

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/327812253>

Blockchain-Based E-Voting System

Conference Paper · July 2018

DOI: 10.1109/CLOUD.2018.00151

CITATIONS

266

READS

19,054

4 authors, including:



[Gisli Hjálmtýsson](#)

Reykjavik University

62 PUBLICATIONS 1,873 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



StratusML [View project](#)



Quality of Service [View project](#)

Blockchain-Based E-Voting System

Friðrik Þ. Hjálmarsson, Gunnlaugur K. Hreiðarsson, Mohammad Hamdaqa, Gísli Hjálmtýsson

School of Computer Science

Reykjavik University, Iceland

{fridrik14, gunnlaugur15, mhamdaqa, gisli}@ru.is

Abstract—Building a secure electronic voting system that offers the fairness and privacy of current voting schemes, while providing the transparency and flexibility offered by electronic systems has been a challenge for a long time. In this work-in-progress paper, we evaluate an application of blockchain as a service to implement distributed electronic voting systems. The paper proposes a novel electronic voting system based on blockchain that addresses some of the limitations in existing systems and evaluates some of the popular blockchain frameworks for the purpose of constructing a blockchain-based e-voting system. In particular, we evaluate the potential of distributed ledger technologies through the description of a case study; namely, the process of an election, and the implementation of a blockchain-based application, which improves the security and decreases the cost of hosting a nationwide election.

I. INTRODUCTION

Electronic voting systems have been the subject of active research for decades, with the goal to minimize the cost of running an election, while ensuring the election integrity by fulfilling the security, privacy and compliance requirements [1]. Replacing the traditional pen and paper scheme with a new election system has the potential to limit fraud while making the voting process traceable and verifiable [2].

Blockchain is a distributed, immutable, incontrovertible, public ledger. This new technology has three main features:

- (i) **Immutability:** Any proposed “new block” to the ledger must reference the previous version of the ledger. This creates an immutable chain, which is where the blockchain gets its name from, and prevents tampering with the integrity of the previous entries.
- (ii) **Verifiability:** The ledger is decentralized, replicated and distributed over multiple locations. This ensures high availability (by eliminating a single point of failure) and provides third-party verifiability as all nodes maintain the consensus version of the ledger.
- (iii) **Distributed Consensus:** A distributed consensus protocol to determine who can append the next new transaction to the ledger. A majority of the network nodes must reach a consensus before any new proposed block of entries becomes a permanent part of the ledger.

These features are in part achieved through advanced cryptography, providing a security level greater than any previously known record-keeping system. Blockchain technology is therefore considered by many [3], including us, to have a substantial potential as a tool for implementing a new modern voting process.

This paper evaluates the use of blockchain as a service to implement an electronic voting (e-voting) system. The paper makes the following original contributions: (i) propose a blockchain-based e-voting system that uses “permissioned blockchain”, and (ii) review of existing blockchain frameworks suited for constructing blockchain-based e-voting system.

II. PRELIMINARIES OF E-VOTING AND BLOCKCHAIN

In this section, we first elaborate on the design considerations when constructing an electronic voting system. Then, we provide an overview of blockchain and smart contract technology and its respective feasibility as a service for implementing an e-voting system.

A. Design considerations

After evaluating both existing e-voting systems and the requirements for such systems to be effectively used in a national election, we constructed the following list of requirements for a viable e-voting system:

- (i) An election system should not enable coerced voting.
- (ii) An election system should allow a method of secure authentication via an identity verification service.
- (iii) An election system should not allow traceability from votes to respective voters.
- (iv) An election system should provide transparency, in the form of a verifiable assurance to each voter that their vote was counted, correctly, and without risking the voter’s privacy.
- (v) An election system should prevent any third party from tampering with any vote.
- (vi) An election system should not afford any single entity control over tallying votes and determining the result of an election.
- (vii) An election system should only allow eligible individuals to vote in an election

B. Blockchain as a service

The blockchain is an append-only data structure, where data is stored in a distributed ledger that cannot be tampered with or deleted. This makes the ledger immutable. The blocks are chained in such a way that each block has a hash that is a function of the previous block, and thus by induction the complete prior chain, thereby providing assurance of immutability. There are two different types of blockchains, with different levels of restrictions based on who can read and write blocks.

