Implementation of Auditable Blockchain Voting System with Hyperledger Fabric

Michał Pawlak[1] and Aneta Poniszewska-Marańda[1][0000−0001−7596−0813]

Institute of Information Technology, Lodz University of Technology, Lodz, Poland
michal.pawlak@p.lodz.pl, aneta.poniszewska-maranda@p.lodz.pl

Abstract. An efficient democratic process requires a quick, fair and fraud-free election process. Many electronic-based voting systems have been developed to fulfil these requirements but there are still unsolved issues with transparency, privacy and data integrity. The development of distributed ledger technology called blockchain creates the potential to solve these issues. This technology's rapid advancement resulted in numerous implementations, one of which is Hyperledger Fabric, a secure enterprise permissioned blockchain platform. In this paper, the implementation of an Auditable Blockchain Voting System in Hyperledger Fabric is presented to showcase how various platform components can be used to facilitate electronic voting and improve the election process.

Keywords: E-voting · Blockchain · Hyperledger Fabric · Auditable Blockchain Voting System.

1 Introduction

An efficient and honest democratic process requires a fair and fraud-free election process [1]. For that reason, the voting process is secured by complex measures. However, that is not enough and elections are still vulnerable to various attacks. What is more, the most common traditional paper-based systems are slow and manipulation-prone. All this undermines trust in such systems and reduces participation in the main democratic process.

There is ongoing research aiming to solve this problem, which resulted in many different solutions, including electronic-based ones [2]. These electronic voting systems provide many advantages, such as quick result calculation, improved ballot presentation, reduced costs and convenient usage due to the possibility of unsupervised voting through a network. However, these systems face many challenges and issues [3]. The most prominent of which is lack of transparency, difficulties with voters' authorization and authentication, and enforcement of data integrity and privacy [4].

Many of these problems may be solved with the application of blockchain technology, which is distributed in a peer-to-peer network system of ledgers. Peers cooperate on validation and management of stored data. This technology enables the creation of anonymous, transparent, manipulation-resistant systems. Since its introduction in 2008, there have been many new and more advanced

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

implementations of blockchain technology. One of them is Hyperledger Fabric, which is an open-source customizable blockchain solution designed for enterprise use. For that reason, it provides complex permission and policy management. Furthermore, Hyperledger Fabric supports smart contracts in popular programming languages like Java, Go and Node.js, which allow for quick development of advanced business logic on top of the blockchain [5].

This paper intends to present an implementation of Auditable Blockchain Voting System (ABVS) concepts in Hyperledger Fabric. The main focus is on how the e-voting blockchain network can be organized in Hyperledger Fabric and how various Hyperledger Fabric components can be used to facilitate and secure electronic voting.

The remaining parts of this paper are organized as follows. Section 2 presents the theoretical background of electronic voting and presents a technical description of Hyperledger Fabric. Section 3 provides an overview of works related to the field of electronic voting and blockchain. In Section 4, the implementation of the Auditable Blockchain Voting System in Hyperledger Fabric is detailed. Section 5 analyses voting properties concerning the presented system while Section 6 presents conclusions.

2 Electronic voting and Blockchain in Hyperledger Fabric

The Council of Europe defines an electronic voting system or e-voting systems as any type of election or referendum, which utilizes electronic means to at least facilitate vote casting [6]. Thus, the term covers a wide variety of different solutions and implementations. In general, electronic voting systems can be classified concerning two characteristics, namely: *supervision* and *remoteness* [7–9]. The classification is presented in figure 1.

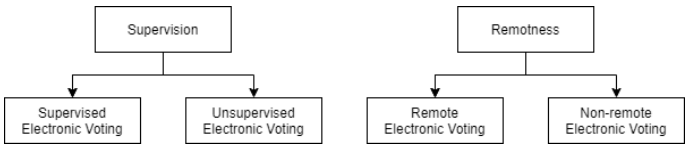

Fig. 1. Classification of electronic voting systems

Supervision divides electronic voting systems depending on the circumstances of voting. In *supervised* systems, elections and referendums are conducted from polling stations under official supervision. This type of e-voting is secure against coercion and vote selling, but it is inconvenient, time-consuming and expensive. On the other hand, *unsupervised* systems allow voting to be conducted from any location as long as voters can access the required facilities to transfer their votes. This type of e-voting is fast and convenient but vulnerable to vote selling, coercion and manipulation.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

Remoteness categorizes electronic voting systems concerning the method of transporting cast ballots for counting. *Remote* systems instantly transfer cast ballots to a remote location via a chosen communication channel, e.g. the Internet. These systems provide fast results and reduced overhead but are vulnerable to attacks on the communication channel and data manipulation during transfer. Non-remote systems store cast ballots locally and provide local results combined to provide a final result after an election ends. These systems are secure against attacks during transfer but are prone to local manipulations and errors during counting.

