Lovely Blockchain

Tsemko Andrii

Introduction

- LovelyBlockchain provides a blockchain system for the voting process.
- All voting tickets and votes of users save as blocks.
- System provides the ability to create a vote for a specified list of users and provides the ability for votes by using voting_tickets.

Advantages:

- Anybody can create their own voting account and create their voting_tickets for a specified list of users.
- Voting tickets can be spent only by a user, who gets this voting ticket.
- Voting tickets can be spent only for a voting account, who create them.
- Any user can calculate the resulted score of each opponent for each vote.

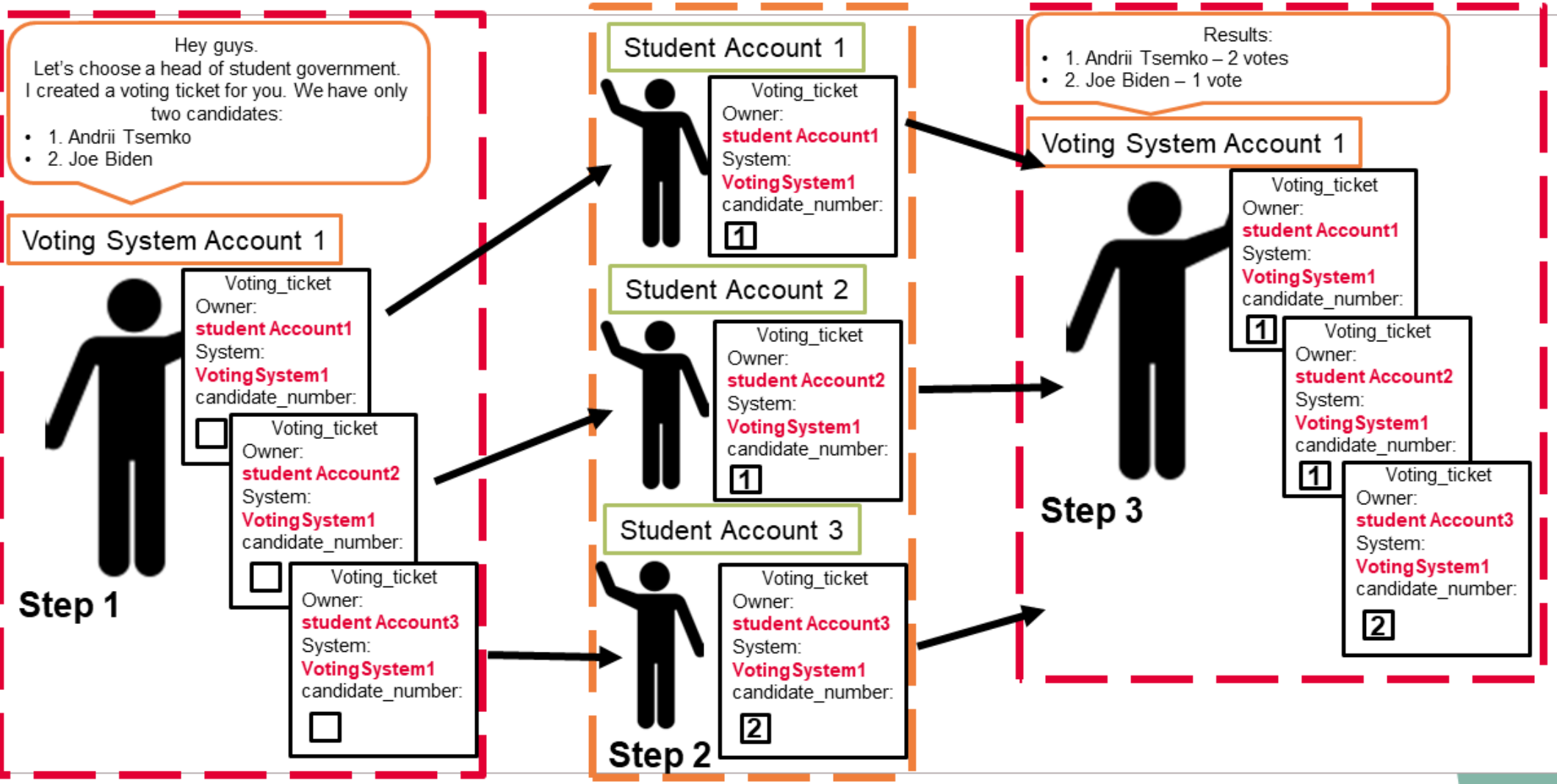
Disadvantages:

