ReadySet Go: A Web Application for Go AI Research and Play

Authors: Christian Arciniega, Ryan Clements, Derek Isensee, Ryan Moncrief, Kien Nguyen

Advisors: Dr. Krishna Kadiyala, Dr. Bingyang Wei

TCU COLLEGE OF SCIENCE & ENGINEERING — DEPARTMENT OF COMPUTER SCIENCE
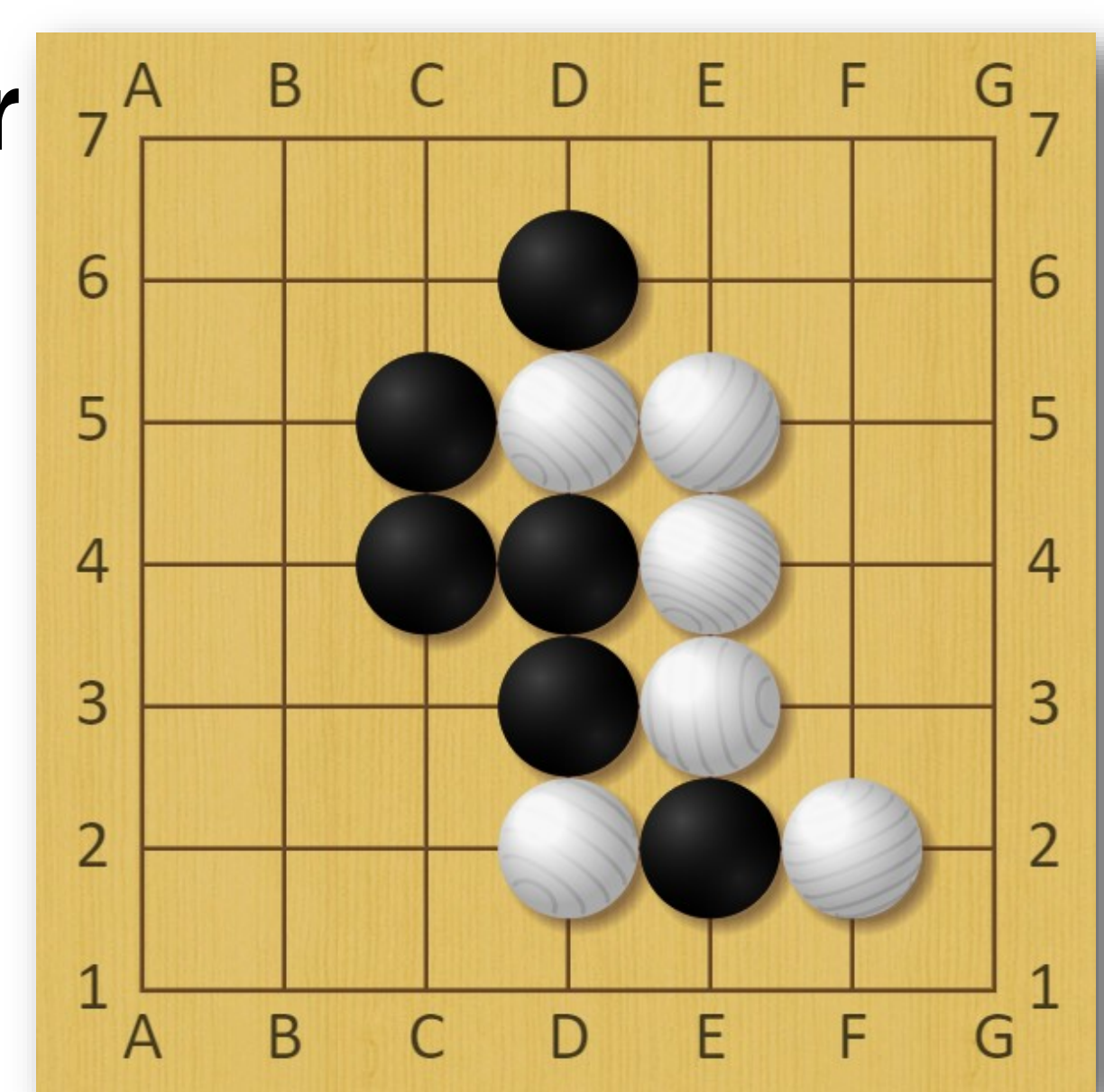
Background

