Voting through the Blockchain: a design approach

October 2018

Abstract

Our paper's goal is to review and implement a state of the art voting system using Blockchain and Smart Contract.To this end we first have a wide overview of currently existing electoral system and try to answer which electoral system should be more suitable for blockchain implementation. Our approach will be to design a contract using widely used frameworks and implement it on a test network for use. Our research will cover the pros and cons of a Blockchain enabled system. We will discuss on the feasibility of such a Blockchain enabled voting system. Given our results we conclude that voting systems embedded in the blockchain can be a safe and advantageous way of conducting elections. It gets rid of traditional IT security problems but brings forth blockchain developers with new administration problems such as registration of voters and costs related to the core of blockchain technologies.

Contents

1 Reviewing new voting systems

1.1 E-voting

Nowadays, obstruction of the polls is one of the biggest problems voters face. It is the most common voter suppression tactics with the objective to lower voter turnout and undermine confidence in the electoral process In the era of digitalization, electronic voting (referring to any form of voting making use of modern technologies to both cast and tally votes) may immediately seem like a good solution [7]. In fact, the topic is raising attention and many countries line Switzerland [14], UK [35] and US [27] have already tried to implement it in some trials, mainly introducing voting machines. Although the main reason to find new solutions is to increase voter participation by giving them multiple opportunities to cast their vote, there is many more benefits associated with electronic voting.

Some of the advantages include:

- An online system could be used independent of the platform and device
- Counting time and human error reduction
- Cost reduction
- Auditability, transparency, security and accuracy
- Increased accessibility for people with disabilities, citizens abroad and the public in general

On the other side some disadvantages are :

- Vote manipulation (data breaching or fraud)
- Trust to the service company
- Vote secrecy is not guaranteed
- A recount is not possible

The principle in e-voting is to keep it as similar as possible to regular voting for security, trust and legislative principles. Also, from a technical standpoint, the system must be simple, transparent, auditable and reusable.

1.2 The Estonian case

So far, one country was able to implement electronic voting: Estonia [33, 25, 36, 34]. The country has been discussing the option of electronic voting since 2001 with the objective of using digital technology to improve efficiency, effectiveness and user experience in the public sector [33]. The project is part of the country's information policy for the upcoming years, which includes the introduction of e-services in all state agencies and training and knowledge-raising

activities for the society. Other than following the general necessary principles for e-voting, the Estonian government decided to structure the process using: ID cards for voter identification, possibility to re-vote deleting the previous choice, priority of the vote given in the traditional way over e-voting.

The first election in which Estonians were able to electronically vote was the national elections in 2007, where more than 30'000 voters used the new system. This allowed some studies to obtain some very interesting conclusions about the efficiency of the system. (Report for the council of Europe, EEI)

Firstly, e-voting remains an instrument popular mainly among young voters. Lack of familiarity with internet technologies (in particular computing knowledge and frequency of Internet use) seems to be the biggest driver, but this problem is expected to disappear over time, when digital native generations will constitute a bigger part of the voters.

Secondly, the trust in the e-voting mechanism remains another big problem. Of course, increasing the citizens' faith in the system counting and producing results correctly would help the cause and increase the voting turnover.

Thirdly, it was found to be neutral in sociological aspects as gender, income, education. This responds to some potential critics about the system introducing biases in the process and not performing well on the democratic level. Furthermore, it was found that e-voting is also politically neutral, not introducing political effects but enhancing the importance of the faith in the politicians.

Finally, the studies showed the change within the communication strategies. Internet information is becoming more important for political parties as their voters use the web as primary form of information. Also, the traditional information channels (TV, radio, ads, events, emails) are losing their effectiveness in reaching the audience of the internet voters.

1.3 Main problems

Thanks to the Estonian case, e-voting is recognized as possible and efficient. Nonetheless, there is a number of aspects that impact the efficiency of this system. The first concern is about fraud and privacy [50]. Remote balloting comes with the risk of having your credentials stolen, vote selling and coercion: this is why it is suggested to allow to vote multiple times, validating only the last valid vote. An uncontrolled environment such as in e-voting cannot guarantee the privacy of the paper balloting. Also, the technology and the sociological aspect have been a problem. The need for a card reader and the software installation might be a constraint in the diffusion of the method. The knowledge required to use the technology is an important factor as well. At last, citizens have doubts about the confidentiality of the system. It must not be possible to relate a vote to a specific voter, as the trust from the voting public is one key

factor to the success of the project and its legitimacy.

1.4 Solution

To solve some of the problems mentioned above, our recommendation is to introduce blockchain technologies, a peer-to-peer network that ensures confidentiality and that votes cannot be modified by third parties. As for the citizens, this technology could solve the trust problem: everyone can control and verify their vote was counted correctly, the count will be correct and the institutions are able to check and audit the results thanks to the decentralized system. Another problem, the hacking one, would disappear. The absence of a centralized database would force them to steal votes one at a time, and they would still need the secure voter ID, reducing the efficiency of the operation at a point where it is not convenient anymore. Furthermore, blockchain achieves privacy for the individual and transparency for the system as a whole.

2 Potential applications of Blockchain technology to existing voting systems

2.1 From Bitcoin to Ethereum

In 2018, Satoshi Nakamoto published a white paper presenting the Bitcoin initiative which aims to transfer online payments from one party to another without relying on banks or intermediaries[39]. It can not be denied that today Bitcoin is the most famous cryptocurrency with a market capitalization of $67 billion. Combined with Ethreum the two cryptocurrencies account for over 70% of the market. In fact, since the publication of Satoshi Nakamoto, more than 1,600 cryptocurrencies have been created.

