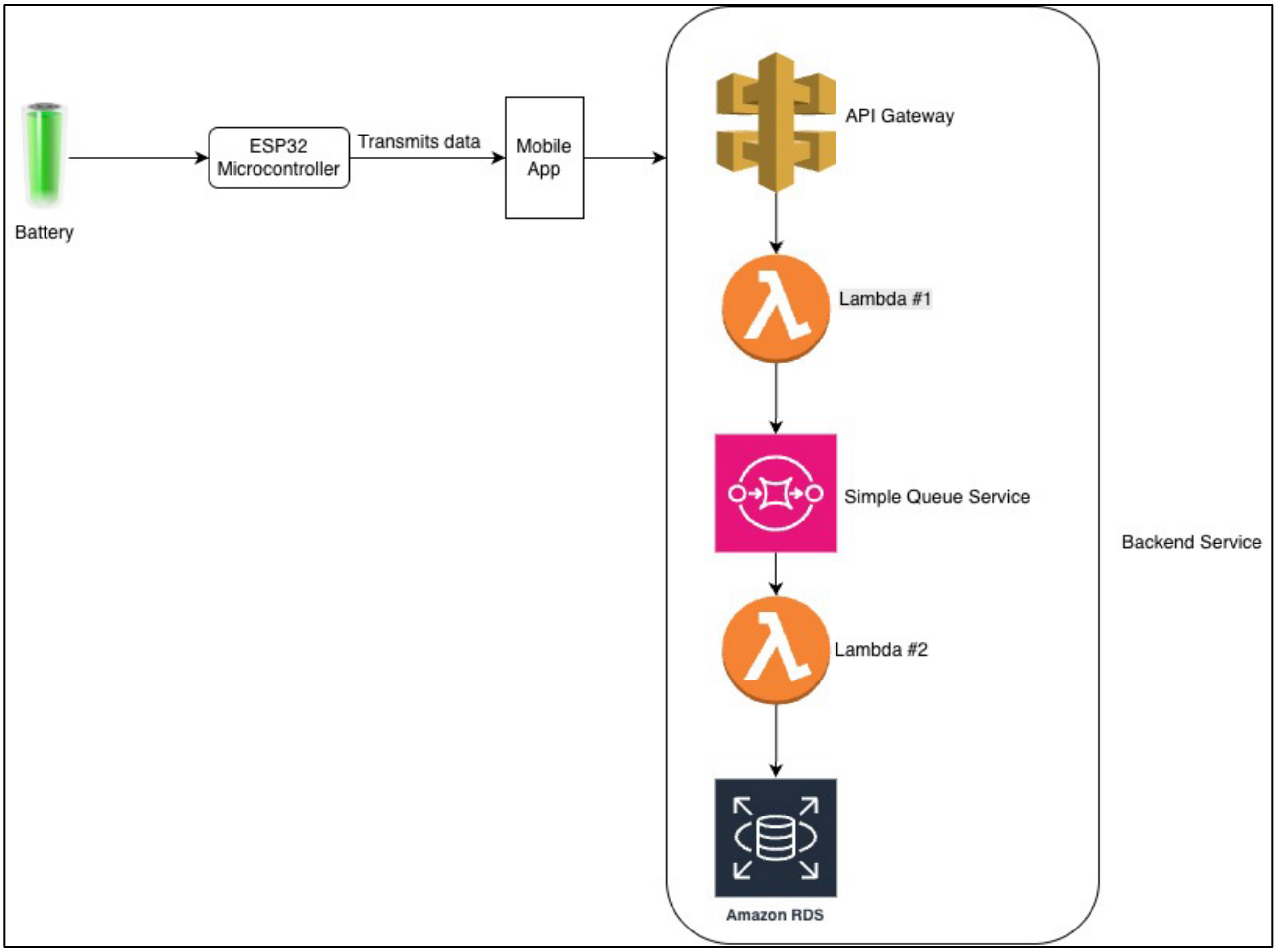
ESP32 based Microcontroller Dashboard

Group 2

Sponsor : FeatherStill

Architecture Overview

High level flow (currently decided / tentative)



Explanation of Each Component

1. Battery

- Powers the ESP32 controller.
- Not part of cloud architecture, but the data source depends on it.

2. ESP32 Microcontroller

- Collects telemetry (temperature, voltage, current, etc.).
- Sends structured JSON data.
- Communicate via Bluetooth (to mobile app).

3. Mobile App

- Receives data from ESP32.
- Packages it into JSON.
- Sends secure HTTPS request to backend.

  Why it exists:

  - The users can see the real time data using Bluetooth
  - Handles authentication

4. API Gateway

Acts as the **entry point to AWS**. Handles authentication (API key/JWT)

Responsibility: Secure front door to backend.

5. Lambda 1 (Ingestion Lambda)

Triggered by API Gateway.

What it does:

- Parses incoming JSON
- Pushes message into SQS
- Returns success response quickly

6. Amazon SQS (Simple Queue Service)

Message queue between Lambda 1 and Lambda 2.

What it does:

- Buffers incoming telemetry
- Absorbs traffic spikes
- Ensures no data loss
- Enables retry mechanism

If 10,000 devices send data simultaneously:

- SQS stores messages safely
- Lambda 2 processes them gradually

7. Lambda 2 (Worker Lambda)

Triggered automatically by SQS.

What it does:

- Pulls messages from queue
- Connects to database

- Inserts telemetry into RDS
- Handles retry on failure

8. Amazon RDS

Managed relational database (Postgres/MySQL).

Stores data in form of parameters which are:

Parameter	**Data type**
Customer_id	String
Module_id	String
Module_type	String
Controller	String
Time_utc	ISO String
Time_local	ISO String
ts	Number (ms)
year	String
Month	String
Day	String
Temp_c1	Float
Temp_c2	Float
V1 to V16	Float
Pack Pin voltage	Float
Output current_A	Float