TCU has launched an interdisciplinary research project into Go AI, headed by Dr. Ze-Li Dou and Dr. Liran Ma, inspired by DeepMind's AlphaGo project. However, the existing project's research interface is not very user friendly and involves long terminal commands. Additionally the project's accessibility was limited to those working directly on campus.
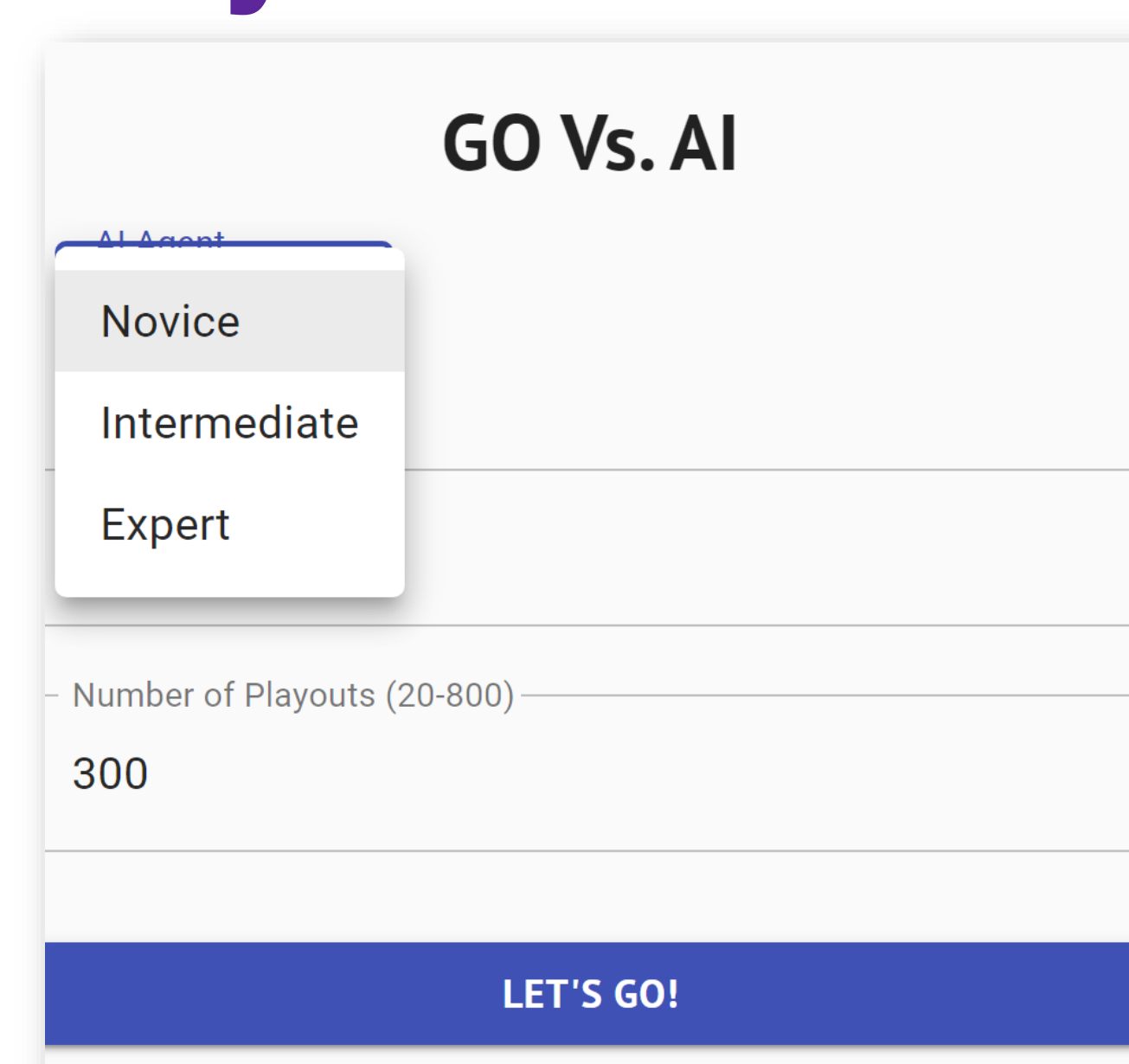

