



Up In the Air:

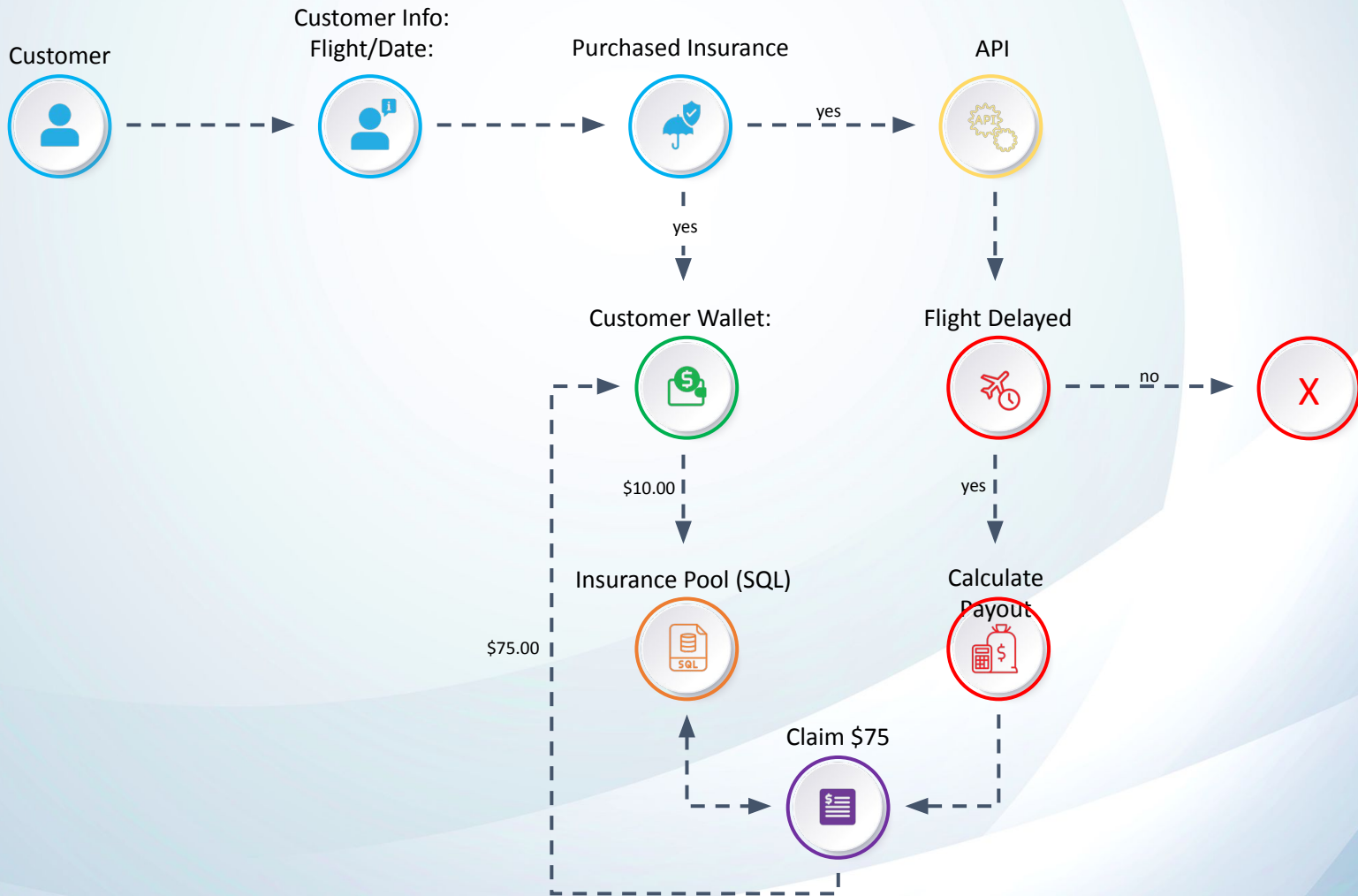
Utilizing Flight Data API's to Automate Insurance Payouts

Team Members: Sean Simmons, Siegfried Vieluf, Santiago Pedemonte, Rawnak Mahjabib, Igor Sogursky

Motivation & Summary

Our objectives:

- Utilize flight API's to source delay data
- Price and payout a delay/cancelation claim based on API delay info in an automated way



API's

- **Shopping for the right API:** There are a ton of flight API's out there to choose from. Not all have the same functionality as we've all experienced with other types of API's. A good starting point was:
<https://geekflare.com/flight-data-api/>
 - we had to go through several iterations to eventually choose two API's with different features
- **We eventually chose two - Aviation Stack and AeroDataBox:**
 - <https://aviationstack.com/> / <https://www.aerodatabox.com/>
 - Both had applicable - and importantly free - functionality for the purposes of this project

Spoiled by Choice - Many ways to solve the same problem



- **We had discussed both a SQL-based and a blockchain-based solutions.**
- **SQL Pros:**
 - Reliable storage.
 - Easily accessible.
 - Centralized.
- **Blockchain:**
 - Using IPFS, records publicly available.
 - Lends itself to Smart Contract implementation.
 - Decentralized

Back End (*Santiago)

- **Buying Insurance:**

- Gather user information: Name, flight number, flight date, ticket number (?), public address, private key.
- Uses `send_tx()` function for payment of policy.
- Generates the record of the passenger:
 - In the SQL implementation, the passenger gets added to our database and given a customer ID and a Transaction ID.
 - In the Blockchain implementation, a .json file is generated containing user information and gets a claim ID. This is what would be published in the IPFS.

- **Claiming Insurance:**

- Ask for the user's unique ID generated at the time of the transaction, and make another API call to confirm flight delay.

DEMO (*Rawnak time to shine)

<https://share.streamlit.io/rawnakmahjabib/up-in-the-air/main>

Pricing Insurance is HARD

- **Flight insurance has many pricing challenges:** Our research and attempts to create a math and data science based approach to pricing flight insurance proved to be too high a mountain for a two week project
 - It turns out that pricing flight insurance is an incredibly complex task, that many data scientists and companies are pouring resources into creating better models to determine the probability of delays
 - Complex models would be necessary - alongside very expansive and expensive data sets - in order to really full explore this aspect of the project

Reasons For Delays:

- **Departing and Arriving Airports**
- **Time of Year**
- **Time of Day**
- **Airline Chosen**
- **Day of the Week**
- **Duration of the flight**
- **Traffic Delays**

Predicting Future Delays

- **Figuring out a future delay is difficult:**

Brainstorming the idea to use a random forest algorithm to predict future flight delays to give insight to the consumer to purchase the insurance had several layers that we couldn't work out in two weeks.

- How do we interpret a delay? Using departure or arrival?
- What caused the delay? A statistical correlation does not signify causation.
- How can we account for Operational risk?

Manipulating the Data to attempt predicting delays:

There were things outside of our controls but a framework that could have worked would have been the following.

- Create models per region at either the departing location or arrival location
- Change the historical data of the weather API of rain/fog/snow to 1 and the rest 0.
- Change airline's as to numerical code
- Run the model knowing we would be missing other reasons of delay