Internet of things

IoT Homework
Exercise 1

Authors:
Daniel Shala - 10710181
Jurij Diego Scandola - 10709931

Academic Year:
2024 - 2025

1 System design

1.1 Forklift Hardware

Each forklift is equipped with the following components to support real-time localization and status monitoring:

- **ESP32** processing unit — serves as the main microcontroller for sensor integration and communication.
- **IMU (Inertial Measurement Unit)** with accelerometer and gyroscope — used for motion tracking, distance estimation, and detecting sudden changes in acceleration (potential collisions).
- **LoRa module** — provides long-range, low-power wireless communication with the LoRa gateway for outdoor tracking and data transmission.
- **BLE module** — scans fixed BLE beacons in the underground warehouse to enable indoor positioning through trilateration.
- **GPS module** — enables accurate outdoor positioning when forklifts are outside the warehouse.
- **Accelerometer -** for monitoring speed, distance with built-in impact detection

1.2 BLE Beacon Setup

To enable accurate indoor localization in the underground warehouse, a BLE-based positioning system is deployed:

- **Fixed BLE Beacons**: Installed at known coordinates throughout the $500\,\text{m}^2$ indoor warehouse. Each beacon periodically broadcasts its ID and signal strength.
- **BLE Trilateration (Backend)**: The ESP32 on each forklift scans for nearby beacons and sends RSSI values to the backend, where trilateration is performed to compute the forklift's indoor position.
- **Beacon Placement Strategy**: Beacons are evenly distributed (e.g., in a grid or triangle layout) to ensure signal coverage and trilateration accuracy with minimal blind spots.

1.3 Communication Strategy

The IoT system adopts a hybrid communication approach tailored for both indoor (underground) and outdoor environments:

- **LoRa Dual-Frequency Communication**:
  - **Sub-GHz Frequency (e.g., 433 MHz)** for **underground indoor communication** due to better wall and ground penetration, ensuring stable connectivity within the 500 m^2 underground warehouse.

  - **Higher LoRa Frequency (e.g., 868 MHz or 915 MHz)** for **outdoor yard communication**, offering better throughput and range across the 1 km^2 area.

- **BLE (Bluetooth Low Energy)**:
  - Used exclusively indoors for proximity-based positioning via trilateration with fixed BLE beacons.
  - The ESP32 scans for beacon signal strengths (RSSI) and sends them to the backend for precise location computation.

1.4 Data Transmission Frequency

To balance real-time monitoring with energy efficiency, the following data transmission strategy is adopted:

- **Position Updates**:
  - Every 5 seconds during active movement.
  - Every 30 seconds when stationary.
- **Impact or Collision Events**:
  - Sent immediately upon detection using interrupt-driven logic.
- **Aggregated Daily Metrics** (distance, speed):
  - Sent once at end of shift or when docked.