In 2014, the **Ethereum Foundation** launched a project to create a decentralized platform that runs smart contracts using *blockchain* technology. To do so Ethereum is build on a blockchain with a built-in-Turing-complete programming language. These blockchain allows anyone to write smart contract using solidity language. This Turing-completeness language makes Ethereum different from Bitcoin and opens large possibilities. As a reminder a language is Turing-complete if it can be used to simulate any Turing machine. This means that the system can solve any problem regardless the time and the cost of the problem.

The philosophy of Ethereum is based on five main principles[11]:

- **Simplicity** : the Ethereum protocol should be as simple as possible
- **Universality** : developers take control of the blockchain with the Turing-complete scripting language

- **Modularity** : the Ethereum protocol should be designed to be as modular and separable as possible
- **Agility** : the Ethereum protocol should be flexible and can be modified in time
- **Non-discrimination** and **non-censorship** : the Ethereum should not attempt to restrict or prevent specific categories of usage

2.2 The Architecture of Ethereum blockchain

The Ethereum system is build on *accounts*, characterized by a 20-byte address. Each Ethereum account contains four field : a nonce, the current ether balance of the account, the contract code (optional) , the storage (optional). Here we have to make a distinction between *externally owned accounts* and *contract account*. Contract code, or smart contract[15] are controlled by their code and the other type of accounts are controlled by private keys (that is to say people). Externally owned accounts can interact between them making transactions. In the Ethereum blockchain transactions are gathered into blocks which are after chained with a hash function. Each Ethereum transaction is defined by [51] :

- **nonce**: A scalar value or a counter to make sure each transaction can only be processed once
- **gasPrice**: A scalar value equal to the number of Ether to be paid per unit of gas for all computation costs incurred as a result of the execution of this transaction
- **gasLimit**: A scalar value equal to the maximum amount of gas that should be used in executing this transaction
- **to**: The 160-bit address of the message call's recipient or, for a contract creation transaction
- **value**: A scalar value equal to the number of Wei to be transferred to the message call's recipient or, in the case of contract creation, as an endowment to the newly created account
- **v**, **r**, **s**: Values corresponding to the signature of the transaction and used to determine the sender of the transaction

2.3 Relevant situations for Blockchain implementation

When considering common elections such as the national and supranational ones, one might disregard the great organization that lies behind; institutions must agree on a timing and locations where to conduct the votes, find the mechanisms and people that will verify the voters' identities and record their votes, while also identifying reliable methods for counting the votes[1]. These

are some of the steps where flaws may have progressively appeared over the two last decades [8]. For instance, recent important elections such as those for the French presidency or the United Kingdom's (UK) prime minister position have been shunned by majorities of the French and British voters. Though it may not justify this phenomenon entirely, Allen et al. (2017)[1] argued that the voting procedure was regarded as time consuming by some voters as in the case in the France because of the identity checks run by assessors for instance. Yet, most western institutions have made the move to adopt new technologies within their existing methods. In Australia for example, one cannot register to vote if he/she does not provide a valid official document (Driving license, Passport, etc.) that confirms one's identity. Not only do such improvements require less efforts from the voter and the assessor, but they also represent a great cost reduction since there is a lower need for assessors.

As previously stated by other academics [45], institutions have shown a desire to evolve with time, and as such to implement new technologies within their voting systems. Technological advancements may help them move further and faster, and for some, even address existing major issues such as potential fraud (REF needed). That is the case with blockchain technology which has exponentially grown in recent years after Satoshi Nakamoto issued his white paper introducing the notion in 2008.

Based on cryptography, blockchains allow multiple individuals to share information and conduct exchange in a decentralized way while remaining fully anonymous. A blockchain is no more than a wide book distributed and accessible to all users who wish [5]. The technology spread quickly mostly thanks of its primary utilization that is cryptocurrencies such as the bitcoin [10] which consisted in a new currency, independent from any banks nor institutions, to use with the help of financial intermediaries. Nonetheless, the potential of blockchain technology does not only lie in its monetary applications. Various authors before us have stated how useful it could be for institutions as it would be able to dis-intermediate and decentralize law, contracts and government [4], Vigna and Casey 2015[48]. Blockchain are often argued to be a reliable alternative to existing organizational hierarchies including firms and governments, mostly because of it is a technology that aims at creating new decentralized institutions [18]. Thus, believe that blockchain might represent the future of voting as it is a reliable solution addressing most of the existing flaws. This statement is in line with the words from other authors before us such as Barnes and Brake, 2016, as well as Osgood, 2016[41]. In 2016, Davidson et al.[18] even gave this solution the name of 'crypto-democracy'.

According to the Institutional Possibility Frontier (IPF) framework, there is a trade-off to be found between the cost of dictatorship and the cost of disorder while organizing a vote [1].

The former consists in the consequences of a great centralization, where rulers' governance follows their personal preferences rather than their peoples'

ones. Negligence is also considered part of the cost of dictatorship. During the 2012 French general elections for the presidency of the political party 'L'Union pour un Mouvement Populaire' (that is the Union for a Popular Movement). Each candidate claimed his victory once the elections closed. Even though the counting of the votes indicated Jean-François Copé as the winner, François Fillon contested this victory because he claimed that some votes were missing, mainly in the South East of France (Le Point, 2012)[44]. Whilst such negligence might have been regarded by some as fraudulent, others considered it to be a human error. Hence, this example perfectly illustrates why negligence is regarded as a contributor to a higher cost of dictatorship.