A *public* blockchain is readable and writeable for everyone in the world. This type is popular for cryptocurrencies. A *private* blockchain sets restrictions on who can read or interact with the blockchain. Private blockchains are also known as being *permissioned*, where access can be granted to specific nodes that may interact with the blockchain [4]. In addition to cryptocurrency, blockchain provides a platform for building distributed and immutable applications or smart contracts.

Smart contracts are programmable contracts that automatically execute when pre-defined conditions are met. Similar to conventional written contracts, smart contracts are used as a legally binding agreement between parties. Smart contracts automate transactions and allow parties to reach agreements directly and automatically, without the need for a middleman. Key benefits of smart contracts compared to conventional written contracts are cost saving, enhanced efficiency and risk reduction. Smart contracts redefine trust, as contracts are visible to all the users of the blockchain and can, therefore, be easily verified. In this work, we define our e-voting system based on smart contracts [5].

III. BLOCKCHAIN AS A SERVICE FOR E-VOTING

This section proposes a new e-voting system based on the identified voting requirements and blockchain as a service. We explain the setup of the blockchain, define the smart contract for e-voting that will be deployed on the blockchain and show how the proposed system satisfies the envisioned voting requirements.

A. Blockchain setup

In order to satisfy the privacy and security requirements for e-voting, and to ensure that the election system should not enable coerced voting, voters will have to vote in a supervised environment. In our work, we setup a Go-Ethereum [7][9] permissioned Proof-of-Authority (POA) blockchain to achieve these goals. POA uses an algorithm that delivers comparatively fast transactions through a consensus mechanism based on identity as a stake. The reason for using Go-Ethereum for the blockchain infrastructure is explained in sub-section C. The structure of the blockchain is illustrated in Figure 1 and mainly consists of two types of nodes.

- (i) **District node:** Represent each voting district. Each district node has a software agent that autonomously interacts with the "bootnode" and manages the life cycle of the smart contract on that node. When the election administrator (see smart contract section) creates an election, a ballot smart contract is distributed and deployed onto its corresponding district node. When the ballot smart contracts are created, each of the corresponding district nodes is given permission to interact with their corresponding contract. When an individual voter casts her vote from her corresponding smart contract, the vote data is verified by the majority of the corresponding district nodes and every vote they agree on is appended onto the blockchain.

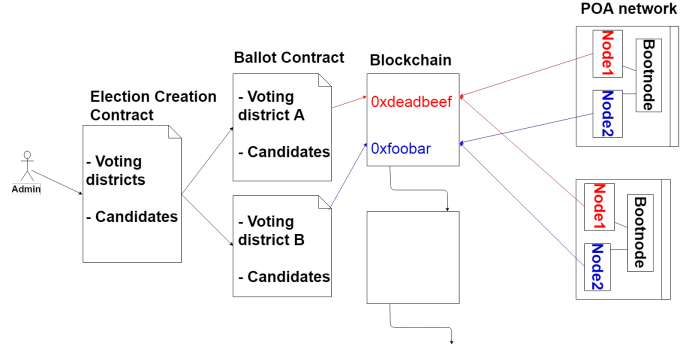


Fig. 1: Election as a smart contract

- (ii) **Bootnode:** Each institution, with permissioned access to the network, host a bootnode. A bootnode is a discovery and coordination service that helps the district nodes to discover each other and communicate. The bootnode does not keep any state of the blockchain and is run on a static IP so that district nodes find their peers faster [6].

After setting up a secure and private blockchain, the next step is to define and deploy a smart contract that represents the e-voting process on the blockchain infrastructure.

