

# Team GR

- Team members
  - Gabriel Rodriguez-Padilla (rodrigg2@mail.uc.edu)
- Project Advisor
  - Brian Bailey (baileybp4@gmail.com)
- Project Area
  - IoT Analytics

# Project Abstract

- The focus of this IoT Analytics project will be building a data pipeline template that is capable of collecting atmospheric data from wireless embedded devices. Hopefully, this project will prove that such a pipeline can be built cost effectively, quickly, and with extensibility in mind. This project will also test new technology “in the field” to determine if the benefits offered by the new technology outweigh any downsides. Note that new technology will mainly be evaluated based on how efficiently it performs, how beginner friendly it is, how safe it is to use, and what software license it uses (with permissive licenses being preferred).

# Project Purpose

- This project is to demonstrate that creating a data pipeline in-house can be done:
  - Quickly
  - Affordably
  - With extensibility in mind

# Project Goals

- This goal of this project is to create a data pipeline template that:
  - Meets the criteria specified by the project purpose
  - Is affordable to implement
    - This requires extensive documentation covering the various pipeline components and their alternatives.
- A secondary goal of this project is testing newer technology “in the field”. Examples of such tech currently include HTMX, the Blacksheep Web Framework, and the Wokwi VS Code plugin

# User Stories

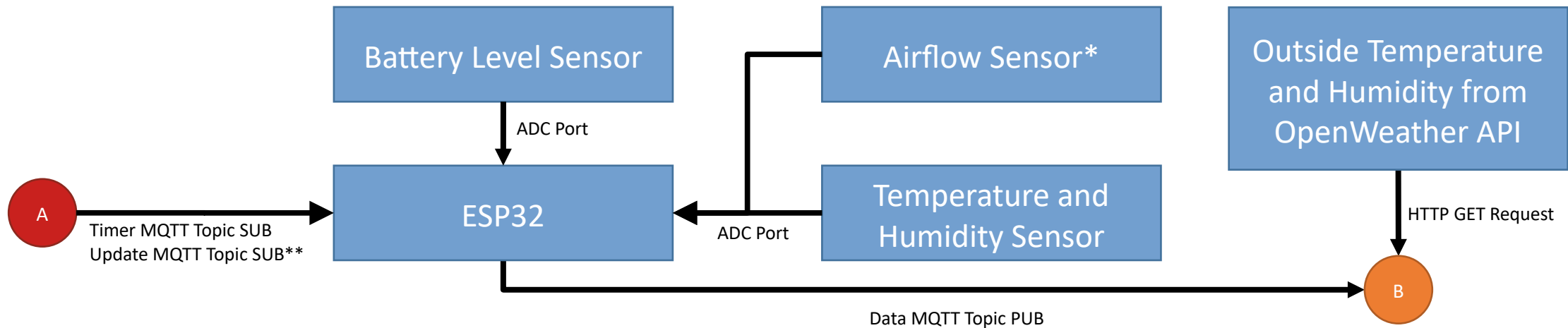
- As a system administrator, I want a system that is easy to set up and maintain, so that I can more easily stay up to date with other tasks.
- As a data analyst, I want a data pipeline that allows me to quickly access and view data, so that I can more easily make decisions based on the data collected.
- As a systems engineer, I want a data pipeline that is easy to extend, so that I can add and remove functionality as system requirements change.
- As a project manager, I want a data pipeline that is cheap to operate, so that my team can have a larger portion of funding available.

# Economic Constraints

- Since this project is being personally funded, the budget for acquiring equipment and material for this project is rather limited.
- The current estimate for the project's budget is no more than \$150.
  - The use of hobbyist electronics and free software will help reduce costs
  - Any purchases will also be done during the holiday season, where holiday sales will further help reduce costs.
  - Luckily, there are plenty of options available for both hardware and software that are both affordable and extensible.
- Since this is a personal project, no economic development or benefit will result from this project.