On the other hand, the cost of disorder is the result of high decentralization, where there is great risk of drifts such as individuals being able to vote more than once and committing the offense of impersonation for example [19].

Allen et al. (2017)[1] illustrate their saying by using a graph (figure 1) where there are both a highly centralized electoral authority that controls the voting process, and a highly decentralized system with a great competition among the several electoral authorities as is the case in most public companies and organizations.

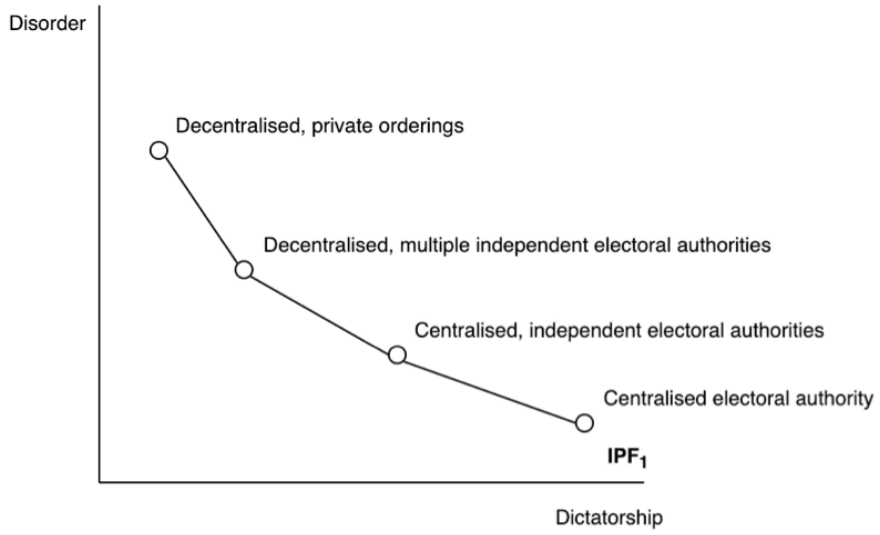

Figure 1: Institutional Possibility Frontier framework

Because of its nature in itself, blockchain technology appears as the solution to decreasing both the cost of dictatorship and the cost of disorder [31]. In other words, uncertainty and opportunism have a lesser effect over the voting process (Figure 2). Moreover, as put forward by FollowMyVote.com (2017) – one of the world's largest online voting platform –blockchain nowadays represents the

most adequate tool to consider the following criteria: security, transparency, freedom, fairness, anonymity, and accuracy.

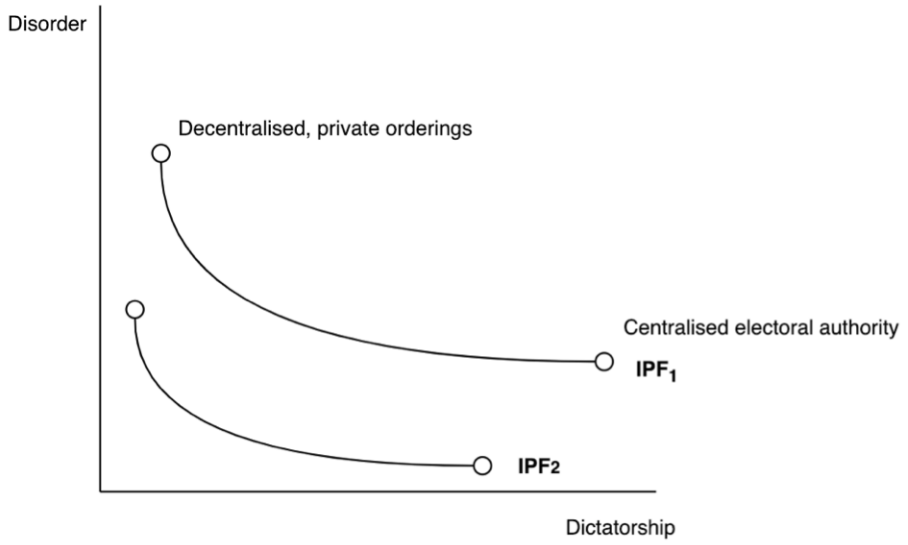

Figure 2: The IFP framework after the introduction of the blockchain technology

In their work on crypto-democracy, Allen et al. (2017)[1] suggested that blockchain also represent a viable solution to reducing the costs of a voting process, notably by affecting the costs of coordinating, monitoring, and consensus. Moreover, they considered the potential of blockchain both in a decentralized and centralized institutions. The former would benefit the system as blockchain would reduce many of the costs of disorders such as fraudulent registration or duplication for example, without having to set a central control. There is value in such implementation for the latter because of it would make the entire voting process more transparent and honest, while reducing the human errors. Hence, it would be much harder to manipulate the process.

3 Designing Contracts for voting

In this part we will discuss about the implementation of a smart contract for voting. The implementation of a voting system on a blockchain technology should fill few requirements [41] :

- Voters should be able to check that their vote was counted
- Voters can only vote once in each election
- Voters cannot see the votes of other voters

- The system should not enable coerce voting
- The system should either produce or hide interim results
- The system must allow for voter abstinence
- The system must be audit-able
- Only citizens registered to vote can vote
- The most practical system requires the least amount of behavioral change for voters
- Different kinds of elections can be implemented into the system

3.1 ETH Blockchain voting APP architecture organization

In this subsection we design a prototype that would be the core of our contract regardless of the electoral system. We set up our system and create a front-end for our application for voting. We present how our system should be designed overall and how it should theoretically be secured. We also present the additional set of actions that should be performed to make the voting system we implement as cost effective and secure as possible.

