DYNALITE

DYNAMIC VISUALISATION OF ROOM OCCUPANCY

Group 7

https://www.evantay.com/tech/dynalite/

Evan, Matthew, Melodies, Joyce

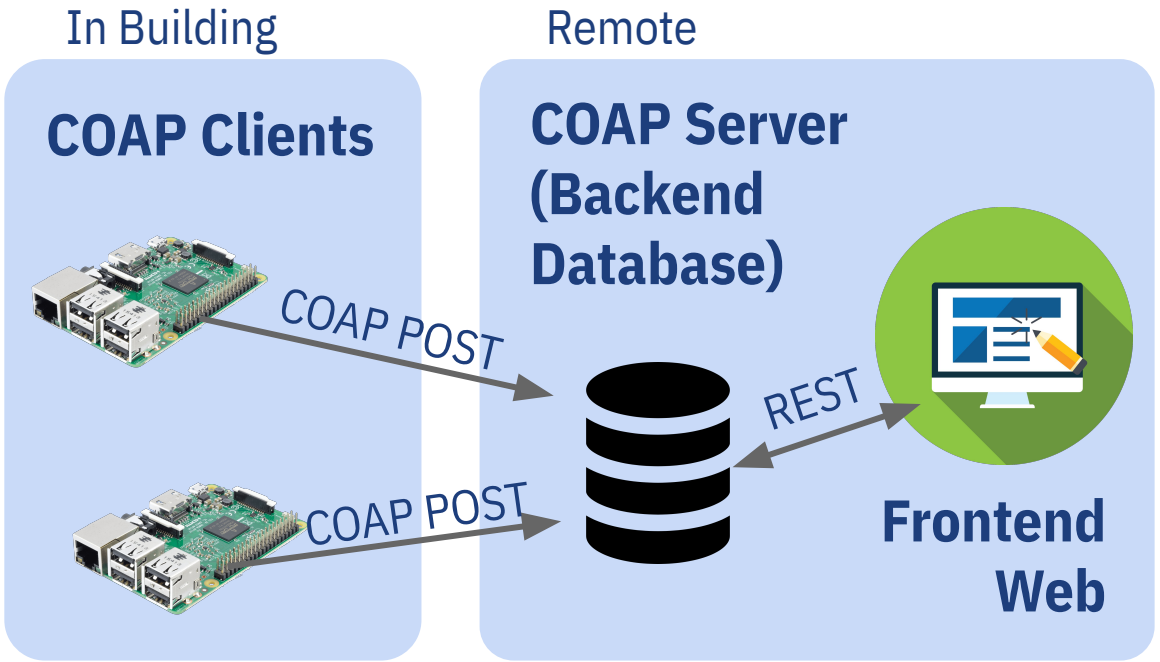
The Problem

- In Residential Colleges, it is inconvenient to figure out which lounges are available
  - Residents have to check each floor at a time

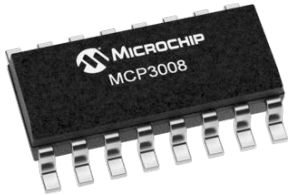
The Solution

- Use light sensors to determine room occupancy
  - Light is on → lounge is in use
  - Data visualisation of room usage over time (determine peak periods)

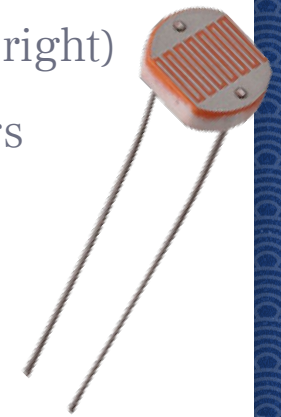
Tech Stack


In Building
Remote
COAP Clients
COAP Server
(Backend
Database)
COAP POST
COAP POST
REST
Frontend
Web