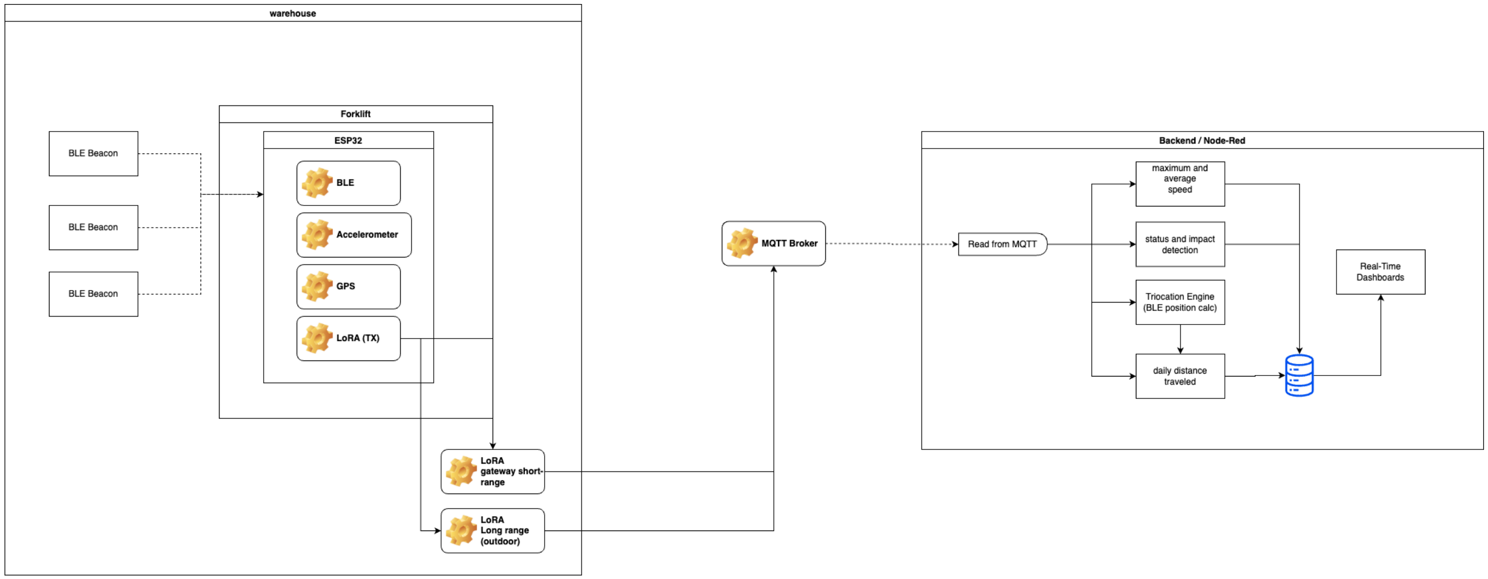
```
Setup:
    Initialize BLE module
    Initialize GPS module
    Initialize LoRa module
    Initialize accelerometer with impact
    interrupt
    Set timer to trigger every 5 seconds
    for data collection
    Attach interrupt to accelerometer
    impact detection pin
Variables:
    positionData
    motionStats
    impactDetected = false
On Accelerometer Impact Interrupt:
    impactDetected = true
```

```
Every 5 Seconds (Timer Trigger):
    if underground:
        positionData = getBLEPosition()
    else:
        positionData = getGPSPosition()
    motionStats = computeSpeedAndDistance()
    impact = null
    if impactDetected:
        impact = { time: now(),
            acceleration: getCurrentAccel(),
            distance: distance,
            location: positionData }
    payload = { timestamp: now(),
        forklift_id: DEVICE_ID,
        position: positionData,
        speed: motionStats.speed,
        distance_traveled:
    motionStats.distance,
        impact: impact }
    sendViaLoRa(payload)
```

1.5 Backend Architecture

The backend system is designed for low-latency data ingestion, efficient processing, scalable storage, and real-time visualization. It integrates with LoRa gateways and BLE trilateration logic to provide complete forklift tracking and telemetry.

- **1. Data Ingestion**
  - **LoRa Gateway → MQTT Broker:** Each forklift sends telemetry data (BLE RSSI, GPS, accelerometer) via LoRa to a gateway.
  - **MQTT Topics:** The gateway publishes messages to specific MQTT topics (e.g., `forklift/ID/data`).
  - **MQTT Broker:** A lightweight broker (e.g. Mosquitto) handles the stream of incoming data from multiple forklifts.
- **2. Data Processing**
  - **Stream Processor:** The backend subscribes to MQTT topics and:
    - Parses incoming sensor data.
    - Performs BLE trilateration using fixed beacon RSSI values.
    - Merges GPS (if outdoor) with trilaterated coordinates (if indoor).
    - Detects impacts based on accelerometer sensor data.
    - Aggregates metrics (distance, average/max speed, impact count).
- **3. Data Storage**
  - **Time-Series Database:** stores time-based telemetry data (positions, speeds, sensor values).
  - **Relational Database:** stores metadata about forklifts, BLE beacon positions, maintenance records, etc.
- **4. Visualization and Monitoring**
  - **Web Dashboard:**
    - Real-time forklift positions on a map (split indoor/outdoor).
    - KPIs: distance traveled, speed, impact alerts, battery levels.



warehouse
BLE Beacon
BLE Beacon
BLE Beacon
Forklift
ESP32
BLE
Accelerometer
GPS
LoRA (TX)
LoRA gateway short-range
LoRA Long range (outdoor)
MQTT Broker
Backend / Node-Red
Read from MQTT
maximum and average speed
status and impact detection
Triocation Engine (BLE position calc)
daily distance traveled
Real-Time Dashboards