Organization of elections is a very important topic for democracies and groups that value individual decisions to decide on common matters. To this regard traditional voting systems are made to be as safe as possible in order to prevent malicious individuals from influencing the outcome of the election with illegal techniques such as stuffing ballot boxes or changing the count of votes [20]. Other new ways of directly intervening into elections would be to hack directly into servers containing vote counts or other important information. Here Blockchain plays a very important role in the sense that it holds network accepted data that is also immutable. A very common approach in traditional voting systems is to identify oneself at the entrance, check if the voter has not already voted, let the voter pick different candidates as well as a letter publicly and then let him go in a private place where the voter can decide on which candidate he decides to give a vote to. Eventually the voter comes out of the private place and publicly put the anonymous letter in the ballot. Then the voter signs a public ledger asserting that his or her vote is counted for this election. The vote is now done. In this traditional process voters have to take envelopes and different candidates in a private rooms and then put publicly their envelops into the ballots. This means that at the end of the day some serious security issues may emerge. Also there could be some human errors while counting the votes [22]. To avoid these traditional human error issues some countries have decided to implement machines that would count votes from security checked voters. But other problems may arise with the upcoming of these voting machines. A concern would be that these machines get hacked and the administration can

no longer evaluate whether or not a vote is "human made" or the result of the hack. Servers could be hacked, thus compromising the entire election. Here you can find a scheme of a typical online voting system:

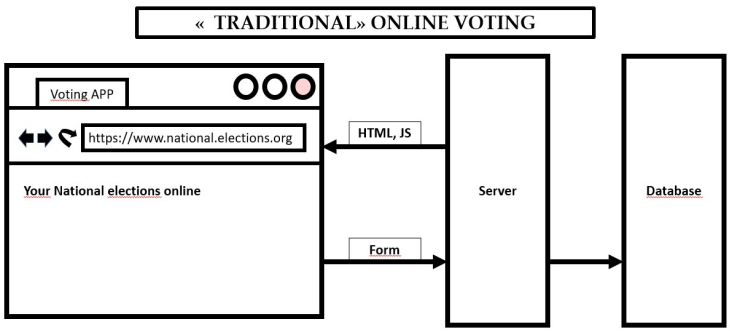

Figure 3: "Traditional" online voting

Here we can find multiple vulnerabilities that may be used by potential hackers. The first one is to hack the servers and tweak the code injected to the voter so that they get a virus or simply chose a given candidate no matter what actions they perform on their browser. A second threat is that the proper database could be attacked by hackers.

With recent political matters and interference of countries onto other countries' elections a safe election would guarantee the validity of the results and democracy could be preserved during the step of voting.

In order to design our decentralized application we have decided to use the Ethereum Blockchain and their integrated tools that are designed to facilitate the development of smart contracts[12]. Smart contracts are pieces of code that Ether holders can interact with. They can serve different purposes such as transferring value, register intellectual property, voting and many other possibilities. We have thus designed a Voting contract in the solidity programming language[17]. The solidity programming language was developed by the Ethereum foundation and is constantly being updated.

What is characteristic of solidity smart contracts is that they are hosted on nodes of the Ethereum network. The have and address and anyone can interact with them. Their code is even accessible to anyone thus no secrecy is allowed. Eventually what that means is that the code in the contract should not have any security breach whatsoever since this would eventually result in a hack of the contract. In order to interact with our Ethereum smart contract that implements a voting system we need to use two particular technologies. The first one is "web3.js"[49], this is a library that interacts with the ABI of the contract. The ABI is the Application Binary Interface, it is automatically

stored along with the address of the contract and is used to interact with it through various languages. Web3 can be implemented in the browser and we can then use it along with "Metamask", a wallet that allows user to hold, send and receive Ether, to make the transactions to the contract. The reason one needs Metamask to vote is that any operation in the Ethereum blockchain just like in any server requires computing power. Computing power is not free and, in Ethereum and other Blockchains, developers call it "gas". The gas is an amount of ether that is used to pay the miners that secure the transactions we make through the contract.

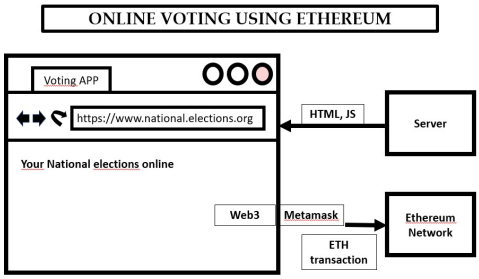

Figure 4: Ethereum voting system

We will later on study this matter in more detail but one should consider that for any transaction to occur, gas is needed. This means that at a bigger scale any user would need some amount of ether to vote. To do this another feasibility study should be done but setting up a "private" or "specific" network in the Ethereum blockchain would definitely do the trick in order to tackle such a problem[43]. The miner would be the administrator of the smart contract i.e the state. This would set another difficulty to the administrator of the contract. Blockchain networks are vulnerable to what is called the "51% attack"[32]. This kind of attack is when some entity which holds more than fifty one percent of the total computing power of the network decides to take control of the network and has the power to chose which data is true and which is not.