Goals

- Develop a user friendly interface for users to play Go versus AI, track their gameplay statistics, and train new AI agents.
- Expand accessibility for many simultaneous TCU users from any device.
- Allow admin user to monitor user roles and accounts
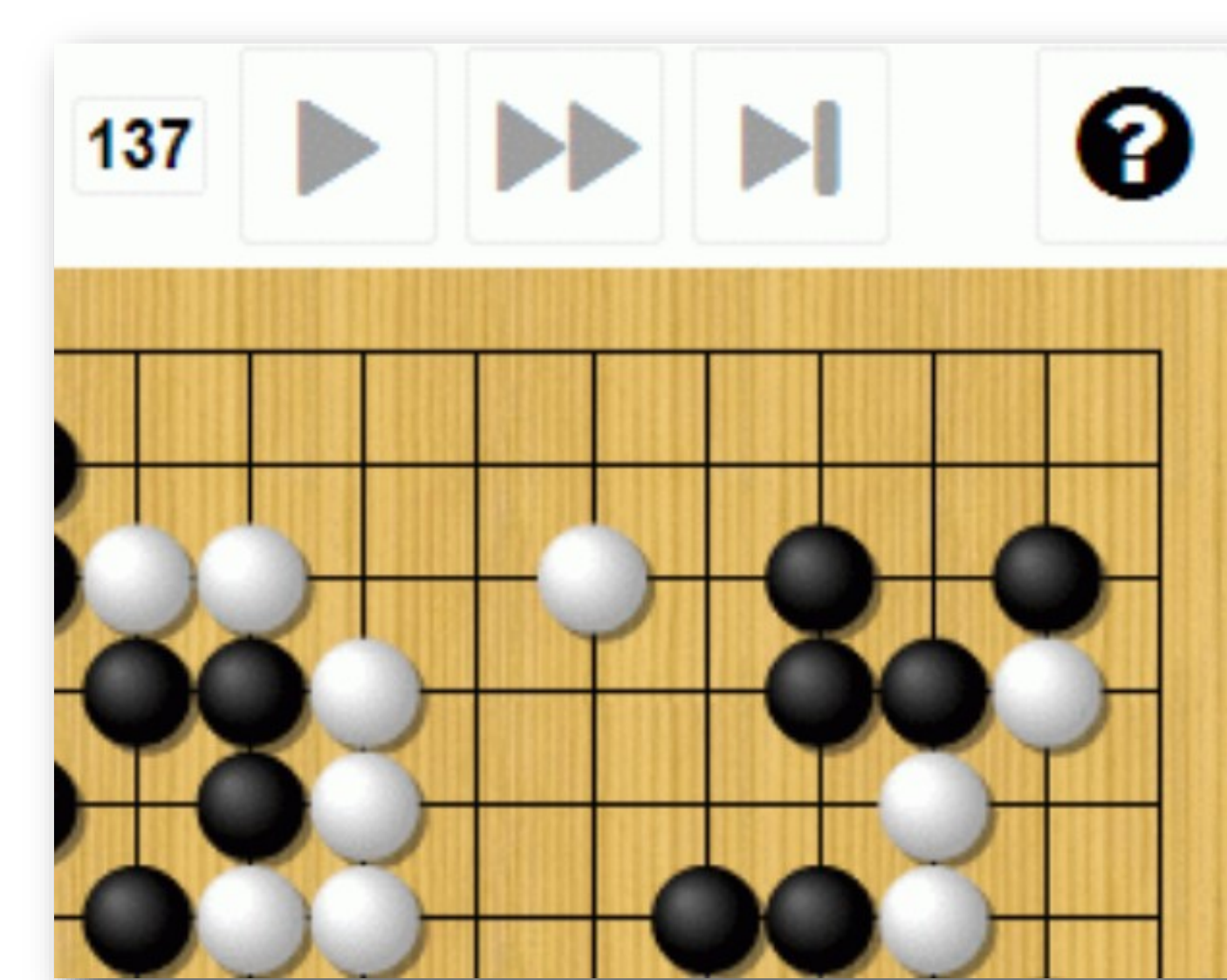- Ensure that user data is secure via https, email confirmation, and Captcha.

