Voting System using Ethereum's Blockchain

Daryna Aloff	*Jonathan Pinder*	*Dilianny Lakitaya*	*Luong Ta*	*David Gladney*
Computer Science	*Computer Science*	*Computer Science*	*Computer Science*	*Computer Science*
Georgia State University	*Georgia State University*	*Georgia State University*	*Georgia State University*	*Georgia State University*
Atlanta, USA	*Atlanta, USA*	*Atlanta, USA*	*Atlanta, USA*	*Atlanta, USA*
daloff1@student.gsu.edu	*Jpinder1@student.gsu.edu*	*dlakitaya1@student.gsu.edu*	*lta2@student.gsu.edu*	*dgladney1@student.gsu.edu*

***Abstract*— Blockchain technology is currently one of the most discussed in a kind of revolutionary innovation, which already found a vast number of applications in various fields. When the world began to realize the possibility of using blockchain technology beyond cryptocurrency, one of the first directions was not related to cash transactions in which this technology began to develop – this was an electronic vote. There is a vast majority of news going on about the investigation of Russian interference in the 2016 United States elections, even still going on today. This is the area that the advantages of transparency and security of technology such as blockchain show their advantages most clearly. Also, the development of the Smart Contract technology in this direction is designed for digital processing, verification or ensuring the fulfillment of negotiations or contract execution, allowing to carry out reliable transactions without third parties. This paper addresses the issue of developing decentralized e-voting application based on Ethereum blockchain technology. Also, a lot of research is done on existing problem solutions. The novelty of the study lies in the fact that many of the complex security issues that electronic voting systems face today can be overcome if the replication, cryptography and verification mechanisms used by blockchain technology are used in their development. The application of this technology in the field of voting should have a positive impact on the safety and transparency of such systems, and, consequently, on the users' trust in them.**

I. INTRODUCTION

In all countries of the world, the circulation of votes is one of the most important problems that impede the electoral process. One of the main reasons for low turnout is associated with a limited period and place where voters can vote. In addition, the process of storing votes within one central authority is relatively risky. This is susceptible to hackers, as well as to manipulation. A decentralized application can solve both these problems and introduce affordable and easy-to-use digital voting while maintaining complete integrity.

The blockchain technology solves the above problems by forming a network of computers, each of which stores a copy of voting data, as well as a set of rules that nodes must follow to add new data sets to the blockchain [1]. Thus, all nodes can check votes for authenticity, and any falsification of data will require access to all available copies of information. In addition, the blockchain opens the door for quick and easy voting via the Internet, where voters can participate in voting from any place, be it work or even from home.

However, two questions arise: what set of rules can be configured for transparent and regulated voting? In addition, since voting information is available to all nodes in the network, how can the anonymity of voters be preserved?

There is a reason for having to go to the polling station to fill in election ballots. Anonymous ballots are the easiest way to protect the integrity of a vote while protecting the privacy of voters. Digital voting was a difficult task because it is difficult to verify that every ballot is valid and maintains the anonymity of voters. Blockchain-based voting can change this with cryptography.

In fact, blockchain-based voting already changes some choices. Right now, the military from West Virginia, USA, who serve abroad, can vote in elections using their mobile phones. Other countries, such as Brazil, Denmark, South Korea and Switzerland, are investigating this type of voting.

A. Related Work

Now there are some projects on the topic of voting using blockchain in its infancy. Among them, one promising project is worth highlighting - FollowMyVote. This project is created specifically for the implementation of elections through online voting in the United States. Using a webcam and government ID number, voters can remotely and securely register and vote for your desired candidates. After they have chosen their candidates, they can use their unique voter ID to literally open the ballot box, find their vote and check that it is present and correct. In addition, voters can monitor the election process in real time, when votes are cast.

No large-scale technical changes exist without concerns, and blockchain voting is no exception. FollowMyVote makes use of a cryptographic key by which a verified voter can cast their ballot. Some critics of blockchain voting argue that this technology will merely force malicious attackers to change the focus of their attacks, focusing instead on compromising voter keys.