B. Election as a smart contract

Defining a smart contract includes three parts: (1) identifying the roles that are involved in the agreement (the election agreement in our case), (2) the agreement process (i.e., election process), and (3) the transactions (i.e., voting transaction) used in the smart contract.

1) *Election roles:* The roles in a smart contract include the parties that need to participate in the agreement. The election process has the following roles:

- (i) **Election administrator:** To manage the lifecycle of an election. Multiple *trusted* institutions and companies may be enrolled in this role. The election administrators create the election, register voters, decide the lifetime of the election and assign permissioned nodes.
- (ii) **Voter:** An individual who is eligible to vote. Voters can authenticate themselves, load election ballots, cast their vote and verify their vote after an election is over.

2) *Election process:* In our work, each election process is represented, by a set of smart contracts, which are deployed on the blockchain by the election administrators as shown in Figure 1. A smart contract is defined for each of the voting districts. The following are the main activities in the election process:

- (i) **Election creation** Election administrators create election ballots using a smart contract in which the administrator defines a list of candidates for each voting district. The smart contracts are then written onto the blockchain, where district nodes gain access to interact with their corresponding smart contract.
- (ii) **Voter registration** The registration of voters phase is conducted by the election administrators. When an election is created the election administrators must define

a deterministic list of eligible voters. This might require a component for a government identity verification service to securely authenticate and authorize eligible individuals. Using such a service is necessary to satisfy the requirement of secure authentication as this is not guaranteed, by default, when using a blockchain infrastructure. In our work, for each eligible voter, a corresponding identity *wallet* would be generated. A unique wallet is generated for each voter for each election that the voter is eligible to participate in.

TABLE I: Example of an transaction in our system

TxHash	Block	To	Value
0xdeadbeef...	1337	N1SC	D
0xG1345edf...	1330	N2SC	P

- (iii) **Tallying results** The tallying of the election is done on the fly in the smart contracts. Each ballot smart contract does their own tally for their corresponding location in its own storage.
- (iv) **Verifying votes** In the voting transaction, each voter receives the transaction ID of his vote. In our e-voting system, voters can use this transaction ID and go to an official election site (or authority) using a blockchain explorer and (after authenticating themselves using their electronic identification) locate the transaction with the corresponding transaction ID on the blockchain. Voters can, therefore, see their votes on the blockchain, and verify that the votes were listed and counted correctly. This type of verification satisfies the transparency requirements while preventing traceability of votes.

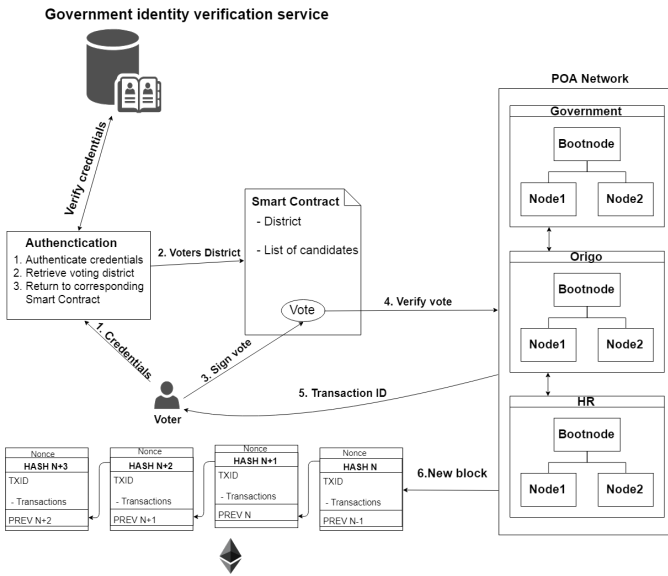


Fig. 2: The voting process

3) **Voting transaction:** Each voter interacts with a ballot smart contract for her corresponding voting district. This smart contract interacts with the blockchain via the corresponding district node, which appends the vote to the blockchain. Each

individual voter receives the transaction ID for their vote for verification purposes. Every vote that is agreed upon, by the majority of the corresponding district nodes, is recorded as a transaction and then appended on the blockchain. Figure 2 is a visual representation of this process. A transaction in our proposed system (see Table I) has information on i) the transaction ID, ii) the block which the transaction is located at, iii) to which smart contract the transaction was sent - which indicates from which voting district the vote was cast, and iv) the value of the transaction, i.e. the vote, indicating which entity (party) the voter voted for. A voting transaction in our system, therefore, reveals no information about the individual voter who cast any particular vote.

