Implementation of a Blockchain-Based Voting System Using Hyperledger Fabric

Mortimer Sotom, David Guest, Emil Barbuta, Guilherme Barreiro Vieira

I. Introduction

Voting is an important part of any democratic society. Even with all the advances of modern technology, the vast majority of countries use paper based voting to conduct their elections. The fact that the voter turnout has decreased in recent years seems to show that this is not the most effective procedure for voting [1]. One could argue that it's time to bring voting into the 21st century, and that online voting is the best way to achieve this. However, for the moment, online voting has not been widely adopted as the perceived lack of security is an important concern. As highlighted in [2], attacks such as placing malware on client machines that can alter votes is just one example of vulnerability. Fortunately, the blockchain technology might provide an innovative approach to tackle this problem. Originally designed to offer a secure system for cryptocurrency transactions, blockchain has been shown to be capable of creating a reliable environment for other application domains as well. This project intends to show that using such a decentralised system of nodes responsible for validating transactions in the whole network can be a viable solution for a safe and robust online voting system.

II. Decentralised Voting - Motive

A. Current System

The current voting system in the United Kingdom works as follows, with three distinct phases:

1) Registration: Voters must join the electoral register to be eligible to vote. To do this online, the simplest method, they require a national insurance number or UK passport. Whenever a voter moves constituency they must update their details on the electoral register to be able to vote in elections affecting their new constituency.

2) Voting: When an election is called (every three to five years for general elections and on a four yearly cycle for local elections), registered voters are mailed poll cards by the Electoral Commission, directing them to their nearest polling station. On the day of the vote, they must present themselves at a polling station, and identify themselves to a registrar who ticks their name off a list of eligible voters in that district. In contrast to countries such as the United States, ID is not required. The voter is given a polling card, with the names of all local candidates, which they must mark and post into the ballot box.

3) Counting: Following the close of polling, ballot boxes for each constituency are delivered to a central hub for counting. There are 650 constituencies across the UK, and in each of these, an army of volunteers will work throughout the night to count, check and cross-check the vote counts for each candidate. When checks have been carried out, the returning officer will announce the count of votes to all candidates and their supporters. In close elections, there will often be a recount of ballots, complete with accompanying discussions around which ballots are considered “spoiled" and should be thrown out.

B. Issues with the Current System

The process has changed very little in the past century, and, while functional, has a number of issues that make it impractical to run votes more often, a demand often made by those who feel technology should allow countries to transition from representative to direct democracies. The issues below are just a sample of these:

1) Monetary cost: Elections run in this manner are very resource intensive, requiring large numbers of paid and volunteer workers to preside over the polling stations and vote counts. The Electoral Commission stated in 2012, that a General Election cost over £100 million to run [3].

2) Time cost and travelling requirements: Additionally, there is the time cost to consider. Voting can be expected to take a minimum of 15 minutes per physical voter, which with a typical electoral turnout of around 30 million, adds up to over 2,500 working years of voters time spent on a single election! Voters in rural areas can expect to spend even longer than this and there may be issues for older voters in these areas even reaching the polls.

3) Lack of transparency in count: There are several occasions during the process where the possibility of vote tampering or miscounting could arise. Although elections in the UK are generally regarded as fair, this does not apply in every country where similar voting systems are used, as the public are unable to verify the number of votes cast in each constituency.

4) Issues with clarity of vote: Even with a pencil, paper, and clear instructions, the intentions of voters are sometimes unclear when it comes to the ballot count, and there have been past controversies around when ballots should be accepted and when they should not be counted (“hanging chads” in Florida being the most famous example [4]).

5) Lack of demographic vote information: While being able to identify the votes of individuals is not necessarily desirable, being able to track the voting patterns of different demographic groups can lead to greater understanding of their priorities - in present systems this can only be done through the use of polls, which are not always accurate.

C. E-voting

Most e-voting solutions will provide partial solutions to the above issues, particularly those around monetary and time costs. However, these systems can introduce problems of their own. According to [5] there are six most important requirements of an an e-voting system. These are paraphrased below:

1) Eligible voters may cast votes which are correctly counted.
2) Non-eligible voters are disenfranchised.
3) Eligible voters may only cast a single ballot.
4) Voting is private and voters cannot be coerced to change their vote for financial or other incentives.
5) It is possible for auditors to check that the final tally has been correctly computed.
6) The results of the voting must be secret until the election period has ended.