What can be done to counter this disadvantage that is inherent to Blockchain technology is to hold the election in a public network. However this would be even more of a hassle in order to set up the system properly since it would involve setting up an account for each and every registered voter and giving the voter the right amount of ether in order to vote. Also, giving the voter some amount ether would be resulting in giving away some money for the citizens to spends just for the election and some citizens would value more this money to spend it on something else if possible. In order to avoid this scenario we would need another applications and contract that would be a transit from the voting account and this transition account would then send the gas to the election contract. This additional step incentives citizens to vote rather than spend is

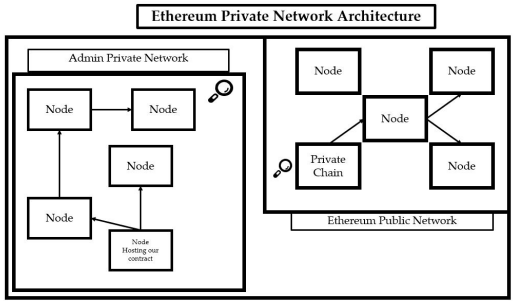

Figure 5: Ethereum Private Network

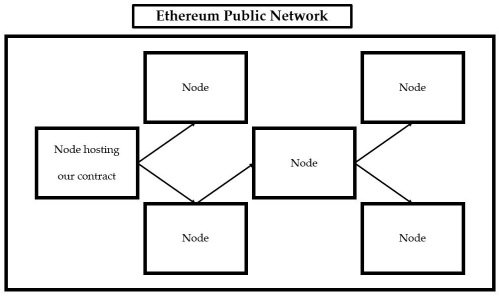

Figure 6: Ethereum Public Network

necessary and has to be taken into account when weighting the pros and cons of such a system.

3.2 Security

Here we actually implement our voting contract in the ETH test environment. We present the way we actually created our contract with Ethereum. We cover the issues and implementation problems that could be avoided and show how we did avoid them. We also present the additional requirements for a wide adoption of the technology.

As we said it before we designed our Ethereum contract with solidity. It grossly resembles the following figure:

However, our contract does not cover the entire secure process since the system would need some additional features in order to make sure a registered voter is a citizens who holds the right to vote. Another important feature of voting systems is that vote should be anonymous and no one should be able to tell who voted for which person. In order two prevent this two options should be studied. The first one is explained in the following figure as it depicts the

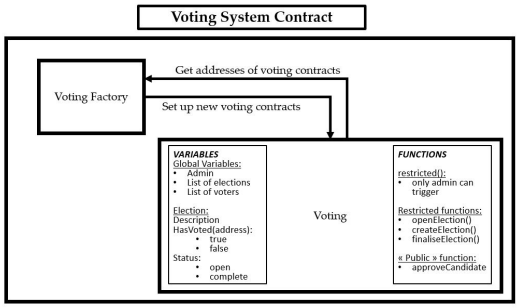

Figure 7: Voting Smart Contract

process of registering a voter in the system using a cryptographic hash function with and additional "salt" to create anew addresses and accounts. That would ensure anonymity of ethereum addresses[31].

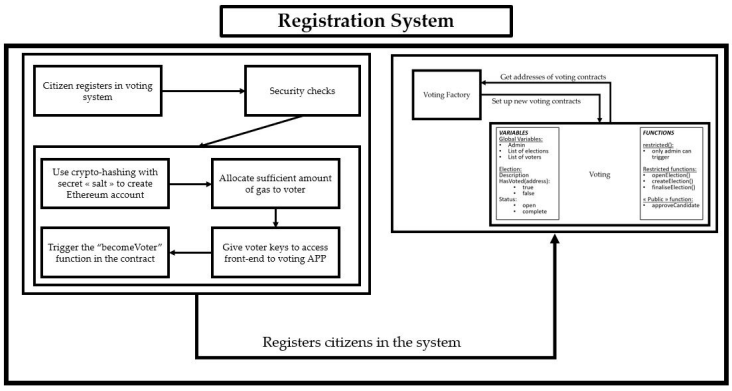

Figure 8: Registration Process

Another way to work around the problem would be to truly implement secrecy and privacy to the Blockchain protocol. This would require changing the paradigm of Ethereum. Working outside Ethereum is not really the problem here but it would require building an entire new blockchain and maintaining it with all security issues that come along (51% attack and other threats). Nevertheless other Blockchain protocol have already been implemented in order to tackle similar secrecy and privacy issues. Monero is the most famous one[29]. This blockchain is maintained by an active community of developers and is constantly tested by hackers so as to improve the security of these very networks. Having an anonymous way of voting would theoretically make the administration work a lot easier since it would make the final system secure and legal.

However determining if a vote comes from a legitimate voter becomes impossible since creating uncertainty on the security of the blockchain voting system as a whole. If administrators would consider the implementation of such an "anonymous by nature" protocol than they should focus on registration security first and keep monitoring registrations as well as the activity of the network in general.

3.3 Case study

In this subject we have a reflection on how voting systems actually are implemented and if our implementation is beneficial for some particular sets of situations. We will have our own voting system tested. And we show our feedback on our experiment.

Each time a user of the Ethereum blockchain wants to operate a transaction, deploy a smart contract or interact with a deployed contract, he has to pay a small amount of Wei to the miners of the blockchain. In fact, miners are providing the blockchain with their computing power but they expect an indemnification in rewards. This small amount of Ether is called gas and its value in Wei is specified by each person who wants to interact with the blockchain. Each operation operated on the blockchain has a defined cost of gas. For instance the following table summarizes the cost of gas of most common operations.

Operation	Gas cost
Addition	3
Subtraction	3
Multiplication	5
Division	5
Equality comparison	3
Storage	5000-20000

The price of Wei is exclusively depending on the market. On December 18 2018, one Ether equals 94,74 US dollars and the average gas price equals to 13 823 213 704 Wei so roughly 1.3e-06 US dollars. Even if this amount of money seems negligible at first sight, it is vital for smart contract developers to avoid bad practice in order to reduce cost [13]. In practice, smart contracts process lots of operations[21] and can also provide public methods usable by various clients unlimited number of times. In the case of a voting system, millions of persons will have to interact with the blockchain[38]. That is why, we have to be very careful about the gas consumption.