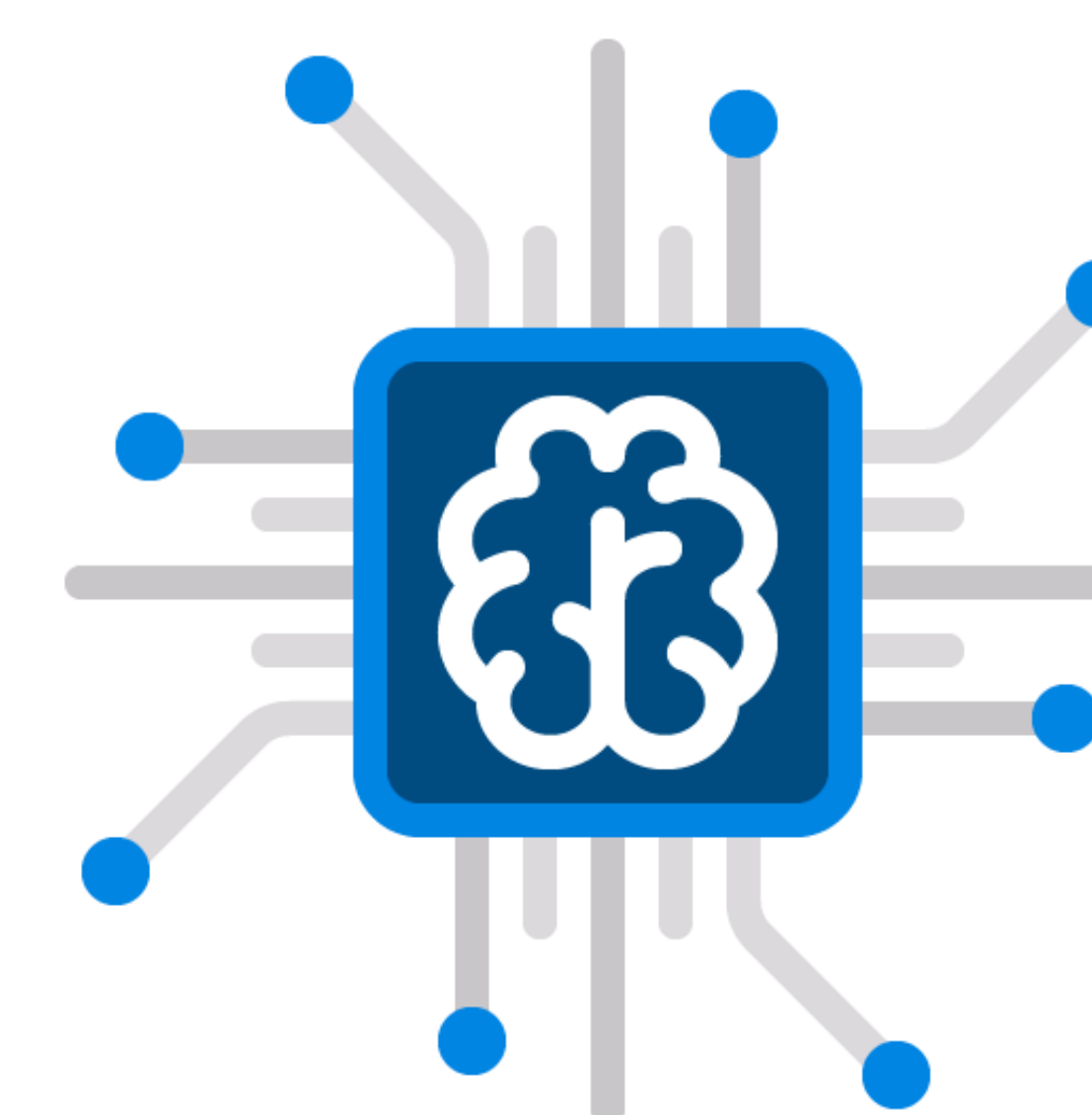### Tools Used

Site Features

Play