- Any blockchain contains an open list of transactions. It is possible, that knowing the owner of a specific address will show all his votes.
- The list of addresses, who will get a voting ticket should be verified manually for checking all unique persons.

Voting Process Step-by-Step

1. Students create their students accounts.
2. User, who want to create a votes should create a VotingSystemAccount.
3. User inform students about candidates in a votes.
4. User, who want to create a votes should MANUALLY prepare a list of students accounts, who should take a part in a voting process.
5. VotingSystemAccount creates a voting tickets and sent them to list of students accounts.
6. Students create a vote transaction, where they spent their voting tickets and sent this transaction to a VotingSystemAccount.
7. VotingSystemAccount calculate the number of vote_number for each candidates from received vote transactions.

Voting Process Step-by-Step



Student Account (Transaction):

- Spent voting_tickets for a vote in transaction.

Vote transaction

Header:
transaction_id: hash(body)

Body:
nonce : int

Operation:
sender_id: int = StudentAccount id – who spent voting ticket.
receiver_id: int = VotingSystemAccount id – who get voting ticket result.
voting_ticket: int = Voting_ticket transaction id
vote_number: int = index of person, for whom student spent his vote.
signature: int

- Correct transaction rules:
  - 1. Voting ticket should not be used yet.
    - In Transaction Database does not exists a transaction, where voting_ticket field equals to voting_ticket in new transaction.
  - 2. Voting ticket can be used only by a student account for whom it was sent previously by Voting System Account. Each student account has unique voting ticket.
    - Voting_ticket points to a transaction, that sent by Voting System Account. This transaction should have field receiver_id that equal to a student account id, who created vote transaction.
  - 3. Vote transaction can sent (spend) specified voting_ticket only for a Voting System Account, who creates it.
    - receiver_id of vote transaction should be equal to a sender_id of voting_ticket transaction.
  - 4. Unique number of nonce
  - 5. Correct signature value

- For taking a part in a voting process, Student Account should receive a voting ticket by a Voting System account.

Voting System Account (Transaction):

- Creates voting_tickets transactions.

Voting_ticket transaction

Header:
transaction_id: hash(body)

Body:
nonce : int

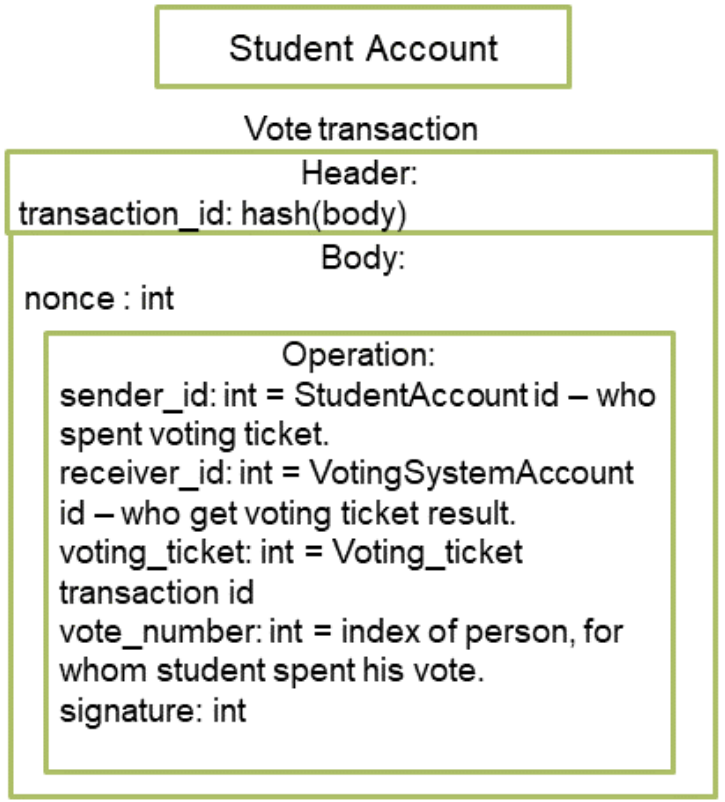
Operation:
sender_id: int = VotingSystemAccount id – who creates voting ticket.
receiver_id: int = StudentAccount id – who receive voting ticket.
voting_ticket: int = 0 (Voting System Account specification).
vote_number: int = 0 (Voting System Account specification).
signature: int

