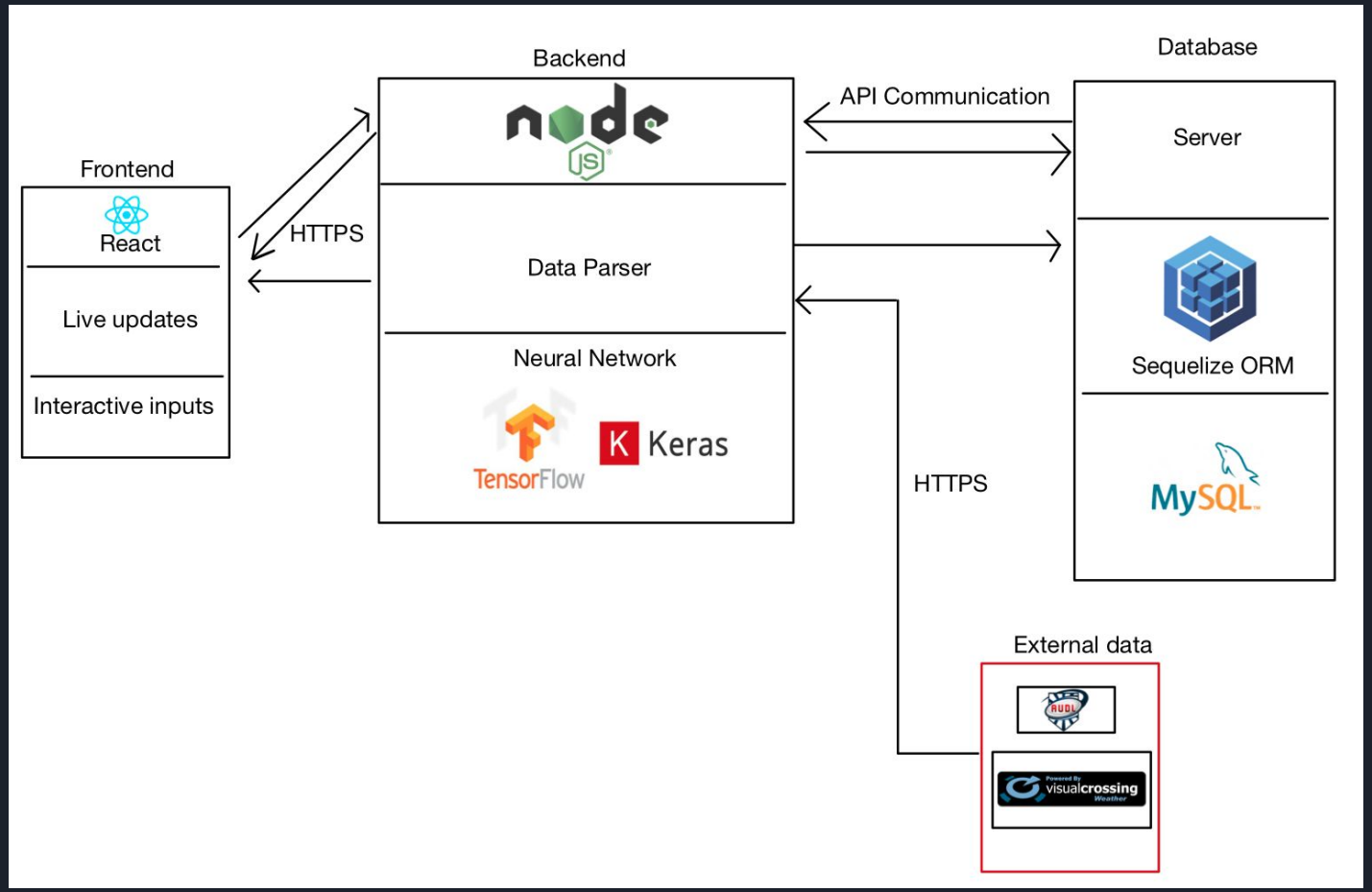
U.P.E (Ultimate Prediction Engine) - Technical Design