C. Evaluating blockchain implementations

As explained at the beginning of this section, in order to satisfy the privacy, security and transparency requirements for e-voting and to ensure that the election system should not enable coerced voting, in our work, we are using a private (permissioned) blockchain for setting up our blockchain infrastructure, where the smart contracts are deployed. In this subsection, we consider three blockchain frameworks (See Table II) for implementing and then deploying our election smart contracts. Those are Exonum, Quorum and Geth.

1) **Exonum:** The Exonum blockchain is robust end-to-end with its full implementation done with the programming language Rust. Exonum is built for *private* blockchains. It has a customized Byzantine algorithm that is used to achieve consensus in the network. Exonum can support up to 5000 transactions per second. Unfortunately, a limitation of the framework is that Rust is the only programming language in the current version, which limits the developers to the constructs available in that language. Exonum is projecting to introduce Java-bindings and platform-independent interface description in the near future to make Exonum more developer-friendly.

2) **Quorum:** An Ethereum-based distributed ledger protocol with transaction/contract privacy and new consensus mechanisms. It is a Geth fork and is updated in line with Geth releases. Quorum has changed the consensus mechanism and is aimed more towards consortium chain based consensus algorithms. Using this consensus allows it to support hundreds of transactions per second.

3) **Geth:** Go-Ethereum or Geth is one of three original implementations of the Ethereum protocol. It runs smart contract applications exactly as programmed without the possibility of downtime, censorship, fraud or third party interference [7][9]. This framework supports development beyond the Geth protocol and is the most developer-friendly framework of those we evaluated. The transaction rate is dependent on whether the blockchain is implemented as a public or private network. Because of these capabilities, Geth was the framework we chose to base our work on, any similar blockchain framework with the same capabilities as Geth could be considered for such systems.

TABLE II: Framework Evaluation

	Exonum	Quorum	Go-Ethereum
Consensus	Custom-built BFT algorithm	QuorumChain, IBFT and Raft-based consensus	PoW, PoS and PoA
Transactions p/s	up to 5000 transactions p/s	Dozens to hundreds	Depends
Private support	Yes	Yes	Yes
Smart Contract Language	Rust	Solidity	Solidity
Programming Language	Rust	Go, C, JavaScript	Go, C, Javascript
Decentralized	Yes	Partially	Optional

Table II shows a summary of the comparison between the three blockchain frameworks.

IV. RELATED WORK

In this section, we present some of the states of the art relevant e-voting systems that use blockchain as a service.

Agora [10] is an end-to-end verifiable blockchain based voting solution designed for governments and institutions. Agora uses their own Token on the blockchain for elections, where governments and institutions purchase these tokens for each individual eligible voter.

A Smart Contract For Boardroom Voting with Maximum Voter Privacy[11], proposed the first implementation of a decentralized and self-tallying internet voting protocol with maximum voter privacy using the Blockchain, called The Open Vote Network (OVN). The OVN is written as a smart contract on the public Ethereum blockchain.

Digital Voting with the use of Blockchain Technology[12] proposed an integration of the blockchain technology to the current voting system in the UK in which the voters can vote at a voting district or on a web browser at home.

Netvote[13] is a decentralized blockchain-based voting network on the Ethereum blockchain. Netvote utilizes decentralized apps (dApps) for the user interface of the system. The Admin dApp allows election administrators to set election policies, create ballots, establish registration rules and open and close voting. The Voter dApp is used by individual voters for registration, voting and can be integrated with other devices (such as biometric readers) for voter identification. The Tally dApp is then used to tally and verify election results.

