TLD Worker Bee

Sponsored By: Harlan Mitchell, Honeywell

Mentor: Austin Sanders

Austen Christensen, Morgan Lovato, Wei Song

College of Engineering, Informatics, and Applied Sciences
Northern Arizona University, Flagstaff, AZ, 86011

Problem and Motivation

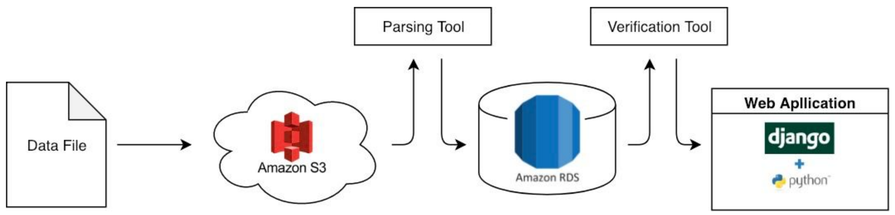
Air travel is the fastest way to travel large distances. For instance, a traveler can get from Phoenix to Los Angeles in a little over an hour by plane when it takes 8+ hours to drive. Every day, there are over one hundred-thousand flights scheduled across the globe with up to one million people in the air at any given point in time. This makes it vital for every plane's engines to constantly be working properly since there is nothing stopping a plane from falling out of the sky other than it's own momentum.

The current process for diagnosing any faults with the aircraft engine is to view the TLD data stored on the Engine Control Unit. The problems with the current solution is as follows:

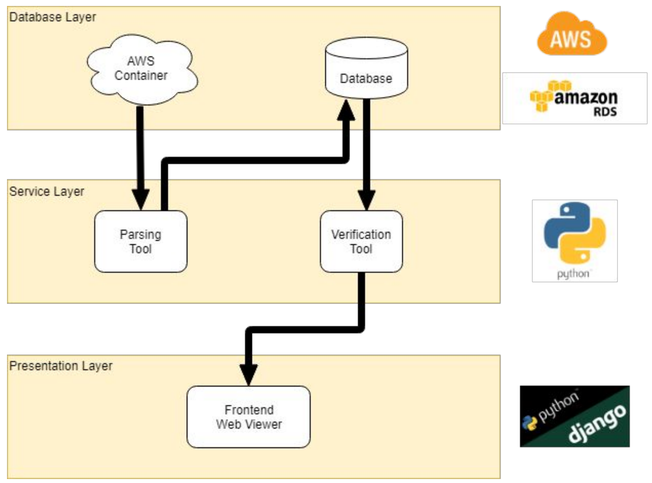
- The program runs on an outdated early 2000s Microsoft OS with no plans on updating.
- The technician has to physically stand next to the aircraft in order to download the data.
- The costs of this process for the owners of the aircraft.

Solution and Architecture

Our solution is a prototype web application that will serve as a data viewing tool for data that is stored in the cloud. The flow of data is as follows:



We chose to base our product on the Model, View, Presenter architectural style.. We were able to break our project into three parts: Database Layer(Model), Service Layer(Presenter), and the Presentation Layer(View).



Our Product

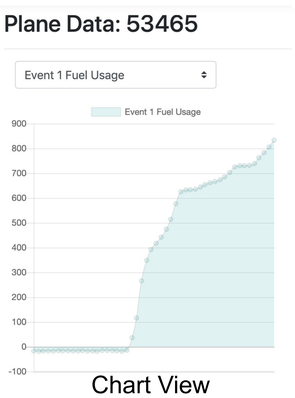


Chart View

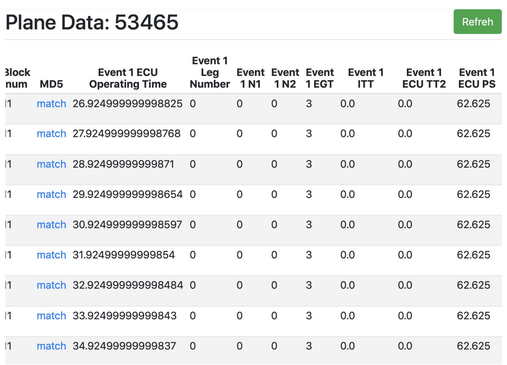
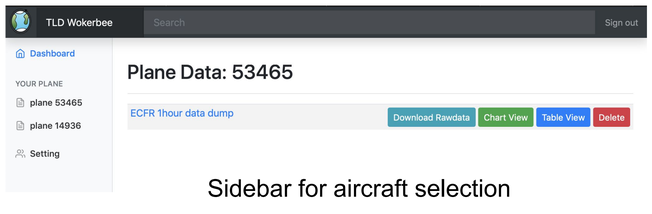


Table View



Sidebar for aircraft selection

MD5 Verification

```
cloud
MD5:705261389264197fd03d38be7c971e60,
local
MD5:705261389264197fd03d38be7c971e60
```

Outcomes

Our solution has improved our client's current process by remotely connecting to the data making the process of viewing this data fast, easy, and reliable.

Challenges

- Cloud and database failure
  - Backup mechanism like Redundant Array of Independent Disks (RAID)
- Network Failure
  - Revert to the current EEI solution as a backup plan
- Network security
  - Add credential process. Apply techniques to avoid SQL injections

Key Features

- Cloud databases are used to eliminate the use of physical data transfer.
- Parsing and verification tools are used to ensure data integrity throughout the process of data transfer, This ensures no incorrect data can be displayed.
- The web application is used as a data viewing tool that pulls data from the databases in the cloud. This eliminates any physical connections needed,

Testing

- Unit Testing
  - Pytest as testing framework
  - Conducted 8 major units, each unit will have equivalence partitions, values, erroneous values and selected inputs for partitions
- Integration Testing
  - Use Pytest with Travis CI to automate our tests
  - Test the way the different layers of our solution interact together to ensure a well working system
- Usability Testing
  - Test if users can navigate web application and perform basic operations
  - Ensure our product is user friendly and does not require any formal training

pytest

Future Work

Our project will be passed to the Honeywell development team where they will incorporate our backend with their front end and authentication system. Once the Honeywell development team is ready, the users will be able to use our product to save them money while increasing the safety of their passengers.