Voters have to be authorized to vote by the government or by the organization managing the election[31]. At first glance, we could think that storing all the addresses of the persons allowed to vote in an array can be a good solution. However, doing so, we would require a tremendous amount of gas to pay provided the elections would gather thousands or even millions of people.

In fact, our voting system have to check if the voter is authorized to vote and if he has not voted yet, for each voter. That is why looping over the arrays of authorized voters and persons who have already voted to check if they are voters or if they have voted is not a good idea. Developers here should use another data structure called mapping in order to avoid linear time search (figure 9). In these data structure, keys are not stored and values are accessed with a Hash function which enables a constant time search.

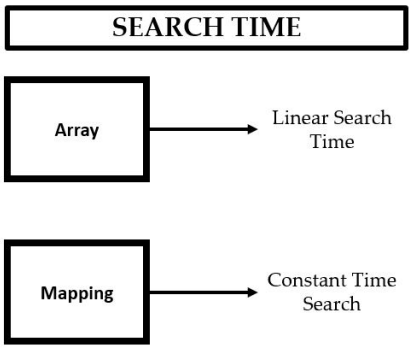

Figure 9: Search Time between array and mapping

The structure of our voting system is based on two smart contracts. The first one, *VotingFactory* is used to deploy new electoral campaign on the blockchain. An electoral campaign can gather multiple elections. Each election can be carried out by the admin of the contract. The administrator has access to restricted functions that allow him and only him to run the elections as he wishes. Even though any one has access the contract and can see the code, it is designed to only enable voters to vote, admins to control the election. This means that even if the election is open there should be no way of taking control of the smart contract. Here the actual design of the contract is crucial since a flaw of security inside the contract would lead to a major failure of the voting system. As discussed previously, we implemented a voting contract that assumes voters to be legitimate voters. This is an external part that we did not implement in a front-end. What we mean by front-end is an interface that is able to communicate with the contract. The front-end we designed however enables registered voters to vote in the election. Admins can also control different states of the contract: create elections, open them and finalize them. In no way administrators can alter the results of the elections. If implemented correctly in a bigger system including a secure voter's registration and easy accessibility to front-end technology like Metamask we could seriously consider the creation of such a system nationwide. It could either be used at home with personal computers provided that voters are familiar with the new voting system and the system being proven

to be secure or in a new sort of ballots in voting rooms. These new ballots could integrate facial recognition technology or other ways of securely identifying voters without the need of human interference. Like ballots that have already been put in use in some states of the United States of America to vote through touch screens, these very touch screens could be the actual front-end interacting with the contract we have designed for the purpose of out research. The advantages of having physical touch screens is double since enables easier security processes and gets rid of server related problems as well as ballot stuffing. However, it would not technically appeal more to people who are not eager to vote than other regular voting systems.

To test our voting system we deployed the *VotingFactory* on the **Rinkeby** Network. The Rinkeby network was implemented to help developers to test their smart contracts. In this network, developers can have ethereum for free. Our contract is deployed on the following address on the Rinkeby Network : "0xCAaf832Ad3cE44f3FCB68604EA5AC3Bc2F5247D4". This contract can be used to create new *Campaigns*. Once the campaign is created, the administrator, that is to say the person who created the campaign can create and open new *Elections*. To have a more user friendly interface, a front-head interface was created with React, a Java Script framework created for building user interfaces on the browser[37].

4 Conclusion

Throughout our research we have first reviewed new voting systems. To this end we first focused on E-Voting, its current state-of-art implementation and different ways it can be used for different purposes. We then focused on the Estonian Case for E-Voting. This case helped us better understand practical implementation of E-Voting and the different challenges one may encounter. We naturally decided to take some time to identify the main problems of E-Voting in detail. This brought us to a possible solution to tackle this problem: Blockchain technology. Our research oriented itself to blockchain, its characteristics, flaws and advantages. We decided to apply what has been discussed in the "potential applications of Blockchain technology to voting systems" in a final Design part of our research. We first described how blockchain voting systems are practically designed. To do so we studied the subject on an organizational level as well as on the technical level. We then made a focus on one of the most relevant challenge of the voting subject: Security. Eventually we presented a possible implementation of smart contracts that can be tested and used in production.