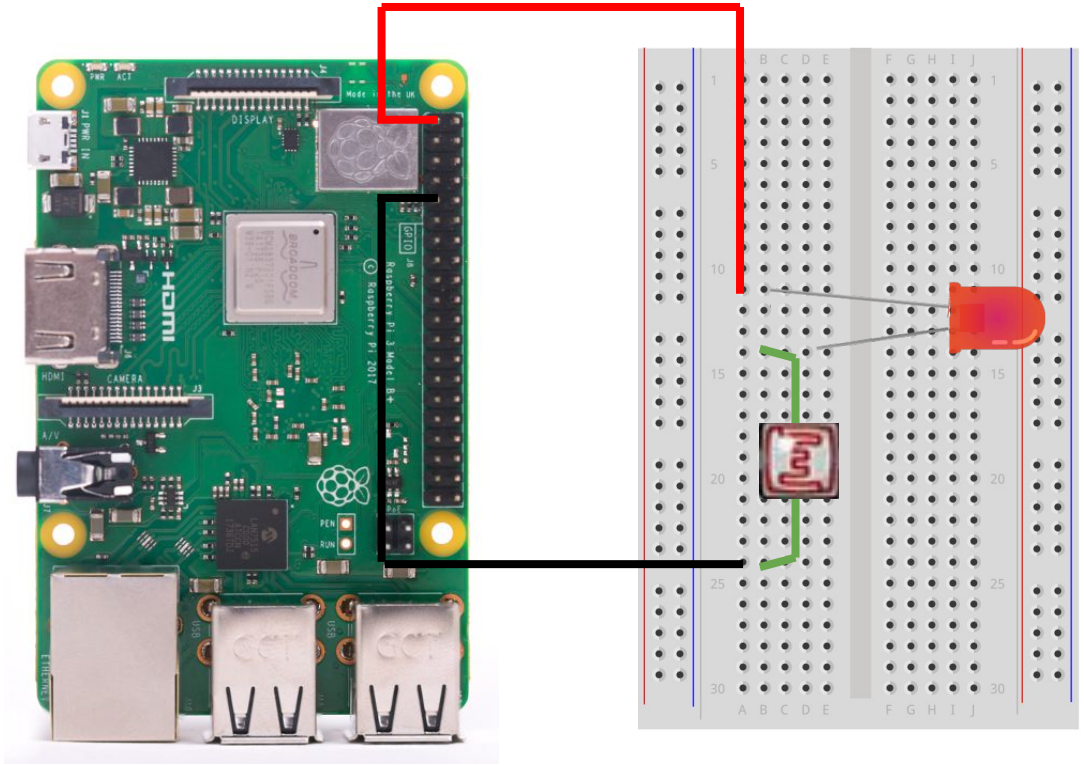
Raspberry Pi + LDR



- Sensor used: **Photoresistor (LDR)**
  - Higher light intensity → Lower resistance
  - Raspberry Pi reads 0 (no light) or 1 (bright)
- Raspberry Pi can only take **digital** readings
  - *Extension:*
    ADCs or Capacitors will have to be used to **quantify amount of light**
  - But ADCs and Capacitors used failed to function properly with the LDR.



Raspberry Pi + LDR



Raspberry Pi + LDR

- Light reading sent to COAP Server every 60s
  - light readings (0 or 1) taken in every 3s
  - Total: 20 readings per minute
  - Take the mean of these 20 readings (rounded) as “isOccupied” boolean to server.
- Flush-read one light value
  - Set GPIO to OUTPUT, **flush to 0**
  - **Wait** for photoresistor to adjust
  - Set GPIO to INPUT, **read value**

Data Security

- **AES-256** to encrypt the payload
  - Server and RPIs share **symmetric key**
  - Application-level encryption

Why not DTLS/ OSCOAP?