For blockchain voting to be seriously considered as a solution to our nation's crumbling election infrastructure, we need to simultaneously see an explosion in wide-spread experimentation with this technology and further attempts to shed light upon the failures and dangers of our outdated and non-homogenous voting systems [2].

B. Our Contribution

We are proposing a voting system that allows users to vote anonymously. The voting system will provide authorization to participate in any voting activity form a central authority, for example, a government organization that issues proof of citizenship.

Our application will provide the platform for a NewsCenter like CNN to provide a poll to watchers. They can verify that all voters who are participating are, for example, US citizens.

When deploying a smart contract, a submitter will be able to verify who should be voting. This selection will allow a platform for allowing voting in democratic states that is transparent.

We will also like to add the functionality to accept votes from certain wallets. Meaning an institution like the University System of Georgia would be able to only survey Georgia State University students if they wanted to and ignore input from others. This will be explained later in the Voter ID Verification.

II. BLOCKCHAIN-BASED VOTING BASICS

Blockchain voting is like analogy voting. The same concepts and processes are used. To conduct digital voting, a citizen must register and prove his citizenship in this jurisdiction. This identification and nationality can then be recorded on the blockchain associated with this user key. Then the citizen needs a ballot paper. In a blockchain, this is likely to be in the form of a special token for voting, which will be entered into a user account. This token will also likely have a time limit in which it can be used for voting, after which it will burn itself with a smart contract or become useless.

When voting on a blockchain, a voting token (bulletin) will be sent to a specific address. Voters would know which address matches the candidate or referendum. Sending a marker to this address will constitute a vote [3].

Technically, it sounds simple enough. Voting is recorded by the blockchain, with its immutability, verifiable and transparency. The counting of votes in order to declare a winner in an election is, in this case, very simple. In addition, you are creating user interfaces that automate and hide the process of sending a token to a specific address. Instead, voters will see a simple online interface to select a candidate or proposal and click "Submit."

III. VOTER ID VERIFICATION

One of the main problems is the verification of the identity of the voter. In order for a blockchain voting to work, we need a system that prohibits people from voting more than once or voting in elections if they are not citizens. This becomes difficult for the blockchain since it doesn't rely on the central authority to verify documents for citizenship or residence.

The solution is likely to rely on sending passports or scanning a driver's license. This identity can then be associated with a mobile device using a password and two-factor authentication or biometrics (for example, a fingerprint or retinal scan). The idea is to verify that the person who submitted the citizenship documents is the same person who is actively behind the computer or smartphone during the voting.

However, as soon as the identity and voting rights are confirmed, it is necessary to separate it from the bulletin itself. It is important to note that one of the key parts of democracy is the secret ballot. No one should know who voted for whom and to prevent external influence on the vote when voting, the information that is registered in the blockchain must not include identifiable information; this means that the information about the sender of the voting token must be hidden. There are various ways to achieve this, including zero evidence of knowledge, ring transactions, or various encryption methods. Each has its advantages, disadvantages and technical problems. True anonymity at the same time as a proven identity is a big problem with block voting.

IV. REALIZATION OF THE E-VOTING SYSTEM

When developing DApp, there are three main steps: writing a smart contract, deploying a smart contract to the Ethereum blockchain network and connecting a graphical user interface. The smart contract plays a crucial role in the development of a decentralized application. It describes all the functions that will allow the user to perform certain data manipulations.

E-voting DApp should meet certain functional requirements. Fig. 1 shows the model of the voting system, which shows the main functions of the application.