These requirements have proven tricky to fully integrate into e-voting systems, and often as a result there is no increase in transparency.

D. Blockchain Voting

Introduction of the E-voting concept brings additional concerns in regards to security, as it becomes more susceptible to hacking and potential manipulation of elections. Savvy hackers could tamper with the way votes are submitted or counted, cast votes for people who didn't intend to vote and even coerce voters into voting in a certain way. It can also be difficult to prove the identity of the people casting the vote as well as preventing the double-spending of votes [6]. Blockchain technologies have the potential to provide solutions to these problems. Using blockchain technology for electoral voting is a relatively new concept and has been implemented for the first time in Sierra Leone, using technology from Agora, a Swiss Lab & Foundation offering decentralized digital voting systems based on blockchain technology [7][8]. There is a lack of transparency in many elections around the world and African countries have suffered from the manipulations of electoral results, causing a lack of trust from the people in the current voting system. The blockchain voting system trial was seen as a way to restore some of that lost trust. Blockchain voting technology brings some desirable characteristics that are necessary in an effective electoral system, and which can help reduce some of the issues outlined above:

- Trust - A citizen can theoretically perform the same actions as under the current voting system, without trusting a 3rd party electoral organiser, due to the trustless nature of the consensus mechanism used in blockchain technologies.
- Auditability - submitted votes cannot be changed due to the immutability characteristic of the blockchain, therefore, every vote can indisputably be traced to the date and time it was submitted.
- Security - The distributed nature of blockchain technologies gives rise to an increased security as every transaction is verified by other nodes in the network. Additionally, a fraction of the system can be attacked without being able to compromise all nodes of the system and its functionality, due to there being no central point of failure.
- Transparency - The trustless aspect, open source, security and the auditability of blockchain systems gives rise to an increased sense of transparency in the election process. This is because at the end of the election, the entirety of the votes cast can be audited to confirm their authenticity. Finally, there is a lower uncertainty about identity of the voters as non-citizens would be rejected by the system.
- Privacy - The use of asymmetric cryptographic keys provides a sense of identity protection and makes it difficult to trace users back to individuals. However, there are reports of using graph analysis to match public keys to individual identites in network such as Bitcoin, indicating this sense of privacy is not fully warranted [9]. Permissioned blockchains provides some response to this, as the access of each participant is well defined and differentiated based on role.

Finally, there are significant monetary cost improvements using a blockchain as the administration costs are greatly reduced, and the travelling requirements are minimalised by introducing the opportunity of voting remotely over the internet.

III. Project Implementation

A. Why Hyperledger Fabric

The first stage was to identify which Distributed Ledger Technology (DLT) was more appropriate for the scope of the project. The two leading platforms in this field are Ethereum and Hyperledger. Ethereum is open-sourced and aimed at developing decentralised applications [10]. It implements the functionality of smart contracts to automate transations or agreements with a proof of work (PoW) protocol. All smart contracts are executed through the Ethereum Virtual Machine so that every single operation is performed by all nodes in the network. Each operation requires a different level of computational power which is measured in gas and paid for with the Ether token. Furthermore, Ethereum maintains total transparency and is therefore permissionless by default. It can also be implemented in a private network, although this was not desirable as all participants have the same permission across the network. It has become a well established platform with many open sourced projects and available documentation within the community.

Hyperledger (HL) was founded by the Linux Foundation and refers to a collection of open-sourced projects with various frameworks contributed by companies such as IMB (HL Fabric) or Intel (HL Sawtooth Lake). HL Fabric is as recent as July 2017 and is consistently being updated by a large team of developers. This does mean the documentation is not always up to date but IBM offers tutorials, examples and 'playground', a testing platform for developers. HL Fabric also supports smart contracts which are validated by 'Peer nodes', a concept explained in more detail in section 5b of this report, by using the Practical Byzantine Fault Tolerance (PBFT) algorithm to reach a consensus. Additionally, it is a permissioned blockchain which provides a modular architecture and configurable services.