Electronic voting systems must aim to satisfy the following criteria to be considered safe and secure [7, 10, 11]: eligibility, privacy, correctness/completeness, fairness, transparency, verifiability, availability.

Eligibility describes a requirement of allowing only authenticated and authorized voters to cast their votes. *Privacy* ensures that voters cannot be linked to their votes, so only the voters themselves know how they voted. *Correctness/Completeness* requires that only valid votes are counted and a given electoral law is enforced. *Fairness* ensures that e-voting systems do not in any way influence election results. *Transparency* is a requirement that all procedures and components are clear and understandable to voters. *Verifiability* requires that e-voting systems can be verified against their requirements and safety criteria. Finally, *availability* describes a requirement that e-voting systems should allow all eligible voters to vote and not prevent anyone from participating in elections.

There exist many different electronic voting systems and each has its own advantages and disadvantages. However, electronic voting systems aim to provide the following advantages: (i) reduction and prevention of frauds by minimising human involvement; (ii) acceleration of result processing; (iii) improvement of ballot readability to reduce the number of spilled ballots; (iv) reduction of costs by minimising overhead.

Unfortunately, electronic voting systems must also face many technical, procedural and legislative challenges. The most important one is a lack of trust, which is a result of: (i) inadequate transparency and understanding of electronic voting systems by common voters; (ii) lack of widely accepted standards against which e-voting systems can be verified; (iii) vulnerability to attacks by privileged insiders and system providers; (iv) costs of implementation and infrastructure.

There is still ongoing research on securing voting systems or solving some (or all) challenges of electronic voting systems. This paper aims to provide a solution to verifiability and transparency issues with the use of blockchain technology implemented in Hyperledger Fabric, which is described in the following section.

Blockchain technology consists of a distributed system of ledgers stored in a chain-like data structure of connected blocks and a consensus algorithm that collectively validates the contents of the blocks in a peer-to-peer network. In general, blockchains can be divided into [8, 12]:

- public and private,
- permissionless and permissioned.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

Public blockchains allow everyone to join a blockchain network and read data contained within, for example, Bitcoin and Ethereum. In contrast, *private* blockchains allow only a selected group of entities to access blockchain data, for instance, MultiChain and Hyperledger Fabric. On the other hand, *permissionless* blockchains allow anyone to join and participate in a consensus algorithm, while *permissioned* blockchains divide participants concerning their permissions, for example, MultiChain and Hyperledger Fabric.

There are numerous blockchain implementations, one of which is already mentioned Hyperledger Fabric developed as an open-source project by the Linux Foundation, which is an esteemed developer community. The Hyperledger Fabric is a part of a whole family of solutions and tools, all designed with the following principles in mind [13]: modularity and extensibility, interoperability, security, token agnosticism, rich and easy-to-use APIs.

Hyperledger Fabric fulfils these principles by the implementation of the following design features, which are its main components [5, 13]: assets, ledger, smart contracts and chaincode, consensus, privacy components, security and membership services.

Assets are represented by a collection of key-value pairs and enable to exchange of valuables over a Hyperledger Fabric blockchain network. Assets form the main business objects that are stored in an immutable *ledger* made of connected blocks. In essence, assets represent facts about a system and a ledger stores current and historical states of them.

However, every system needs some kind of business logic. In the case of Hyperledger Fabric-based systems, it is provided by *smart contracts*. They are executable programs, which define common terms, data, rules, concept definitions and processes involved in the interaction between involved parties. In order to utilize these programs, they must be packed and deployed on a blockchain network. In the context of Hyperledger Fabric, packed and deployed smart contracts are called a *chaincode*. Chaincode can interacts with a ledger primarily with *put*, *get*, *delete* and query operations. Furthermore, chaincode provides a rich API that provides methods for interaction between chaincodes, chaincode events and client-related requests. Every such transaction is recorded in a ledger. It is important to note that chaincode is stored on blockchain nodes, called peers, and can only be executed by them when explicitly installed.

However, before a transaction can be committed to a block and a ledger, it must first be endorsed. How this process is conducted depends on an endorsement policy, which describes which members (*organisations*) of a network must approve generated transactions. A model of a smart contract with a connected application is presented in figure 2. The model presents a smart contract for car exchange, which allows querying for a car, updating car properties and transferring car ownership between organizations. Furthermore, it has an associated policy, requiring both organizations to approve a transaction before it can be committed.