- **txThings**: **no DTLS/ OSCOAP** support
- **aiocoap**: has DTLS/OSCOAP support, but
  - Lack of support for COAP Server

COAP Client

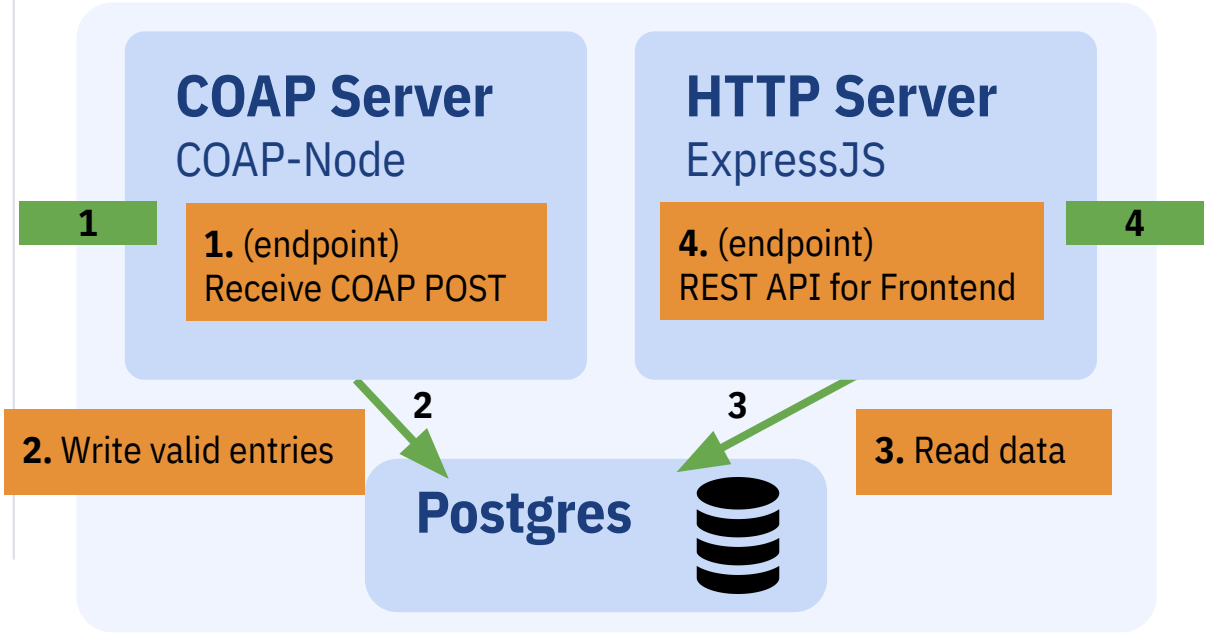
<->

COAP Server

- COAP POST sent every minute
  - Uses aiocoap library (asyncio COAP)
- CoAP SERVER (URI: coap://www.evantay.com/)
  - Will always reply so long as it receives the CoAP message:
  - Error 401, Error 400, Error 500 or
  - Success 200

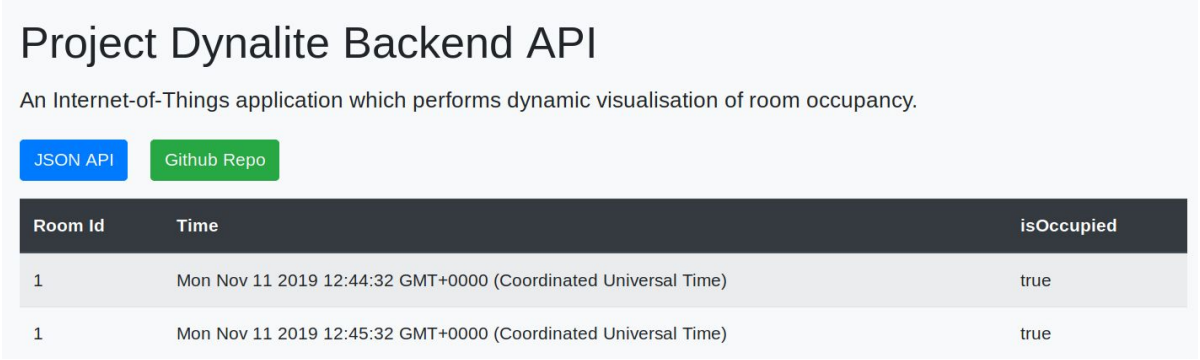
Backend Web Server

- 2 back-end NodeJS servers



Backend Web Server

- **Defined Model** for room occupancy data
  - Using Sequelize ORM (with Postgres dialect)