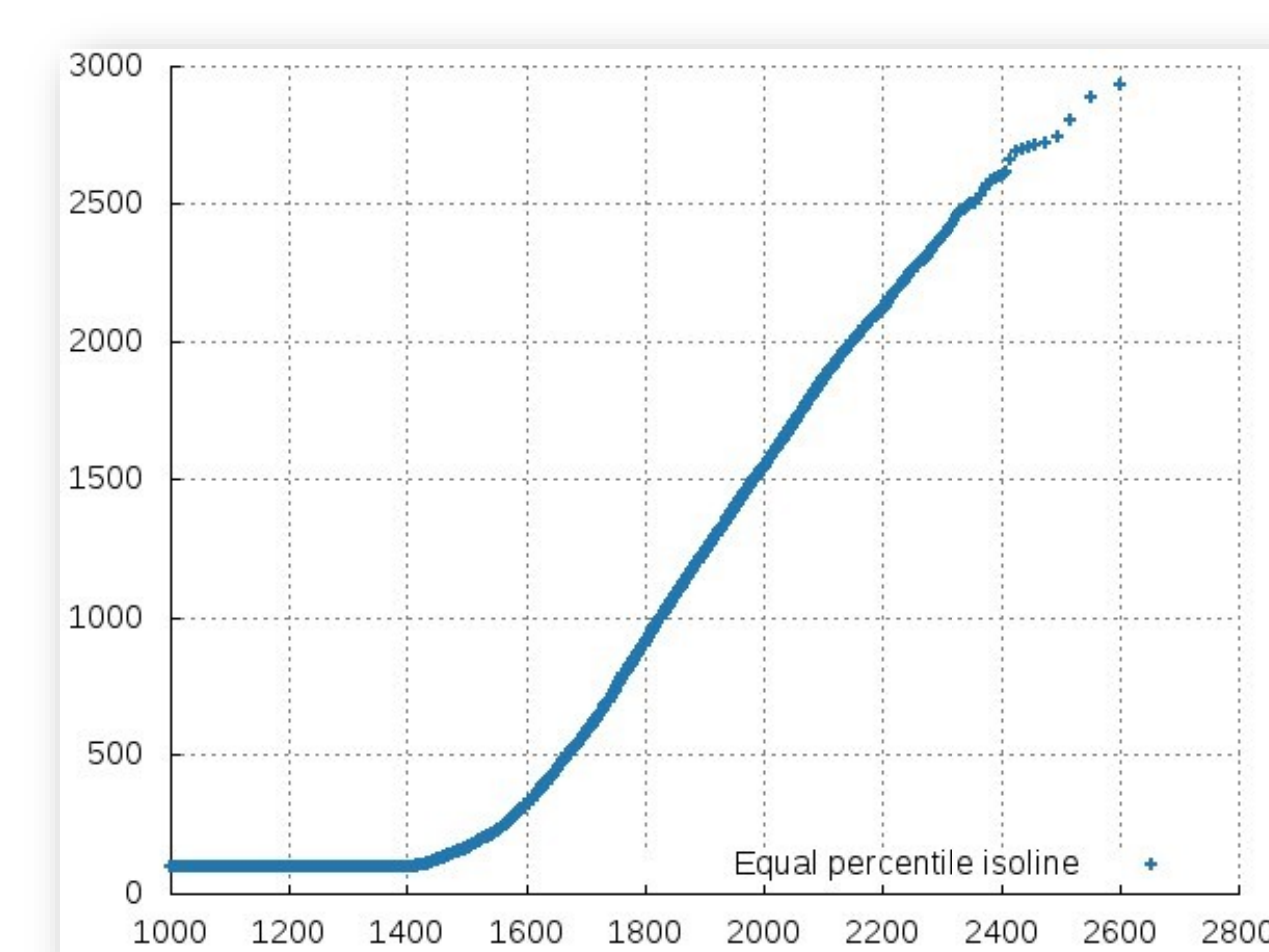Users can select parameters and play against an AI agent of their choice