Brandon, Karim, Philip, Samuel

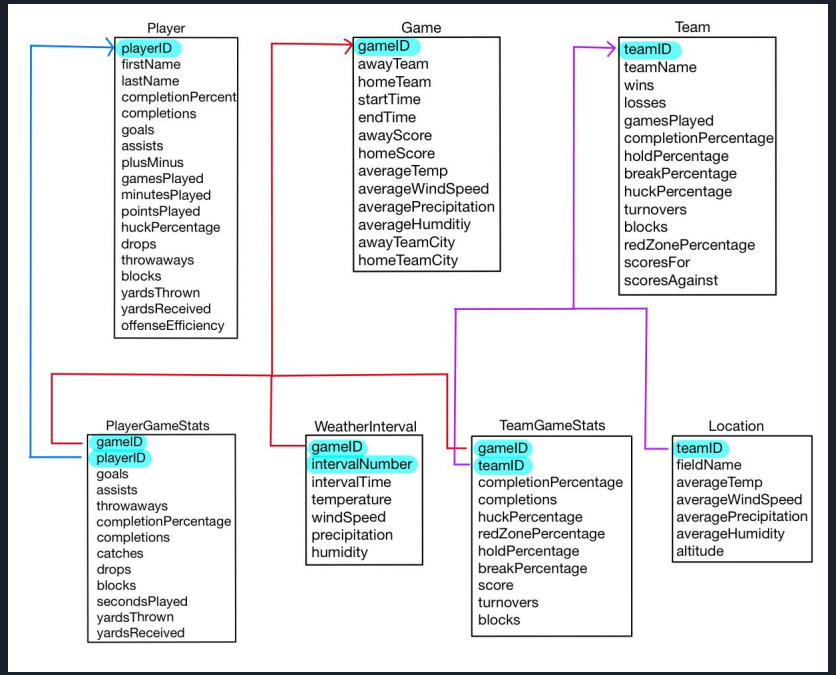
Overview and System Goals

- Ultimate Frisbee Prediction App for the AUDL (American Ultimate Disc League)
- Web Application built around a prediction engine
  - Weather conditions during past games + team and player records from previous seasons
- Provide Ultimate Frisbee Fans with accurate game predictions for the upcoming season
- Live Updates for Game Predictions as they occur


Backend
Database
API Communication
Server
Frontend
React
HTTPS
Data Parser
Live updates
Neural Network
Interactive inputs
TensorFlow
Keras
Sequelize ORM
MySQL
HTTPS
External data
visualcrossing

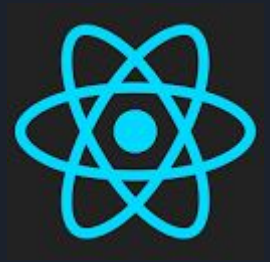
Data Handling

- Database design
- Data scraping
  - AUDL
  - Visual Crossing API requests
- Data cleaning and verification
- Constant updates to data stores



Technologies Used

- Node.js Backend
- React Frontend
- AWS EC2 Cloud platform
- MySQL data server
  - Sequelize ORM
- Python (TensorFlow Library and Keras API) for ML algorithm



AUDL Integration to Neural Network

Completed

- Player Statistics
  - Completions
  - Hucks
  - Drops
  - Points Scored
- Team Statistics
  - Wins
  - Losses
  - Score

Future

- Comparisons on ideal data
- Compare in conjunction with weather data
- Integrating into model
- Determining best statistics for model