The main distinctive advantages of our approach over the previous approaches are the following:

(a) Our approach is based private blockchain implementation. Public-based e-voting systems are inefficient concerning financial cost. This inefficiency occurs because of the high gas cost and the gas limits which are set for smart contracts in the network. Another risk using the public blockchain is that high traffic in the network, which could affect the throughput of votes in the system, making it less time efficient. Moreover, a 51% attack poses a threat to a public blockchain, where any individual can sign transactions.

(b) Our approach uses *district-based voting*. Remote voting methods such as those presented, offer no coercion resistance and therefore could lead to false election results. Moreover, web browser voting is one of the largest attack vector for hackers as they could potentially exploit voters through their own devices.

V. CONCLUSION

In this paper, we introduced a blockchain-based electronic voting system that utilizes smart contracts to enable secure and cost-efficient election while guaranteeing voters privacy. We have shown that the blockchain technology offers a new possibility to overcome the limitations and adoption barriers of electronic voting systems which ensures the election security and integrity and lays the ground for transparency. Using an Ethereum private blockchain, it is possible to send hundreds of transactions per second onto the blockchain, utilizing every aspect of the smart contract to ease the load on the blockchain. For countries of greater size, some additional measures would be needed to support greater throughput of transactions per second.

REFERENCES

- [1] Sos.ca.gov. (2007). *Top-to-Bottom Review* | California Secretary of State. Available at: <http://www.sos.ca.gov/elections/voting-systems/oversight/top-bottom-review/>.
- [2] Nicholas Weaver. (2016). *Secure the Vote Today* Available at: <https://www.lawfareblog.com/secure-vote-today>.
- [3] TechCrunch. (2018). *Liquid democracy uses blockchain to fix politics, and now you can vote for it*. Available at: <https://techcrunch.com/2018/02/24/liquid-democracy-uses-blockchain/>.
- [4] Ajit Kulkarni, (2018), "How To Choose Between Public And Permissioned Blockchain For Your Project", Chronicled, 2018.
- [5] "What Are Smart Contracts? A Beginner's Guide to Smart Contracts", Blockgeeks, 2016. Available at: <https://blockgeeks.com/guides/smart-contracts/>.
- [6] Salanfe, *Setup your own private Proof-of-Authority Ethereum network with Geth*, Hacker Noon, 2018. Available at: <https://tinyurl.com/y7g362kd>.
- [7] Geth.ethereum.org. (2018). *Go Ethereum*. Available at: <https://geth.ethereum.org/>.
- [8] Vitalik Buterin. (2015). *Ethereum White Paper* Available at: <https://github.com/ethereum/wiki/wiki/White-Paper>.
- [9] Ethdocs.org. (2018). *What is Ethereum? — Ethereum Homestead 0.1 documentation*. [online] Available at: <http://ethdocs.org/en/latest/introduction/what-is-ethereum.html>.
- [10] Agora (2017). *Agora: Bringing our voting systems into the 21st century* Available at: https://agora.vote/Agora_Whitepaper_v0.1.pdf.
- [11] Patrick McCorry, Siamak F. Shahandashti and Feng Hao. (2017). *A Smart Contract for Boardroom Voting with Maximum Voter Privacy* Available at: <https://eprint.iacr.org/2017/110.pdf>.
- [12] Andrew Barnes, Christopher Brake and Thomas Perry. (2016). *Digital Voting with the use of Blockchain Technology* Available at: <https://www.economist.com/sites/default/files/plymouth.pdf>.
- [13] Jonathan Alexander, Steven Landers and Ben Howerton (2018). *Netvote: A Decentralized Voting Network* Available at: <https://netvote.io/wp-content/uploads/2018/02/Netvote-White-Paper-v7.pdf>.
- [14] Jelurida, "Jelurida", 2017. Available at: <https://www.jelurida.com/sites/default/files/JeluridaWhitepaper.pdf>.