HL Fabric was therefore selected as the project platform due to the following properties:

- Permission blochain network
- No cryptocurrency or token involved
- No mining involved with the PBFT method
- Playground: testing platform with user interface for demo
- Not possible to keep cast votes secret with Ethereum architecture
- Public nature of votes cast will allow vote selling with Etherum

B. Project Model

This project is based on three fundamental layers. Hyperledger works as the incubator for the blockchain technology, Fabric provides the framework for the architecture layer and the Composer toolset is related to the application layer to program the business network. The business network is the core of the voting system and its architecture is illustrated in figure 1. The model file (.cto) describes the list of resources used, the script file (.js) defines the smart contracts which are referred to as transactions, and the access control file (.acl) determines the read/write/delete operator permissions for each participant or user. These three files are then combined through a command into a business network archive file (.bna) in order to deploy the project.

The model file defines the resources by attributing variables, a type and any relationship to the resource. A resource can have one of three types: an asset which refers to a physical object, a participant represents an individual or organisation, and a transaction for the smart contract. All resources are identified by an ID or a specified variable name. The first resource, *user*, is of type participant and is identified by a *userID* obtained after completing the registration process. A regular expression, or regex, is set up to ensure the correct format is inputted when a *user* is referred to. The participant *voter* is an extension of *user* and therefore inherits the variable *userID*. Additionally a *voter* has a boolean *hasVoted* indicating whether they have already voted or not to ensure each *voter* can only vote once. Similarly, a participant *candidate* is an extension of *user* with extra variables for the candidate's name and political party. The only asset in the project model is *ballot*, identified by a *ballotID*. The *ballot* is attributed to a *voter* and also has a boolean *used* to ensure a *ballot* can only be used once. The final resource *vote* is the transaction itself. A transaction requires a *ballot* which is linked to a *voter*, as well as a *candidate* to determine who the user is voting for.

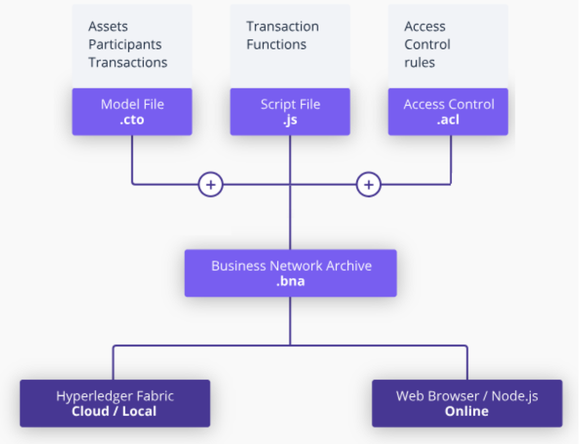

Fig. 1. Business Network Architecture [11]

The script file contains the code detailing what happens during a transaction *vote*. The boolean of the *voter* and *ballot* and the owner of the *ballot* are first checked to ensure the user is voting for the first time with their own ballot, and that their ballot hasn't been used. This provides a double safety measure to prevent double voting. The *voter* loses ownership of the *ballot*, which is attributed to the *candidate* specified by the *voter*. The *voter* and *ballot* information is updated to *hasVoted* and *used*. The resource *candidate* has an extra variable *amountOfVotes* solely to prove the concept of this project. This variable is incremented by one each time a *voter* votes for that *candidate*.

Finally, the access control file sets the restrictions and permissions of all participants. Any *user* has the permission to read *candidate* information. *Voters* are given the right to create transactions while *candidates* are not. For the sake of transparency, any *user* can read the information of *voters*, which reveals only the *userID*, and whether that user has already voted or not. The *userID* cannot be linked to the real person since this information is not included in the blockchain. No *user* is given the right to update or delete any information whatsoever. Network administrators would have the permission to read any information through the peer node structure. In the source code presented, administrators have full rights (create, read, update and delete) to allow the showcasing of the full capabilities of the model.

IV. A REAL WORLD IMPLEMENTATION SCENARIO

In order to understand potential uses of the blockchain voting application, the below scenario shows how the application could be integrated into the voting process. It draws heavily on the current system outlined in the Motives section of this report. Figure 2 also shows the process flow of how this scenario would work.

Scenario: Voting with the decentralised e-voting application

Registration: This phase of the voting process will be the most similar to the current system. A trusted third party will still be required to take the role of the Electoral Commission and oversee the registration process. Voters will still be required to register for their constituency votes, with the only difference being that user IDs for the voting application will be sent out, rather than polling cards as in the current system.

