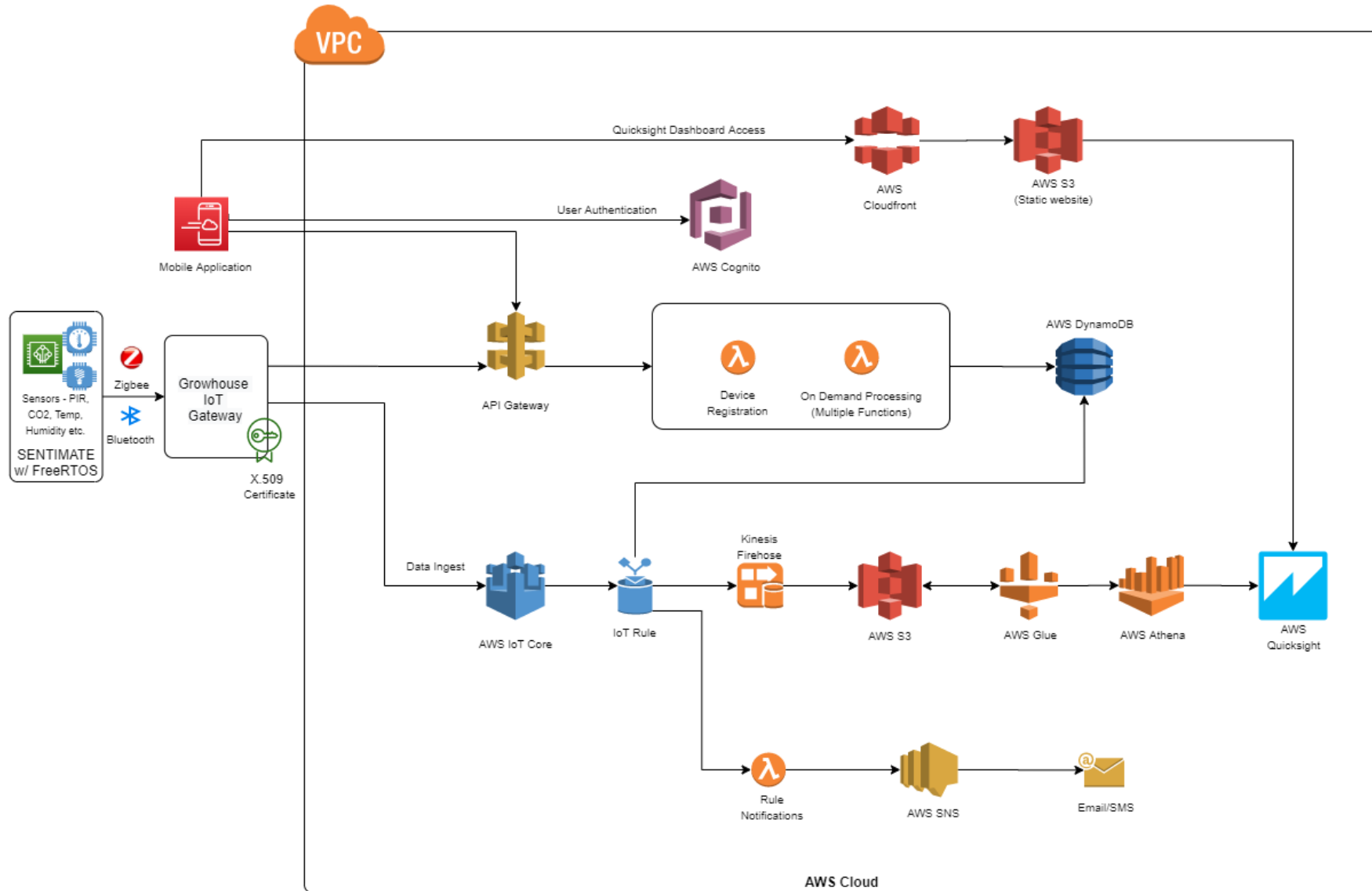


## Contents

---

EFR32 IoT Gateway Block Diagram.....	2
Gateway Registration.....	3
Sensor Registration .....	3
Gateway Deletion.....	3
Sensor Deletion .....	4
Telemetry.....	4
IoT Rule for Threshold value .....	4
User Management.....	4
Quicksight Dashboard .....	5