Users can also request a hint for a potential move

Research

Researchers can train new AI agents using various parameters and publish them to be played by anyone.
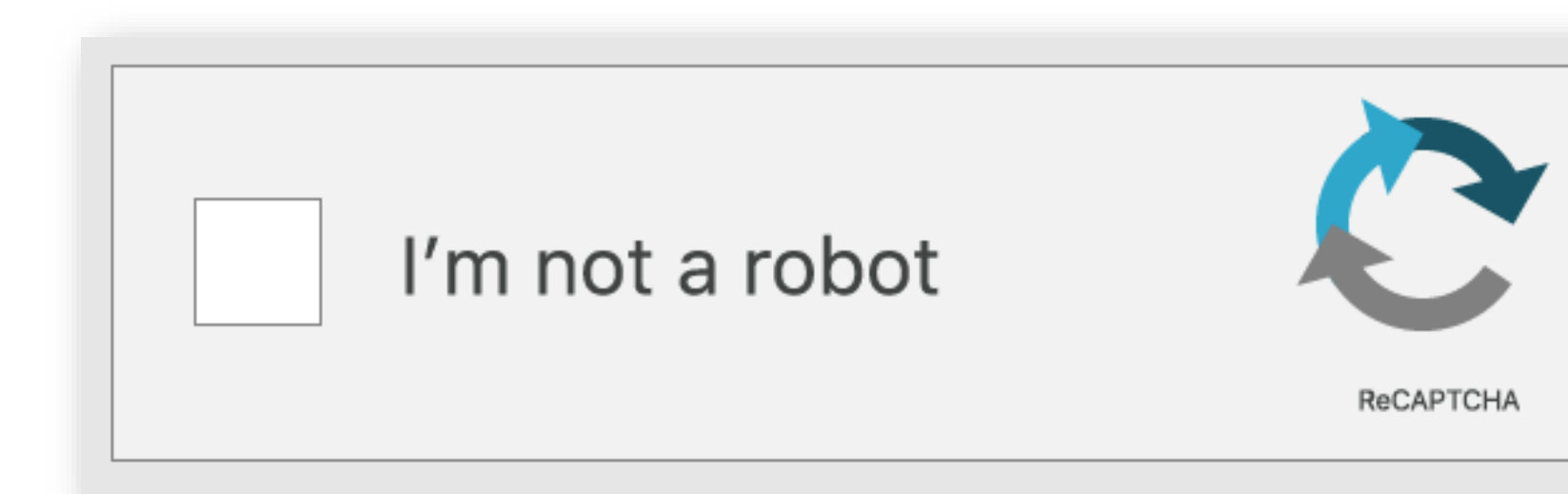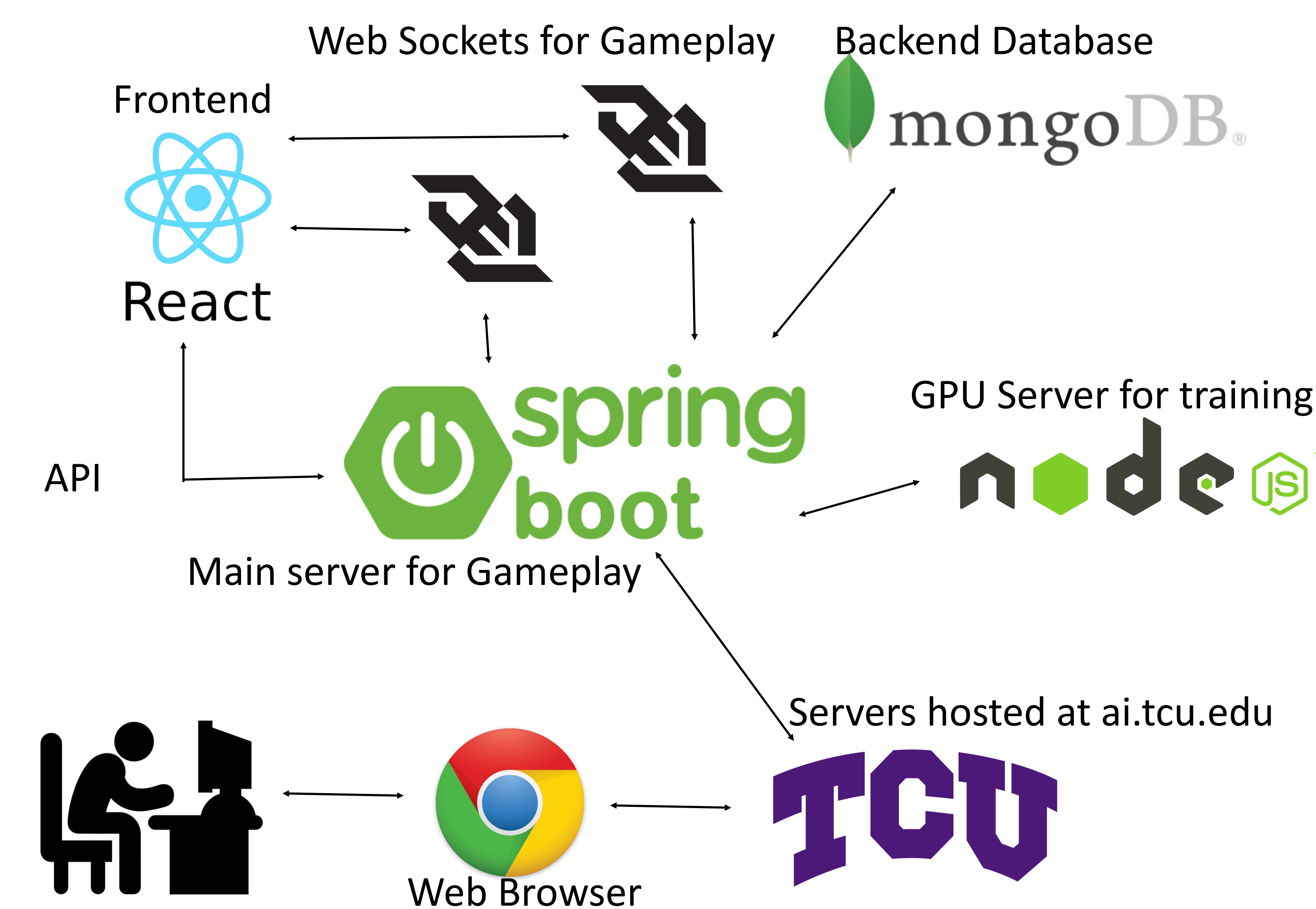

Real time ELO graphs show AI training in progress

Challenges

Because we were working with an already established complex system to implement online, our team faced many challenges, including configuring the server to allow for multiple simultaneous users, setting up and communicating between multiple servers, and formatting JavaScript libraries to be universally available across frontend and backend. Apart from these, another difficulty we had was shifting from a microservice architecture to monolithic, which was not the original design.

Security

When creating an account, users must succeed on a Captcha and verify their email.

The entire site is protected with https.

System Architecture

- Frontend: React
- Web Sockets for Gameplay
- Backend Database: mongoDB
- API
- Main server for Gameplay: spring boot
- GPU Server for training: node JS
- Servers hosted at ai.tcu.edu
- Web Browser → TCU

Resources

Material UI documentation- https://material-ui.com/

WGo JS- http://wgo.waltheri.net/

Go React Framework https://github.com/ajhyndman/go-react-redux-elm

Acknowledgements