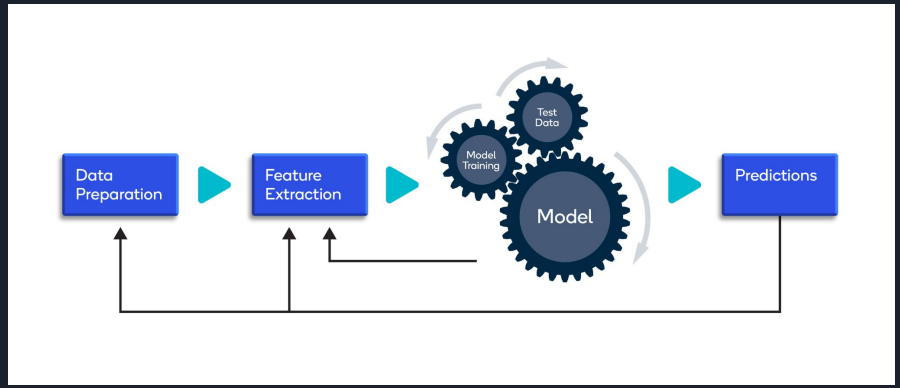

```
{"playerID":"labramows", "firstName":"Lou", "lastName":"Abramowski",
"completionPercentage":null, "Goals":0, "assists":1, "plusMinus":-3, "gamesPlayed":7,
"minutesPlayed":39, "pointsPlayed":51, "huckPercentage":null, "drops":0, "Throwaways":5,
"blocks":1}
```

Weather Integration to Neural Network

- Weather Data from Visual Crossing Weather API
  - Multiple 15 minute intervals during past games
  - Average weather conditions for each game
  - Temperature
  - Precipitation
  - Windspeed
  - Humidity
- Average weather conditions for each team's home field
  - Information on types of weather conditions that each team has experience with
- Integration with Player and Team Data

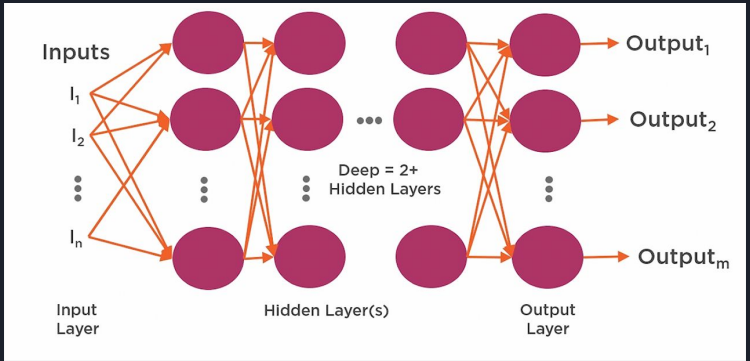
Structure of ML Algorithm (Data)

- Sequential model using Keras API that trains data from our database to predict game outcomes
  - Train data consists of both individual game statistics as well as general season statistics (As previously mentioned in DB design)
  - These data inputs will be weighted automatically by the model to determine the best possible fit/distribution
  - Non-numerical data (Such as Home/Away team name in Game data) converted into numerical data using OneHotEncoder (Sklearn)
  - Dependent variable(Win/Loss)



Structure of ML Algorithm (Layers)

- ANN Layers will use ReLU as an activation function (Very adaptable)
- Output layer will use sigmoid function (For probabilistic output)
- Training loop to fit the model and measure accuracy



Future Components

- Two models in development
  - AUDL Teams and Weather - Karim
  - AUDL Players and Weather - Brandon
- Compare both for accuracy and latency
  - Split historical data in a train-test split and compare to output

- Front end design - Samuel
  - Setup of input and prediction output
- Backend design - Philip
  - API development
  - Data distribution

Alpha Prototype

- Database and server set up and available
- Basic predictions using a subset of the data
  - A few factors from players
  - One or two weather points
  - A few factors from teams
  - Player and team calculations separate initially
- Basic GUI and output
  - Small input form for a few factors
  - Prediction - little to no breakdown… yet
- Only 2 layers when implementing NN (Adding more layers in future models)
- Largely focusing non-client side

Technical Challenges and Risks

- Model prediction accuracy (Counter by performing a large number of training generations but might be costly)
- Incomplete field location data (Manual research)
- New vs veteran players and teams
- Frequent change in players on teams
- Unknowns
  - Coaches on each team
  - Incomplete player data

Questions?