## EFR32 IoT Gateway Block Diagram



## Gateway Registration

---

1. The gateway receives the request to register the gateway from the mobile app over BLE.
2. The gateway calls the AWS API Gateway 'creatething' API to register.
3. The API invokes a lambda function that will:
  - a. create a thing in AWS IoT Core
  - b. generate necessary certificates
  - c. create SNS subscription for SMS and Email
  - d. create a gateway entry in the AWS DynamoDB table
4. All of the necessary details are sent in the API response to the gateway which stores them for future use.
5. Gateway sends the "success/failure" response to the mobile application over BLE

## Sensor Registration

---

1. The gateway receives the request to register the sensor from the mobile app over BLE.
2. The gateway will search for the available Zigbee devices and send the list to the mobile application.
3. User will select which sensor to provision and the payload will be sent to the gateway.
4. The gateway calls the AWS API Gateway 'creatething' API to register.
5. The API invokes a lambda function that will:
  - a. create a thing in AWS IoT Core
  - b. create a sensor entry in the AWS DynamoDB table
6. All of the necessary details are sent in the API response to the gateway which stores them for future use.
7. Gateway sends the "success/failure" response to the mobile application over BLE

## Gateway Deletion

---

1. The gateway to be deleted is selected by the user in the mobile app.
2. The mobile app searches for the selected gateway over BLE.
3. Once the gateway is found, the mobile application calls the AWS API 'deletething' to delete that gateway from the cloud
4. The API will invoke a lambda function that will:
  - a. remove any certificates associated with that gateway
  - b. delete any sensors associated with that gateway
  - c. remove the entries from the AWS DynamoDB table
5. The response is collected by the mobile application
6. The mobile application then sends a 'gateway deletion' request to the gateway.
7. The gateway will clear all the details it had stored when registering the gateway
8. If the gateway is not available over BLE, the mobile application will call the API 'deletething' and then the gateway needs to be hard-reset.

## Sensor Deletion

---

1. The sensor to be deleted is selected by the user in the mobile app.
2. The mobile app searches for the gateway which is associated with sensor over BLE.
3. Once the gateway is found, the mobile application calls the AWS API 'deletething' to delete that sensor from the cloud
4. The API will invoke a lambda function that will:
  - a. delete any sensors associated
  - b. remove the entries from the AWS DynamoDB table
5. The response is collected by the mobile application
6. The mobile application then sends a 'sensor deletion' request to the gateway.
7. The gateway will clear all the details it had stored when registering the gateway
8. If the gateway is not available over BLE, the mobile application will call the API 'deletething' and then the gateway needs to be hard-reset.

## Telemetry

---

1. The gateway connects to the AWS IoT Core securely using the certificates it stored while gateway registration
2. Once connected, the gateway sends the telemetry data to AWS IoT Core using MQTT protocol.
3. When the data reaches cloud, IoT Rules are invoked, which check for value thresholds and send notifications via Email/SMS if required.
4. Another rule is invoked to write this data in the DynamoDB table.
5. A separate rule is invoked which puts this data received in the Kinesis Firehose stream.

## IoT Rule for Threshold value

---

1. Rule will be invoked once cloud will receive data on MQTT topic from provisioned sensor.
2. Rule will execute Lambda function which will check current threshold value for specific sensor.
3. Lambda function will also check notification value for Email and SMS.
4. Based on threshold value and notification value user will get alert notification via Email/SMS if value would be out of range.

## User Management

---

- For the user management, AWS Cognito service is used which allows user sign/signup flow easily.
- When the user opens the Mobile App, he/she can create an account which whose details will be stored in the Cognito User Pool.

## Quicksight Dashboard

---

1. Quicksight dashboard is used to prepared History chart for specific sensor.
2. Dashboard further getting publish using aws public domain url and same url is being used by mobile application to display the data.
3. IoT Rule will be invoked when sensor data comes to aws cloud and that data get pass to kinesis firehose to store in S3 bucket.
4. S3 bucket data will be read by Glue to convert into specific format and this converted data again publish to Athena.
5. Athena will store data into workgroup which will be used by Quicksight to prepare dashboard.
6. Once this setup done, need to create quicksight analyses for different kind of dashboard look.