List of Figures

Smart contract solidity code

```
// credits to Stephen Grider which Udemy's course @StephenGrider/EthereumCasts
// inspired us to create this contract
// @MCSZN
pragma solidity ^0.4.17;

contract VotingFactory {
  address[] public deployedCampaings;

  function createCampaign(string description) public {
        address newCampaign = new Voting(description, msg.sender);
        deployedCampaings.push(newCampaign);
    }

    function getDeployedCampaigns() public view returns (address[]) {
        return deployedCampaings;
    }
}


contract Voting {
  struct Election {
    string description;
    bool open;
    bool complete;
    uint yesCount;
    uint noCount;
    uint blankCount;
    mapping(address => bool) hasVoted;
  }

  Election[] public elections;
  address public admin;
  string public descriptionCampaign;
  mapping(address => bool) public voters;
  uint public votersCount;

  // this function creates a restiction blocks other funcs
  // only admin can execute restricted function
  modifier restricted() {
    require(msg.sender == admin);
    _;
  }

  // Contrustor method only called once by admin
  function Voting(string description, address creator) public {
    admin = creator;
    descriptionCampaign = description;
  }

//  function getDeployedElections() public view returns (Election[]) {
//      return elections;
//  }
```

```
53
54  function getSummary() public view returns (
55     uint, uint, uint, address
56    ) {
57      return (
58        this.balance,
59        elections.length,
60        votersCount,
61        admin
62      );
63  }
64
65  // this function is called only after security processes in
        browser or other process
66  // increment count of total voters
67  function becomeVoter() public {
68    voters[msg.sender] = true;
69    votersCount++;
70  }
71
72
73  // election can only be created by the admin
74  // it belongs to the wider Voting contract
75  // the election can be accessed through the contract by registred
        voters
76  function createElection(string description) public restricted {
77      Election memory newElection = Election({
78        description:description,
79        open: true,
80        complete: false,
81        yesCount: 0,
82        noCount:0,
83        blankCount:0
84        });
85
86      elections.push(newElection);
87  }
88
89  // this function is for voting it returns a string value to
        confirm the vote
90  // it takes two inputs index and choice: index for the election
        index & choice
91  // index can take any value, choice can take either value 0, 1 or
        2.
92  function approveCandidate(uint index, uint choice) public returns
        (string) {
93    // 'storage' is for temp memory! 'memory' is for long term
94    // here we make a short cut to our data
95    Election storage election = elections[index];
96
97    // checks if elections are open
98    require((election.open) && (!election.complete));
99    // checks if approver is a voter for the election
100   require(voters[msg.sender]);
101
102   // if approver has already voted he gets kicked out
103   // if not then he's allowed to vote
```

```
104     require(!election.hasVoted[msg.sender]);
105
106     // voter has voted !
107     election.hasVoted[msg.sender] = true;
108
109     // putting the vote in the right place !
110     // returns a string value to make user exit contract
111     // string value should confirm user vote has been counted
112     if (choice == 0) {
113       election.yesCount++;
114       return ("Your vote is counted!");
115     }
116     if (choice == 1) {
117       election.noCount++;
118       return ("Your vote is counted!");
119     }
120     if (choice == 2) {
121       election.blankCount++;
122       return ("Your vote is counted!");
123     }
124   }
125
126   function finalizeElection(uint index) public restricted returns (
          uint yes, uint no, uint blank) {
127     // short cut data
128     Election storage election = elections[index];
129
130     // check if request is not already completed and if it has
            begun
131     require(election.open);
132     require(!election.complete);
133
134     // check mark the election as complete and closed
135     election.complete = true;
136     election.open = false;
137
138     // returns the values !
139     return (election.yesCount, election.noCount, election.
            blankCount);
140   }
141 }
```

References

[1] Darcy Allen, Chris Berg, Aaron Lane, and Jason Potts. The economics of crypto-democracy. 2017.

[2] R Michael Alvarez, Thad E Hall, and Alexander H Trechsel. Internet voting in comparative perspective: the case of estonia. *PS: Political Science & Politics*, 42(3):497–505, 2009.

[3] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. A survey of attacks on ethereum smart contracts (sok). In *Principles of Security and Trust*, pages 164–186. Springer, 2017.

[4] Marcella Atzori. Blockchain technology and decentralized governance: Is the state still necessary? 2015.

[5] S Barta and RP Murphy. Understanding bitcoin: a liberty lover's guide to the mechanics and economics of crypto-currencies, 2014.

[6] Massimo Bartoletti, Salvatore Carta, Tiziana Cimoli, and Roberto Saia. Dissecting ponzi schemes on ethereum: identification, analysis, and impact. *arXiv preprint arXiv:1703.03779*, 2017.

[7] Benjamin B Bederson, Bongshin Lee, Robert M Sherman, Paul S Herrnson, and Richard G Niemi. Electronic voting system usability issues. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 145–152. ACM, 2003.

[8] Chris Berg. *Liberty, Equality and Democracy*. Connor Court Publishing Pty Ltd, 2015.

[9] Stefano Bistarelli, Marco Mantilacci, Paolo Santancini, and Francesco Santini. An end-to-end voting-system based on bitcoin. In *Proceedings of the Symposium on Applied Computing*, pages 1836–1841. ACM, 2017.

[10] Rainer Böhme, Nicolas Christin, Benjamin Edelman, and Tyler Moore. Bitcoin: Economics, technology, and governance. *Journal of Economic Perspectives*, 29(2):213–38, 2015.

[11] Vitalik Buterin et al. Ethereum white paper, 2014. *URL https://github.com/ethereum/wiki/wiki/White-Paper*, 2013.

[12] Vitalik Buterin et al. A next-generation smart contract and decentralized application platform. *white paper*, 2014.

[13] Ting Chen, Xiaoqi Li, Xiapu Luo, and Xiaosong Zhang. Under-optimized smart contracts devour your money. In *Software Analysis, Evolution and Reengineering (SANER), 2017 IEEE 24th International Conference on*, pages 442–446. IEEE, 2017.

[14] Michel Chevallier, Michel Warynski, and Alain Sandoz. Success factors of geneva's e-voting system. *Electronic Journal of e-Government*, 4(2), 2006.

[15] Christopher D Clack, Vikram A Bakshi, and Lee Braine. Smart contract templates: foundations, design landscape and research directions. *arXiv preprint arXiv:1608.00771*, 2016.

[16] Shaen Corbet, Brian Lucey, and Larisa Yarovaya. Datestamping the bitcoin and ethereum bubbles. *Finance Research Letters*, 26:81–88, 2018.

[17] Chris Dannen. *Introducing Ethereum and Solidity*. Springer, 2017.

