# Voting System Design

# 1. System Overview

This voting system is designed for Aleo DAO to facilitate its decision-making process through voting. The system consists of smart contracts, a user interface, and a blockchain network, supporting anonymous voting and real-time result viewing.

# 2. Smart Contract Design

## 2.1. Contract Features

- Adding candidates
- Voting (including the collection of voting fees)
- Viewing voting results

## 2.2. Contract Structure

```
program VotingContract.aleo {

    struct Candidate {
        name: string,
        votes: u64
    }

    candidates: map<string, Candidate>;
    mapping hasVoted: address => bool;

    public func addCandidates(candidateNames: array<string>) {
    }

    public func vote(candidateName: string) {
    }

    public func getVotes(candidateName: string) -> u64 {
    }
}
```

# 3. Frontend User Interface

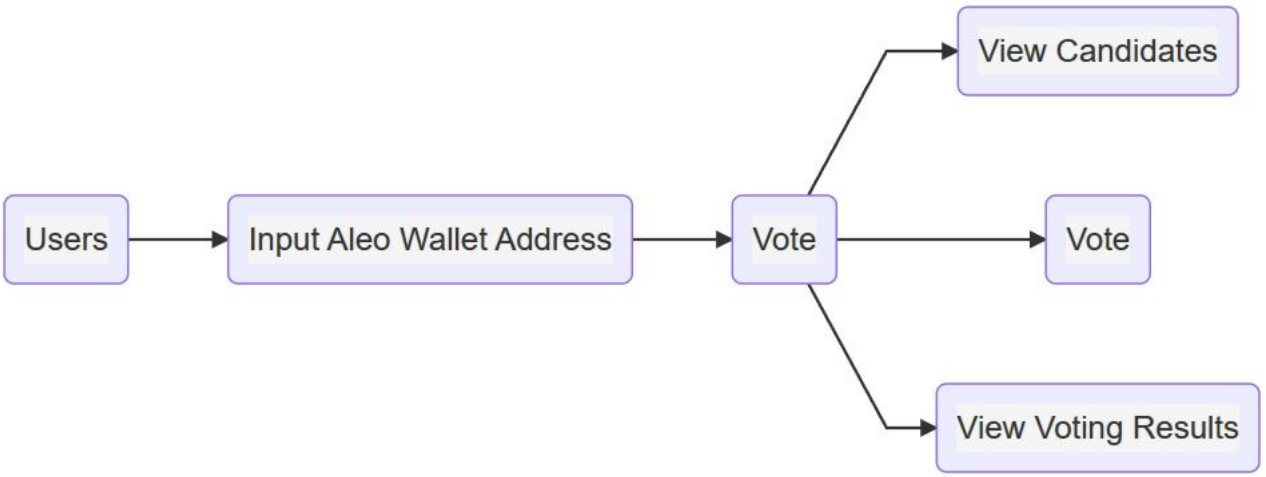
## 3.1. Functional Requirements

- Input Aleo Wallet Address: Bind the Aleo Wallet Address.
- View candidates: Display all candidates.
- Vote: Users select a candidate to vote for, and voting confirmation information is displayed.
- View results: Display voting results in realtime.

## 3.2. Technology Stack

- Frontend framework: React.js or Vue.js
- Interaction with smart contracts: Aleo SDK
- UI design: Bootstrap

## 3.3. System Use Cases




# 4. System Architecture

## 4.1. Components

- User Interface (UI)
- Smart Contract
- Blockchain Network

## 4.2. Data Flow

Users interact with smart contracts through the UI, the UI passes user operations to the smart contract, the smart contract processes requests and updates the blockchain status, and the UI retrieves the latest data from the blockchain and displays it to the user.

# 5. Security and Privacy

- Anonymous voting: Encryption technology is used to protect the privacy of votes.
- Preventing double voting: Smart contract logic ensures a single vote.

# 6. Testing and Deployment

- Testing network: Functional testing is conducted on a test network.

- Deployment: Deploy the smart contract that has passed testing to the main network.
- Monitoring: Monitor the system's operation status and respond to issues promptly.

---