Because of the distributed nature of a blockchain network, participating nodes must agree on a common state of the network and contained within data

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

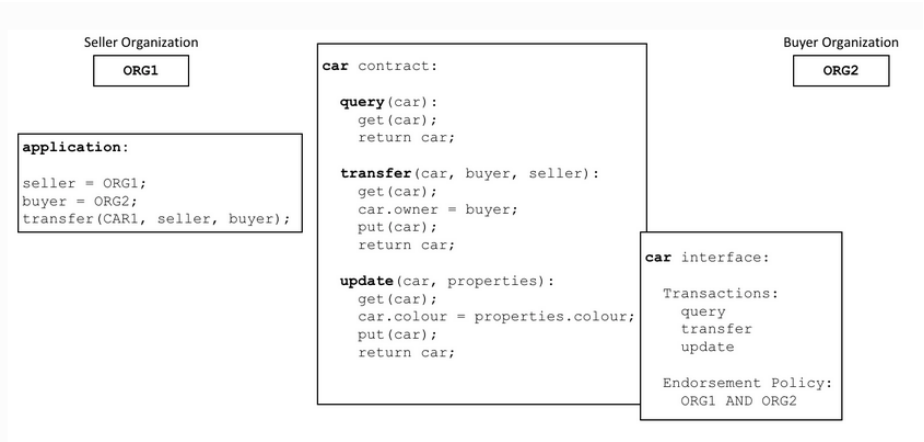

Fig. 2. Model of a smart contract, adapted from [5]

to achieve a *consensus*. To achieve that, Hyperledger Fabric's network is made of two types of nodes. The first type, already mentioned, consists of *peers*, which store copies of ledgers and validate transactions by creating *transaction proposals* (executed transaction without updating the network's state). Valid transactions are then passed to *orderer nodes*, which elect from themselves a leader, which is allowed to modify the network's state by creating and committing new blocks. The leader is selected via the Raft algorithm [14]. It is important to note that these roles are not exclusive and a single node can fulfil multiple roles.

To provide privacy and separation in a blockchain network, Hyperledger Farbic allows creating *consortiums* and *channel*. A *consortium* is a set of organizations which intend to cooperate with each other. A *channel* is a separated communication mechanism that allows members of a consortium to freely communicate with each other in separation from other involved organizations, which can have their own consortiums and channels in the same network.

Security is provided by issuing, via Public Key cryptography, cryptographic certificates to organizations, network components and client applications. As a result, each entity can be identified, and its rights and privileges within the system can be managed to a significant degree. When combined with separation provided by consortiums and channels, Hyperledger Fabric delivers an environment suitable for private and confidential operations. In the context of security, the *Membership Service Provider* (MSP) must be mentioned. MSP is a component abstracting membership operations and provides mechanisms for issuing certificates, validating certificates and user authentication.

Figure 3 presents a model of the described components. As can be seen, there are two peer nodes and one orderer node connected to a single channel. Each node stores its own copy of the channel's ledger and a copy of the chaincode. The whole network interacts with an application that triggers smart contracts stored in the chaincode.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

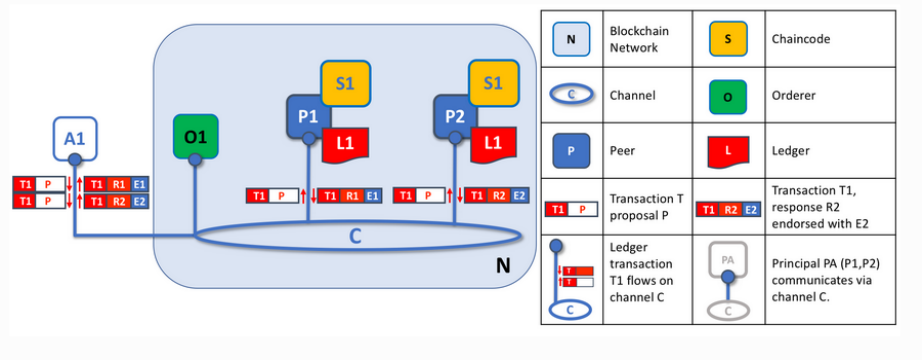

Fig. 3. A model of a single Hyperledger Fabric channel, adapted form [5]

Hyperledger Fabric components are highly configurable and modular, which allows a high degree of customisation [5, 13]. This makes this solution ideal for many applications, which require security and privacy. Moreover, for these reasons, it was chosen as a platform for an e-voting solution.