Voting: The voting process itself will be vastly different. Registered voters will use the provided IDs to log into the application, where they will have the ability to create a transaction on each of the channels on which they are eligible to vote. This transaction will allow them a choice of available candidates for the channel, or the option “None of the above" for those wishing to register a process vote. Upon submission, this transaction will alter the ownership of the associated ballot to the chosen candidate. Once ownership of the ballot is changed, it cannot be reverted - similar to the finality of dropping the vote into a ballot box in the current system.

Counting: This phase will benefit the most from the increased transparency afforded by the voting process. Immediately upon the end of the voting period, an event will be triggered granting read-only access across all channels to all participants. All participants will be able to scrutinise the votes cast in each channel, or constituency, for discrepancies. Shared queries will allow the participants and electoral authorities to immediately understand who won, and, in elections where demographic information is kept, the sections of the electorate that backed the winners.

In terms of practical considerations, the Hyperledger Fabric architecture allows a throughput of 3,500 transactions per second with latency of a second [12]. With over 12 million transactions an hour, even allowing for likely spikes in evening votes, this should be sufficient for an electorate of up to 60 million - this is in excess of the recent UK turnout figures, and would be sufficient for the size of national elections in nearly every democracy. Larger electorates may require a different Hyperledger cluster for each state or region, but this could still be achieved at a considerably lower cost than the current voting system.

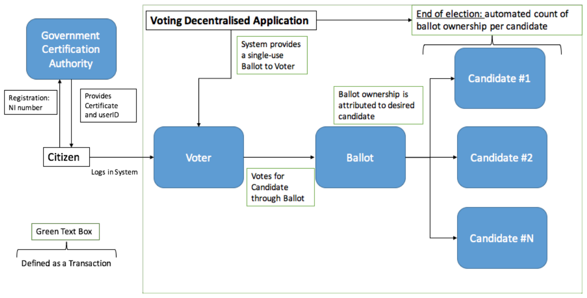
Government Certification Authority
Registration: NI number
Provides Certificate and userID
Citizen
Logs in System
Green Text Box
Defined as a Transaction
Voting Decentralised Application
System provides a single-use Ballot to Voter
Voter
Votes for Candidate through Ballot
Ballot
Ballot ownership is attributed to desired candidate
End of election: automated count of ballot ownership per candidate
Candidate #1
Candidate #2
Candidate #N

Fig. 2. Process flow for an election using the blockchain voting application

V. Extensions and Limitations

A. Extensions

The voting application that has been developed is just a prototype with the intention of showing the potential of blockchains in the context of e-voting. It is by no means a fully-functional product, and could benefit from any or all of these further extensions::

- *Vote counting* - the application performs vote counting by keeping a local variable for each candidate and increasing it when a ballot is transacted to the candidate. Ideally the application should have an automatic feature that counts all ballots for each candidate when the voting deadline is reached.
- *Network reveal* - in order to ensure trust and prove that all voting transactions are legitimate, the blockchain application should publish the entire network when the deadline for elections is met. This way, every user is granted full reading rights and can access all of the transactions made.
- *User Interface* - the applications currently lacks a front-end. Since the target users are non-technical individuals, developing a user-friendly web application to support its usage is a must in order for the product to be deployed and used in a professional environment. For instance the current version requires voters to refer to candidates by a unique user ID, but the UI might provide an abstraction so that voters could select candidates by their name and party.
- *Ballot allocation* - whenever a user registers and is considered eligible for voting they should automatically receive a unique voting ballot that can be used only once for voting. The application does not currently support an automated process for this.

B. Limitations

The Hyperledger Fabric architecture contains the concept of “peer nodes” which have exclusive rights in reading the system for monitoring purposes as well as storing a whole copy of the ledger - these nodes can be considered analogous to network administrators. Hence, peer nodes need to be carefully chosen to avoid any coercion from their part, as conflicts of interest are a problem. By increasing the amount of people who have read capability access to the network, one is increasing the decentralisation of the system. However, at the same time, there is a corresponding increase in the risk of information being leaked by the peer nodes. An entity with an interest in knowing the current state of the election could elect to offer a monetary reward to one of the responsible peer nodes, in exchange for the current vote count. Hence, one could argue that peer nodes should be both anonymous and scarce, with the downside of increasing centralised control over the system. Ideally these individuals would be unbiased and incorruptible, however, in real life this would be next to impossible to guarantee. There are some academic proposals in how to deal with the allocation of these participants[13][14], however, it may be best to have an outside entity with sufficient technical expertise employed to monitor the system and keep the network up. This was the model used in the Agora example, which was run by an outside Swiss organisation[8]. Another possible implementation in order to avoid the readability of the content of votes by the network administrators, would be the use of blind signatures in order to encode the content of the ballots [14]. In contrast, making everything public is a possibility, however, this would defeat the point of using a permissioned blockchain, as a public blockchain such as Ethereum would be more appropriate. A public blockchain would add different dynamics to the elections, as the entire population would know with certainty the exact status of the election, which would dramatically change the information available to voters about how they could cast their vote to best realise their preferences.