[18] Sinclair Davidson, Primavera De Filippi, and Jason Potts. Economics of blockchain. 2016.

[19] Simeon Djankov, Edward Glaeser, Rafael La Porta, Florencio Lopez-de Silanes, and Andrei Shleifer. The new comparative economics. *Journal of comparative economics*, 31(4):595–619, 2003.

[20] Saghar Estehghari and Yvo Desmedt. Exploiting the client vulnerabilities in internet e-voting systems: Hacking helios 2.0 as an example. *EVT/-WOTE*, 10:1–9, 2010.

[21] Christopher K Frantz and Mariusz Nowostawski. From institutions to code: Towards automated generation of smart contracts. *IEEE*, 2016.

[22] Stephen N Goggin, Michael D Byrne, and Juan E Gilbert. Post-election auditing: effects of procedure and ballot type on manual counting accuracy, efficiency, and auditor satisfaction and confidence. *Election Law Journal: Rules, Politics, and Policy*, 11(1):36–51, 2012.

[23] Dimitris A Gritzalis. Principles and requirements for a secure e-voting system. *Computers & Security*, 21(6):539–556, 2002.

[24] Ernest J Henley and Hiromitsu Kumamoto. *Reliability engineering and risk assessment*, volume 568. Prentice-Hall Englewood Cliffs (NJ), 1981.

[25] Tarmo Kalvet. Management of technology: The case of e-voting in estonia. In *Computer Technology and Development, 2009. ICCTD'09. International Conference on*, volume 2, pages 512–515. IEEE, 2009.

[26] Gregor E Kennedy and Quintin I Cutts. The association between students' use of an electronic voting system and their learning outcomes. *Journal of Computer Assisted Learning*, 21(4):260–268, 2005.

[27] Tadayoshi Kohno, Adam Stubblefield, Aviel D Rubin, and Dan S Wallach. Analysis of an electronic voting system. In *Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on*, pages 27–40. IEEE, 2004.

[28] Kathrin Konczak and Jérôme Lang. Voting procedures with incomplete preferences. In *Proc. IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, volume 20, 2005.

[29] Andrea Kriskó. Crypto currencies–currencies governed by belief bitcoin, piggycoin, monero, peercoin, ethereum and the rest. In *Conference book Konferenciakötet*, page 283, 2016.

[30] Nir Kshetri and Jeffrey Voas. Blockchain-enabled e-voting. *IEEE Software*, 35(4):95–99, 2018.

[31] Kibin Lee, Joshua I James, Tekachew G Ejeta, and Hyoung J Kim. Electronic voting service using block-chain. *Journal of Digital Forensics, Security and Law*, 11(2):8, 2016.

[32] Iuon-Chang Lin and Tzu-Chun Liao. A survey of blockchain security issues and challenges. *IJ Network Security*, 19(5):653–659, 2017.

[33] Epp Maaten. Towards remote e-voting: Estonian case. *Electronic Voting in Europe-Technology, Law, Politics and Society*, 47:83–100, 2004.

[34] Epp Maaten[1] and Thad Hall. Improving the transparency of remote e-voting: The estonian experience. *Electronic Voting 2008 (EVOTE08)*, page 31, 2008.

[35] Ann Macintosh. Characterizing e-participation in policy-making. In *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*, pages 10–pp. IEEE, 2004.

[36] Ülle Madise and Tarvi Martens. E-voting in estonia 2005. the first practice of country-wide binding internet voting in the world. *Electronic voting*, 86(2006), 2006.

[37] Guiot Marceau. Votingeth. `https://github.com/marceauguiot/VotingEth`, 2018.

[38] Patrick McCorry, Siamak F Shahandashti, and Feng Hao. A smart contract for boardroom voting with maximum voter privacy. In *International Conference on Financial Cryptography and Data Security*, pages 357–375. Springer, 2017.

[39] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Research paper*, 2008.

[40] Svein Ølnes. Beyond bitcoin enabling smart government using blockchain technology. In *International Conference on Electronic Government and the Information Systems Perspective*, pages 253–264. Springer, 2016.

[41] Ryan Osgood. The future of democracy: Blockchain voting'. *COMP116: Information Security*, 2016.

[42] Rachel O'Dwyer. The revolution will (not) be decentralized: Blockchains. *Commons Transition*, 11, 2015.

[43] Marc Pilkington. 11 blockchain technology: principles and applications. *Research handbook on digital transformations*, page 225, 2016.

[44] Reuters. L'ump en plein psychodrame, 2012.

[45] Maarten Simons and Jan Masschelein. Hatred of democracy... and of the public role of education? introduction to the special issue on jacques rancière. *Educational philosophy and theory*, 42(5-6):509–522, 2010.

[46] Grider Stephen. Ethereum casts. `https://github.com/StephenGrider/EthereumCasts`, 2017.

[47] Yu Takabatake, Daisuke Kotani, and Yasuo Okabe. An anonymous distributed electronic voting system using zerocoin. *IEICE*, 2016.

[48] Paul Vigna and Michael J Casey. *Cryptocurrency: How Bitcoin and Cybermoney Are Overturning the World Economic Order*. Random House, 2015.

[49] Peter Namisiko Wanjala. A beginners journey to ethereums smart contracts: Engineering smart contracts and dapps in solidity and web3. js. 2018.

[50] Scott Wolchok, Eric Wustrow, Dawn Isabel, and J Alex Halderman. Attacking the washington, dc internet voting system. In *International Conference on Financial Cryptography and Data Security*, pages 114–128. Springer, 2012.

[51] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.