Project Dynalite Backend API

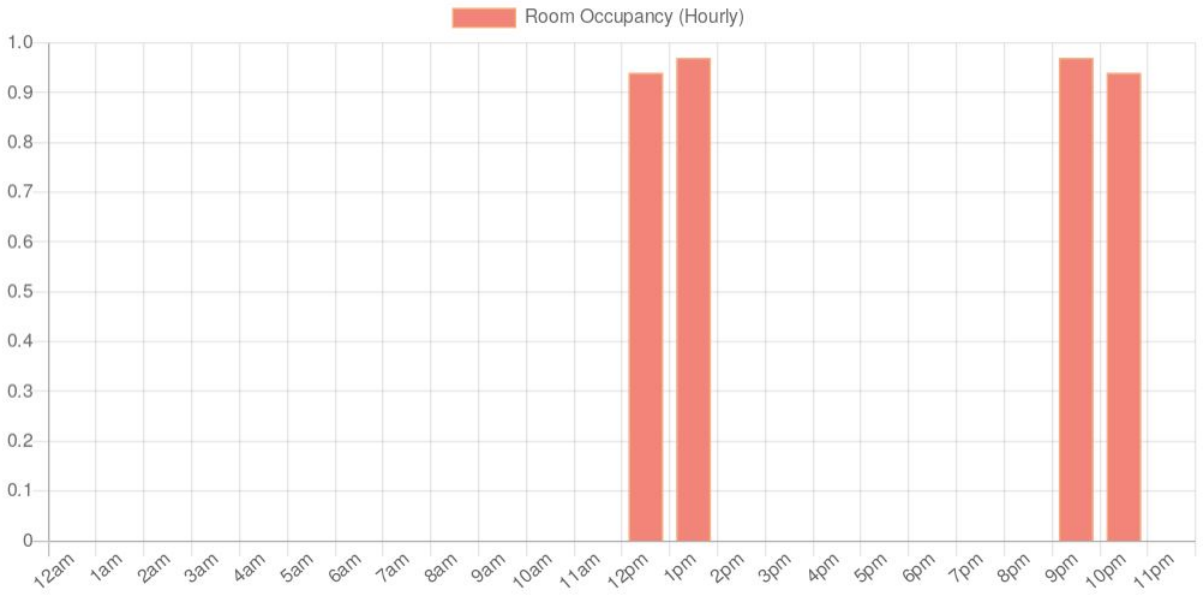
An Internet-of-Things application which performs dynamic visualisation of room occupancy.

JSON API | Github Repo

Room Id	Time	isOccupied
1	Mon Nov 11 2019 12:44:32 GMT+0000 (Coordinated Universal Time)	true
1	Mon Nov 11 2019 12:45:32 GMT+0000 (Coordinated Universal Time)	true

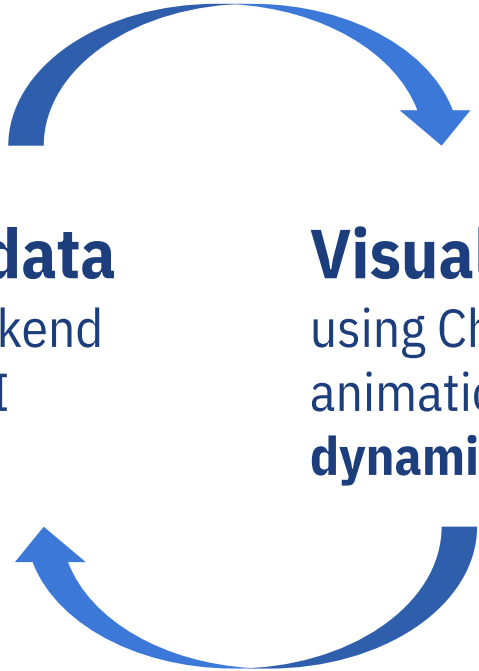
- Deployment: Nginx, Docker-compose

Frontend React Web Server



- Stack: ReactJS, ChartJS

Frontend React Web Server



Fetch data
from backend REST API

Visualise data
using ChartJS (supports animations, transitions, **dynamic data**)