# Design Diagram (1/2)

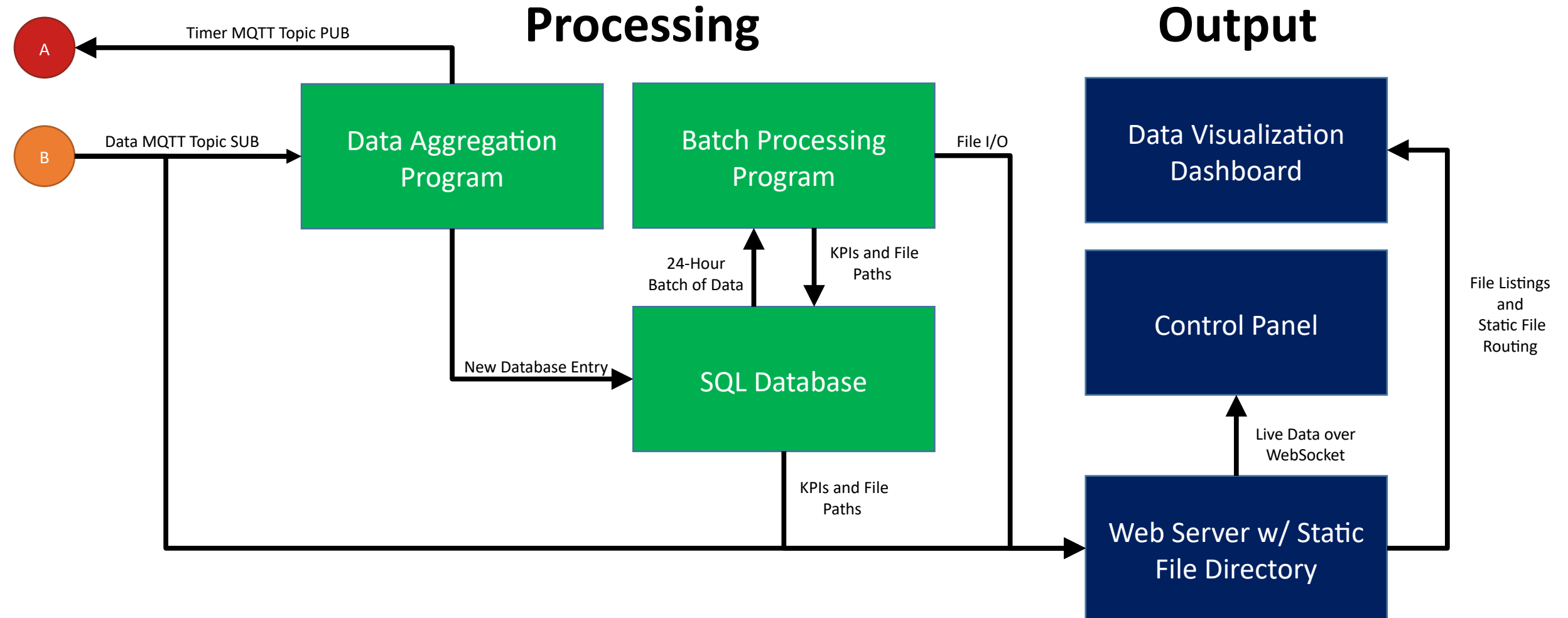
## Input



\*To be implemented later

\*\*The Update MQTT Topic publisher is a separate program to be ran by the admin. Therefore, it is not shown in the pipeline's design diagram

# Design Diagram (2/2)





# Security Constraints

- Data being sent through the MQTT topics must be authenticated and encrypted.
  - Authentication will prevent unauthorized actors from publishing data to the topics. This helps ensure that the data pipeline operates as intended.
  - Encryption will ensure that the data being sent will remain confidential. This is especially crucial when sending source code to the clients for an over-the-air (OTA) update.
- Source code being sent for an OTA update via the Update MQTT Topic must utilize JSON Web Token verification. That way, each client can verify that the code was sent by the system administrator before proceeding.
- Having a sign-in page for the web pages being served would prevent unauthorized users from viewing the collected data. However, this is a lower priority vulnerability since the web pages can only view data.

# Current Project State (1/2)

- Hardware
  - The project's embedded devices will use the ESP32 series of microcontrollers:
    - Low price and power consumption
    - Wide variety of peripherals (such as ADC ports)
    - Wireless capability
    - Software development can be done in Arduino (C++), MicroPython, and Rust
  - Next step is to select which sensors and battery pack will be used

# Current Project State (2/2)

- Software
  - Software development was being done in Rust. However:
    - The type system is very strict
    - The borrow checker can be difficult for novices to understand
    - Third-party package resources are lacking when compared to Python
    - Though Actix is among the fastest web frameworks, it can be difficult to extend/customize its functionality
  - Software development is now being done in Python:
    - Due to Python's simplicity and extensive third-party package support, more work can be done within an equivalent amount of time
    - MicroPython, in conjunction with the Micropy CLI, makes embedded software development relatively simple
    - The Blacksheep Web Framework has proven easier to extend and organize than Actix

# End-of-Term Deliverables

- Fully functioning MQTT network and control panel. This leaves the following for next term:
  - Code migration from the Wokwi Sim to hardware
  - Data aggregation and processing implementation
  - Data pipeline template documentation/tutorial

# Proposed Expo Demo

- Fully functioning data pipeline. At the demo, only temperature and humidity will be recorded since air flow will be difficult to record/set up. This implies that incorporating airflow measurements into the pipeline will be a last priority.