SE 575 Project: E-voting System

Description

Fraud and corruption in the votings are very common. Hackers have always tried to destroy voting systems in any kind of elections. How to make the election system more secure has become a focus of attention. In our project, we attempt to build upon the Blockchain infrastructure to create a secure game voting system . The application of blockchain technology could eliminate voter fraud, preventing rigged election and providing clear ballot records. We will design and implement an e-voting system to realize the following goals:

1. Ballot-privacy: No outside observer can determine for whom a voter voted.

2. Individual verifiability: A voter can verify that the ballot containing his vote in the published set of "all" votes

3. Universal verifiability: Anyone can verify that the result corresponds with the published set of "all" votes

Group Members

Saffat Hasan
Fnu Alisha
Naijia Wang (online)
Zexi Yu

Features

1. **User registration:**
   To register for a voter, enter all the required information including game username, nation, first name, last name, email address and hit the Register button.

2. **User role control:**
   Use your registered game username for login

3. **Vote for a Candidate:**
   Pick your favorite player and enter the correct voterID to cast the vote. It counts only the first vote as valid from each address and for each valid transaction, add 1 to the candidate.

4. **Avoid double voting:**
   Verify the vote is counted correctly in real time for any voter and after casting the vote, the voter is not allowed to log in with the username anymore.

5. **Verify the vote:**
   The user can check the public ID and voting result by entering the game username.

6. **Tallying votes:**
   In the home page, hit the get poll standings and a pie chart of the current voting results will be shown.

Technologies Used

Node.js
Vue.js
HTML
JavaScript
Hyperledger Fabric SDK
IBM Blockchain Platform
IBM Kubernetes service
Deploy on AWS

Architecture Description

1. The user interacts with the web application UI (Vue.js) to login/register, cast their vote, and query the current state of poll standings.

2. Based on the kind of client interaction, the Node.js server interacts with the IBM Blockchain Platform service and generate client APIs for data exchange.

3. The IBM Blockchain Platform service creates the Hyperledger Fabric network on IBM Kubernetes Service, which is maintained by the developer to launch smart contract on network.

Architecture Diagram