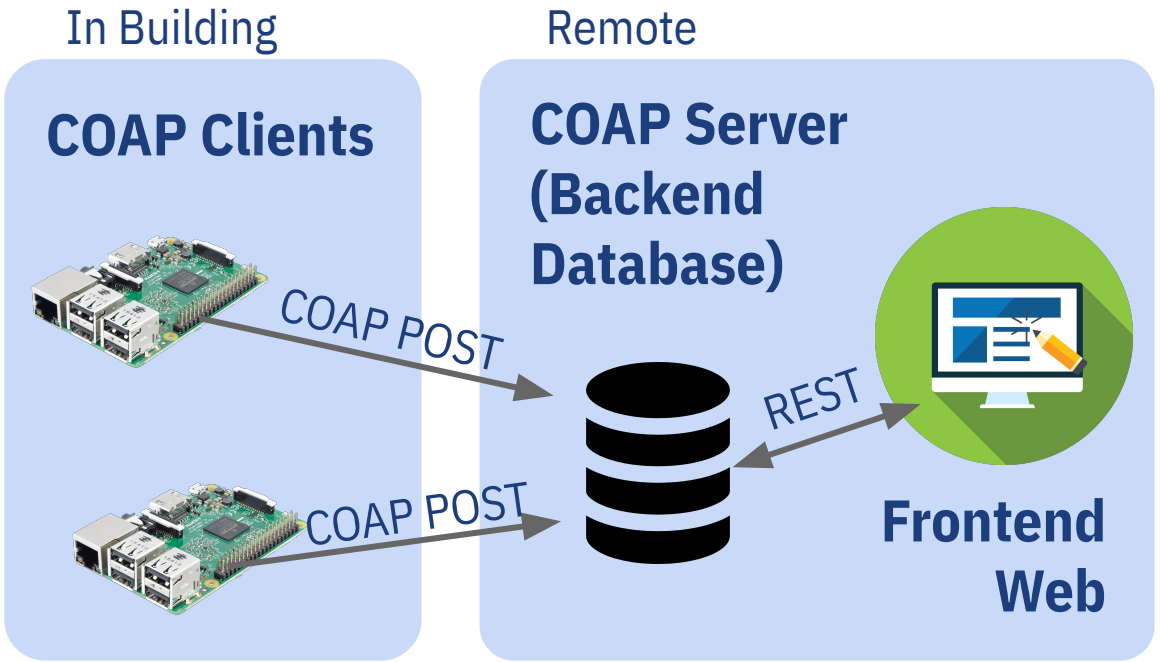
Value-Add:

Data Visualisation

- **Real-time hourly data**
  - Residents can **see if room is currently occupied**
- **Aggregated data by day**
  - Residents can **plan events in advance** (choosing days with fewer people studying)

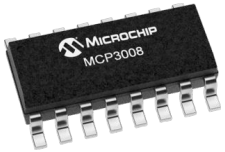
- **Calibration to SG time** using moment-timezone library

Review


In Building
Remote
COAP Clients
COAP Server
(Backend
Database)
COAP POST
COAP POST
REST
Frontend
Web

Electrical Circuit Design

- Using Analog-Digital Convertor/ Capacitor
  - Failed to debug the circuit (even after 2 weeks!)
  - Possibly due to failure of components?



- No additional resistors as the LDR is a resistor itself
  - LED in circuit was pretty dim even in a lighted room, indicating sufficient resistance
  - Hardware will not be damaged

Connection Problem with NUS Wifi

- RPI using **NUSWifi**:
  - Web server **could not receive messages**
  - Tried various ports, none worked
  - Works using other networks or 4G hotspot

- *NUS permissions/ firewall policies?*

Using COAP vs HTTP

- We are sending COAP messages over the internet, from the RPi to the Web Server
  - HTTP / HTTPS is an option
- COAP is built for **resource-constrained** environments
  - Less overhead and achieves the same purpose (no need for security features etc)

THANK YOU!

GitHub Repository:
https://github.com/Happytreat/dynalite