Furthermore, one could argue that there is a possibility that user ID details could be stolen, misdirected or lost in transit which would cause the reliability of the vote to be questioned. To combat this possibility, a requirement for a biometric measure alongside the user ID would go a long way towards guaranteeing the integrity of the vote, although voters could still be coerced or forced to vote in a certain way.

There are clear flaws that can be identified in an e-voting system and there are many trade-offs to be contemplated when developing such a system. Elections are complex, and guaranteeing their legitimacy and effectiveness while considering every possible attack is difficult.

VI. Conclusion

A reliable and truthful voting system is crucial for any democratic society. Electoral voting systems have barely changed in the last century and most still implement the paper ballot voting method, with some exceptions (e.g. Estonia). However, this current system has a number of downsides including monetary costs, lack of transparency in vote counting and a slow decline in the percentage of citizens voting. The implementation of a voting system with blockchain technology has the potential to solve many of those issues while maintaining the characteristics of transparency, security, auditability and privacy leading to trust. A proof of concept of this project was built using Hyperledger Fabric and the Composer tools. Deploying this type of project for real world scenarios requires a number of aspects to be carefully thought through and planned in advance. A registration process must be put in place for citizens to prove their identity and obtain a userID to gain access to the private blockchain network. Peer nodes are critical for the decentralisation of the system but introduce a weakness into the system as information could be leaked. There are a number of other limitations but this project could have the potential to revolutionise electoral voting systems if implemented correctly.

References

[1] M. Montgomery, “One place where blockchain could really help: Voting.” https://tinyurl.com/yd6cjlsx. Accessed: 2018-05-13.

[2] A. Barnes, C. Brake, and T. Perry, “Digital voting with the use of blockchain technology.” https://www.economist.com/sites/default/files/plymouth.pdf. Accessed: 2018-05-13.

[3] The Electoral Commission, “The cost of electoral administration in great britain.” https://tinyurl.com/ya3agnar. Accessed: 2018-05-12.

[4] A. Agresti and B. Presnell, “Misvotes, undervotes and overvotes: The 2000 presidential election in florida,” *Statist. Sci.*, vol. 17, pp. 436–440, 11 2002.

[5] A. Buldas and T. Mägi, “Practical security analysis of e-voting systems,” in *Proceedings of the Security 2Nd International Conference on Advances in Information and Computer Security*, IWSEC’07, (Berlin, Heidelberg), pp. 320–335, Springer-Verlag, 2007.

[6] BBC news, “E-voting experiments end in norway amid security fears.” http://www.bbc.co.uk/news/technology-28055678. Accessed: 2018-05-13.

[7] M. del Castillo, “Sierra leone secretly holds first bvlockchain-audited presidential vote.” https://tinyurl.com/y7atw63w, 03 2018. Accessed: 2018-05-13.

[8] Y. Kazeem, “The world’s first blockchain-supported elections just happened in sierra leone.” https://tinyurl.com/y7atw63w, 03 2018. Accessed: 2018-05-13.

[9] J. Barcelo, “User privacy in the public bitcoin blockchain,” 2007.

[10] B. Vitalik, “A next generation smart contract and decentralised application platform - etherium white paper.” https://github.com/ethereum/wiki/wiki/White-Paper, 2018. Accessed: 2018-04-25.

[11] Y. Kazeem, “Hyperledger composer documentation.” https://hyperledger.github.io/composer/latest/introduction/introduction.html, 2018. Accessed: 2018-04-20.

[12] M. Vukolić, “Behind the architecture of hyperledger fabric.” https://www.ibm.com/blogs/research/2018/02/architecture-hyperledger-fabric/. Accessed: 2018-05-13.

[13] R. Osgood, “The future of democracy: Blockchain voting,” 2016.

[14] M. Kucharczyk, “Blind signatures in electronic voting systems,” *International Conference on Computer Networks*, pp. 349 – 358, 2010.