Electronic voting using Ethereum smart contracts

Lucas Bordignon, Matheus Bittencourt, Vinicius Macelai

November 23, 2018

1 Motivation

Electronic voting has been used in Brazil for over 20 years now and recent tests shows that, even with the development over all that time, the current system is not proved as a really secure system. The main issue with the solution proposed by the Brazilian Government is the lack of auditability, where you can only apply for testing the system in a controlled environment when they allow.[1]

All around the world the cost of elections are really high. For the 2010 General Elections in the UK, estimated the costs of the general election to be £113,255,271. This figure consists of £28,655,271 for the cost of distributing candidates' mailings and a further £84.6 million for the conducting of the poll. That means on average, it cost about £3.77 to register and count a single vote. If we only take the costs for conducting the poll (£84.6 million), it would cost £2.82 to cast a single vote. [3]

In centralized systems as the Brazilian one, there needs to be an enforcing body to enact the system, known as an Oracle. An independent source of truth must verify that conditions in the system occurred in order to fulfill the obligation.

Working on decentralized systems as some Ethereum smart contract, there is no need to trust anyone. Everything that is done on a smart contract can be verified by every node that is running on the network. And the data is public, meaning that everyone can check which data the smart contract is holding and audit during or after the election in order to find some inconsistencies and with Blockchain, immutable data.

2 Functional requirements

- Record a vote: People can vote as many time as they wish, the last vote should be the only one computed.
- Authentication to vote: Only authorized people can vote, the address should be on a white list.
- Possibility to audit the votes: The system should be verify every vote and that they are supossed to be there.
- Voter can be ensured that her/his vote has been counted.
- Integrity of the poll: this network ensures the integrity of the Blockchain by securing it so that no adversary can alter entries.
- Web interface to see and vote: Alogn with the smart contract, that should be a web application to see and vote.

3 Non-functional requirements

- Use Ethereum Network.

- Use as less Gas as possible: To run every transaction there is a cost, therefore the cost should be minimum to be viable.
- There is no super user that can have powers to change specifics of the contract

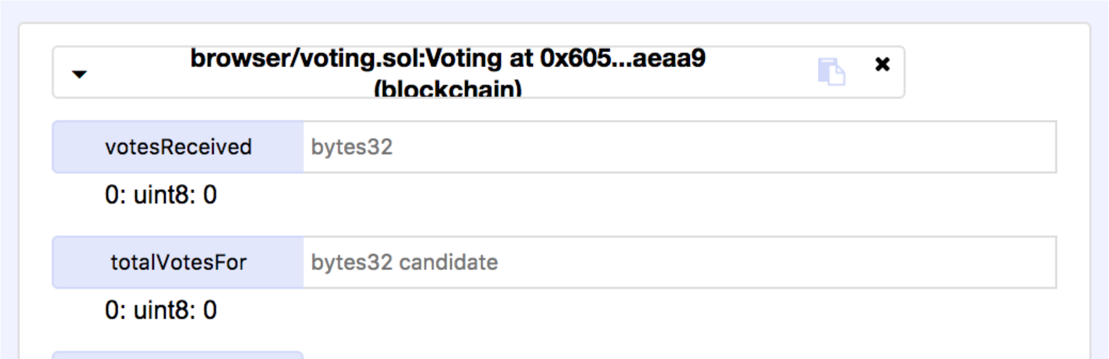
4 Proof of viability

To show that this project is viable, here is a piece of code written in Solidity, which is the programming language available on Ethereum network. This code is a simple voting app that initializes a set of candidates, let anyone vote their candidates and display the vote totals for each candidate.

Together with a simple web application, we can create a web interface for voting and visualizing this election result, showing that is possible to do this project, but our main goal is to be a full election process, where you need authentication to participate, along with more complex operations.

```
pragma solidity ^0.4.6;

contract Voting {
  /* The key of the mapping is candidate name stored as type bytes32 and value is
  an unsigned integer which used to store the vote count */
  mapping (bytes32 => uint8) public votesReceived;

  /* List of candidates */
  bytes32[] public candidateList;

  function Voting(bytes32[] candidateNames) {
    candidateList = candidateNames;
  }

  function totalVotesFor(bytes32 candidate) returns (uint8) {
    require(validCandidate(candidate));
    return votesReceived[candidate];
  }

  function voteForCandidate(bytes32 candidate) {
    require(validCandidate(candidate));
    votesReceived[candidate] += 1;
  }

  function validCandidate(bytes32 candidate) returns (bool) {
    for(uint i = 0; i < candidateList.length; i++) {
      if (candidateList[i] == candidate) {
        return true;
      }
    }
    return false;
  }

  function getCandidateList() constant returns (bytes32[]) {
    return candidateList;
  }
}
```

Using a tool, we can see the votes received and the total votes for everybody, it is possible to add a web page that reads this data and display in form of a table using web3 JavaScript and HTML.



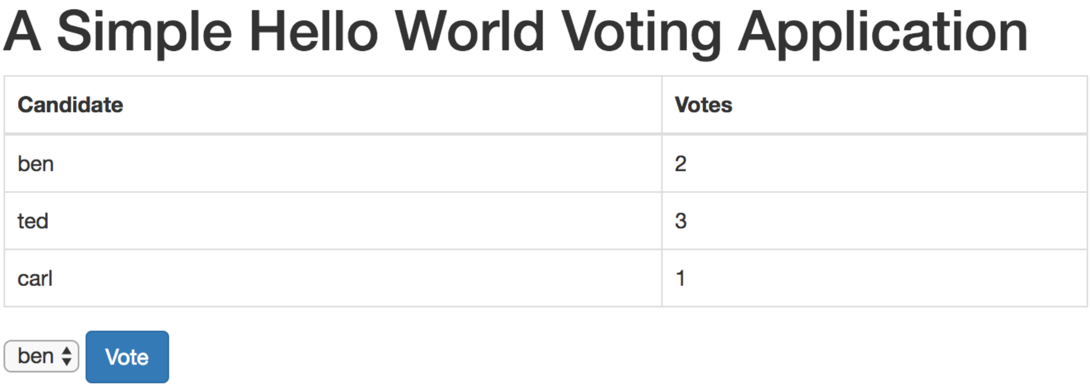
browser/voting.sol:Voting at 0x605...aeaa9 (blockchain)

bytes32

0: uint8: 0

bytes32 candidate

0: uint8: 0



A Simple Hello World Voting Application

Candidate	Votes
ben	2
ted	3
carl	1

Images taken from [2]

References

[1] Cardoso Luders Matias. Aranha, Yóssis. Execução de código arbitrário na urna eletrônica brasileira., 2018.

[2] Madrona Venture Labs. Ethereum voting app, 2017.

[3] Dominik Schiener. Voting on the ethereum blockchain: An analysis., 2015.