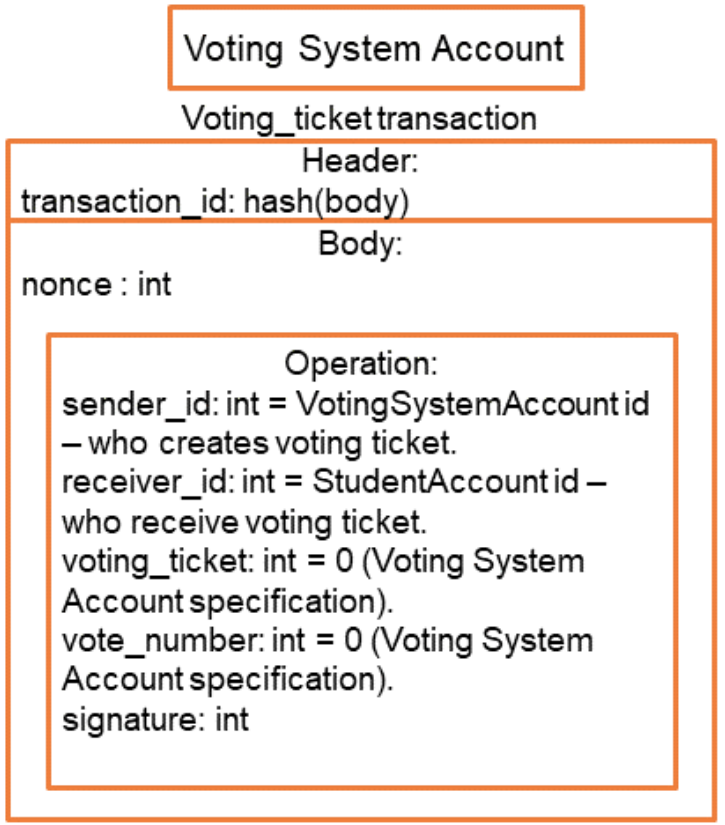
- Correct transaction rules:
  - 1. Voting ticket should not be create yet.
    - In Transaction Database does not exists a transaction, where voting_ticket created by a this Vote System Account and sent for a the same Student Account.
  - 2. Voting ticket == 0. Specify, that this is a voting_ticket type of transaction. This type of transaction creates new voting_tickets that does not need any previous transaction.
  - 3. Vote number == 0. Used for verifies the same voting_tickets by only hash value of operation of it.
  - 4. Unique number of nonce
  - 5. Correct signature value

- For creating a vote, Voting System Account should create a voting_tickets for list of known student accounts.

Accounts (Transactions):

The structure of transactions is the same for both types of transactions.





Summary

- LovelyBlockchain is a blockchain system for public votes.
- This system provides access for anyone to create votes (voting tickets and votes).
- Vote transaction should point on the user voting_ticket. This princess makes the voting process clear and does not require any central for manually verifying votes.
- The disadvantage of this system is a process of manually collecting the student's addresses list, which should have voting_tickets.