Project Report: CBR3A – UB Site

Activity 1

Developing and Testing a New BLE Scanner for Cloud Integration

Design

The objective of this activity was to create a simplified BLE scanner capable of detecting nearby beacons and sending the data directly to the cloud. Unlike previous implementations, this new scanner was designed to exclude telemetry and MQTT features, focusing solely on beacon detection and cloud communication. Additionally, efforts were made to troubleshoot and update the telemetry app and the previous scanner due to deprecated API URLs.

Method

The flow of the BLE scanner app was structured as follows:

1. Beacon Detection: The app scans for nearby BLE beacons in its vicinity.
2. Filtering: It filters the detected beacons to select only specific ones based on predefined criteria.
3. Cloud Integration: The filtered beacon data is transmitted to the cloud via a stable and secure connection.
4. Cooldown Mechanism: To avoid overloading the cloud with redundant data, the app enforces a 10-second cooldown period between consecutive data transmissions.
5. Scheduled Operations: The app is programmed to automatically shut down during the night and restart in the morning, conserving energy and resources.

Implementation

- New Scanner Development:

A new BLE scanner was implemented with a single feature to detect beacons and send the detection data to the cloud. The scanner's functionality was tested to ensure compatibility with the cloud API.

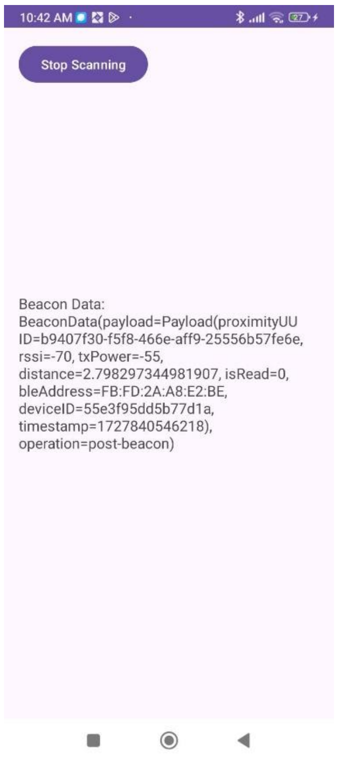
The new system was stripped of unnecessary telemetry and MQTT features to streamline its operation and improve reliability.

- Fixing Legacy Systems:

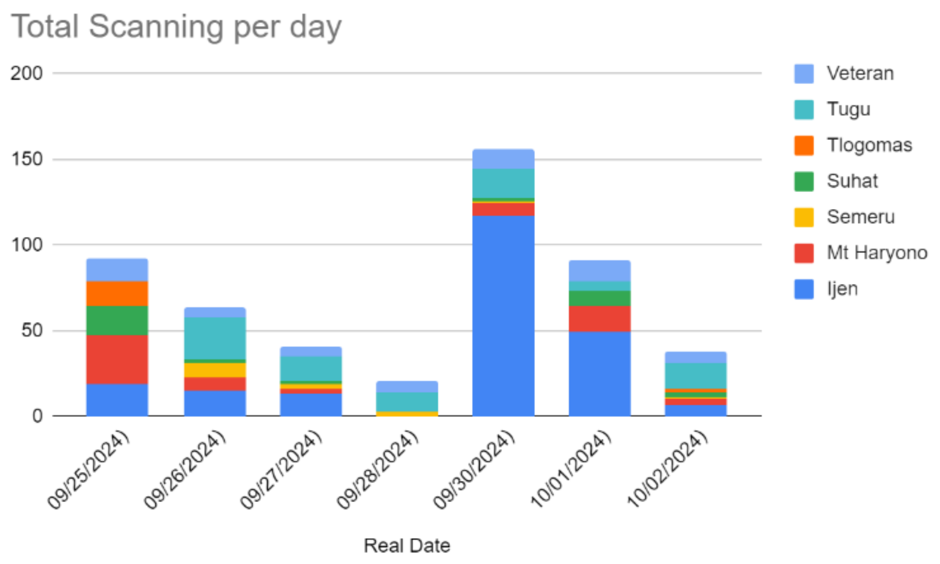
  The telemetry app and old scanner were updated with the new API URLs to address the issue of deprecated endpoints.

  Modifications were tested to ensure the data flow from the old systems to the cloud resumed without disruptions.

Experiment



```
Beacon Data:
BeaconData(payload=Payload(proximityUU
ID=b9407f30-f5f8-466e-aff9-25556b57fe6e,
rssi=-70, txPower=-55,
distance=2.798297344981907, isRead=0,
bleAddress=FB:FD:2A:A8:E2:BE,
deviceID=55e3f95dd5b77d1a,
timestamp=1727840546218),
operation=post-beacon)
```



The app flow was tested under real-world conditions. The scanner successfully detected nearby beacons, filtered them based on the set criteria, and transmitted the data to the cloud at controlled intervals. The cooldown mechanism worked as intended, preventing excessive data uploads, while the automatic shutdown and restart ensured resource optimization.