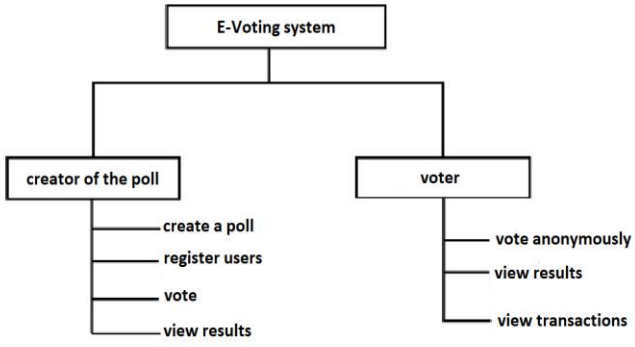

Fig. 1: Model of the e-voting application

The main functions provided by the voting system are:

- *create a poll and be its administrator with the ability to register voters;*
- *cast your vote for a particular object of voting;*
- *view the voting results;*
- *view transactions related to the smart contract to make sure they are honest.*

When accessing a smart contract, it allows the user to create a poll by transferring an array with voting elements, such as: the

name of the poll, the question being discussed, voting options. After creating a poll, he can register new users in his poll, who after registration will be able to vote in this vote. Also, the smart contract provides the ability to access the block with data on the voting results. In addition, the user has the ability to view the transaction history, and this option is provided by the blockchain network in which the smart contract is deployed.

V. PROPOSED PROTOCOL

Selection of a smart contract development tools

For this project we used Solidity - high-level language for EVM with a syntax close to the programming language JavaScript. Its similarity with JavaScript allows to quickly adapt to the writing of smart contracts for developers who are engaged in web-programming. In addition, we also needed the node.js software platform for developing Dapp, which has many useful libraries.

Also, we have used Ganache-CLI tool, which allows to raise the blockchain simulator with the RPC protocol enabled on one device. This tool creates 10 trial accounts in this blockchain. The balance of each test account has 100 ETH. This procedure allows to speed up the testing of the written program, since there is no need to spend time raising a real private blockchain, creating accounts, etc. In addition to Ganache-CLI, node.js has such an important development library as Web3.js. This library allowed us to use the ethereum API using regular JavaScript. Metamask was used in order to provide a connection to the desired Ethereum network through its remote node, thanks to which the user does not need to deploy the node. Metamask is the de facto standard for dApps on the web. It injects a Web3 instance into a window object making it available for JavaScript code. Metamask allowed us to create new or import existing accounts to work with the blockchain. This makes it possible to sign transactions with a private key that is stored locally with the user.

For the front-end it was decided to use ReactJS framework. Although React has a higher learning curve, it makes application development simple and easy to understand. In addition, it can be ideal for complex and high-performance software solutions. In this project we implemented truffle boxes, that are helpful boilerplate code, pre-configured to help you get up and running quickly developing your dApp. We used react box which is essentially an example truffle project which has been merged with an ejected create react app to create a barebones solidity and react example dApp.

Logical simulation of the voting system

Figure 2 shows a sequence diagram showing the interaction of objects.

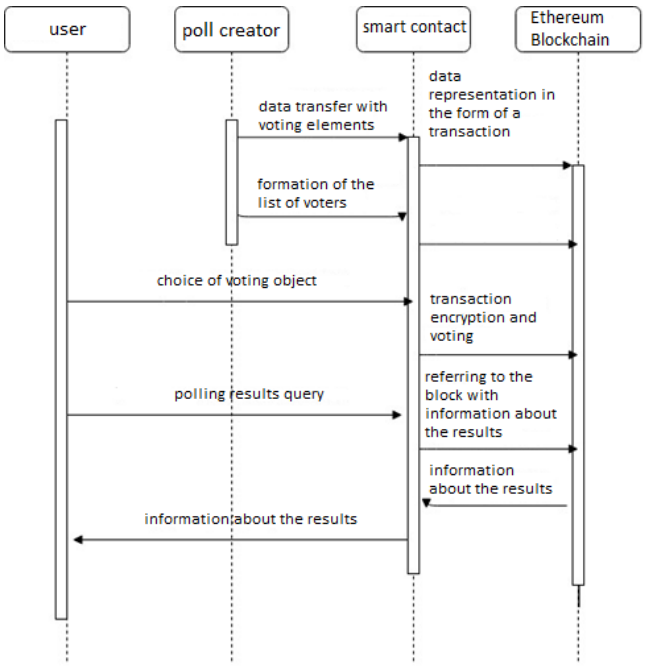

Fig. 2: Diagram of the sequence of the electronic voting system

The interaction between objects and subjects occurs as follows:

1. The developed smart contract is addressed by the creator of the new voting and creates a new poll.
2. Data such as “Name of the vote”, “Discussed question”, “Voting options” are formed into a block and sent to the blockchain network.
3. The creator of the vote registers voters in the system by adding the addresses of their accounts to the voter list, which gives them the right to vote in this poll.
4. The formed voter list is sent to the blockchain.
5. The user accesses the smart contract and selects the object of voting in the survey.
6. The name of the voting object for which the vote was cast is encrypted and sent to the blockchain.
7. The user accesses the smart contract in order to find out the voting results.
8. Smart contract accesses the necessary block with information in the blockchain.
9. Information about the voting results is displayed to the user.

VI. IMPLEMENTATION OF BASIC FUNCTIONS

In the case of a voting system being developed, a structure is used at the beginning of the contract, that is, a complex data type to which the main functions will refer.

```
struct Poll{
    string name;
    string voterType;
    string choice1;
    string choice2;
```

```
        string choice3;
        uint voteCountchoice1;
        uint voteCountchoice2;
        uint voteCountchoice3;

struct Candidate{
        string name;
        uint voteCount;
    }
    struct Voter{
        string identification;
        bool canVote;
    }
```

When a user votes, a new Voter struct is created and added to the mapping. In order to count the number of votes a certain candidate has, we loop through all the Voters and count the number of votes. Thus, these mappings will hold the history of all Voters. It implements require statements that will stop execution if the conditions are not met. First require that the voter hasn't voted before. We do this by reading the account address with "msg.sender" from the mapping. If it's there, the account has already voted.

These structures will contain such data as poll name, discussed question, voter type, vote counters for each of the candidates, the status of that vote or a different account or not and whether the survey was created or not.

In order to carry out an intermediate verification of the smart contract, use the Ganache-CLI tool, which runs from the bash-console and creates a local test blockchain with ten accounts. It is called from the nodejs console, using the ganache-cli command, after which we are shown 10 test accounts.

I. Obstacles

Some of the obstacles we faced came from interacting with functions from the front-end. The main issue was using the props and components that React offers to pass user input to our functions used. After following a number of tutorials we were relatively unsuccessful in getting all of our functions working properly.

VII. Obstacles

Some of the obstacles we faced came from interacting with functions from the front-end. Passing parameters from html object to react components and calling the smart contracts functions was a relatively challenging obstacle.

The main issue was using the props and components that React offers to pass user input to our functions used. After following a number of tutorials we were relatively unsuccessful in getting all of our functions working properly.

II. Conclusions

The development of technology and society is pushing humankind to search for cheaper and more reliable methods of voting. The use of blockchain technology in voting can solve most of the problems of existing electoral systems. In this paper, we proposed a system of online election based on Ethereum blockchain smart contracts. The system supports full-featured online voting, and the survey results will be calculated automatically and anonymously. Compared to traditional voting, electronic voting is a more economical system, more transparent and impartial.

It was also challenging to observe the blockchain using web3 and also, to call functions from the contract. But we were able to create a contract that was deployed and we were able to observe transaction blocks using Metamask and Gancahe.

References

[1] A. Sandre. (2017). Blockchain for voting and elections. Available at: https://hackernoon.com/blockchain-for-voting-and-elections-9888f3c8bf72

[2] F. Caiazzo and M. Chow. (2016). A blockchain implemented voting system. Available at: http://www.cs.tufts.edu/comp/116/archive/fall2016/fcaiazzo.pdf

[3] P. McCorry, S. Shahandashti and F. Hao. (2017). A Smart Contract for Boardroom Voting with Maximum Voter Privacy. Available at: https://eprint.iacr.org/2017/110.pdf