3 Related works

There are numerous publications regarding blockchain-based electronic voting. In [15] presents an overview of electronic voting systems created with blockchain technology. The authors review multiple research articles and describe presented in the systems, along with their advantages and disadvantages.

The authors of [16] describe an implementation of the electronic voting system in Hyperledger Fabric-based on an approach presented in [17]. The described approach allows conducting voting traditionally with paper and electronically with software. In addition to Hyperledger Fabric, the system utilizes blind signatures, secret sharing schemes and identity mixer to provide security and privacy.

[8] presents results of investigation of scalability and performance scalability constraints of e-voting systems based on the blockchain technology. The authors conducted experiments concerning voting population, block size, block generation rate and transaction speed on both permissionless and permissioned blockchain, namely, Bitcoin and MultiChain.

The author of [7] demonstrates a model of the blockchain-based electronic voting system that intends to provide coercion resistance, receipt-freeness and universal verifiability using zero-knowledge Succinct Non-iterative Arguments of Knowledge and Bitcoin blockchain implementation.

Chaintegrity introduced in [18] is intended to achieve scalability, verifiability and robustness in large scale elections. The main used components are smart contracts, homomorphic and Paillier threshold encryptions, and a counting Bloom filter and Merkle hash tree. Furthermore, the authors provide extensive documentation of conducted testing, including performance evaluation.

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

There exist some commercial blockchain-based electronic voting systems. One of the most prominent is Agora [19]. It is a Swiss-made e-voting system based on a custom blockchain implementation. It consists of five components/layers: Bulletin Board blockchain functioning as a communication channel, Cotena logging mechanism, Bitcoin blockchain for recording transactions in ledgers, Valeda global network for validating election results, Votapps application layer for interaction with the Bulleting Board.

An example of a non-blockchain e-voting system is Hellios, an open audit, remote and unsupervised e-voting system written in JavaScript and Python Django framework. Helios is a web-application following centralized client-server architecture. The system utilizes Sako-Kilian mixnet to provide anonymity and to prove correctness.

However, the woks presented above possess the drawbacks for an electronic voting system targeting low-coercion risk, Internet based elections. Not only the system has to be auditable, safe and transparent, but also minimize the work voters have to perform to keep the system integral. The traits can be achieved by decentralizing the system through blockchain usage.

The proposed approach presents an implementation of Auditable Blockchain Voting System concepts in Hyperledger Fabric together with e-voting blockchain network organized in Hyperledger Fabric and various Hyperledger Fabric components to facilitate and secure electronic voting.

4 Auditable Blockchain Voting System with Hyperledger Fabric

This section presents the Auditable Blockchain Voting system implementation in Hyperledger Fabric. The system was designed as a remote and supervised voting system. However, it is possible to use it in an unsupervised environment. ABVS is intended to improve the existing voting system in Poland. The following subsections will present various aspects of ABVS in separate subsections.

4.1 Auditable Blockchain Voting System overview

Details and an initial idea behind Auditable Blockchain Voting System can be found in [4], but in general a process utilized in ABVS can be divided into three phases (Fig. 4).

In the *election setup* phase, a set of trusted public and private organizations is defined by the National Electoral Commission (NEC). The selected organizations will provide the blockchain infrastructure required to run a Hyperledger Fabric network. In the next step, NEC sets up an ABVS hyperledger-based blockchain network. This consists of creating a Certificate Authority, which provides complying with X.509 standard certificates and distributes them between accepted organizations, which made up the ABVS network. Furthermore, NEC generates Vote Authentication Tokens (VITs), which are one-time numerical codes similar to Indexed Transaction Authentication Numbers (iTANs) that allow citizens to

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

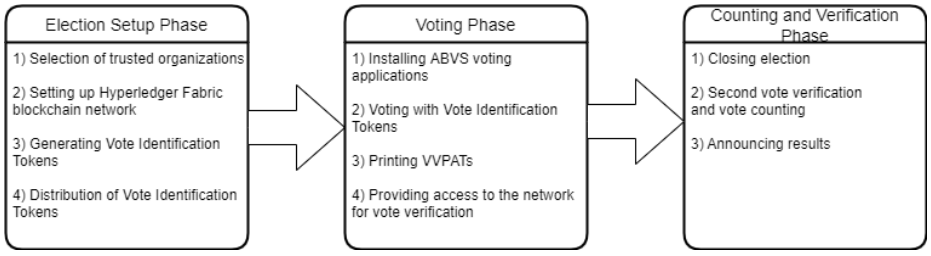

Fig. 4. Auditable Blockchain Voting System overview

cast their votes. VITs are then distributed over the country, and the citizens can retrieve them from local offices after authorization and authentication. All retrievals are recorded in a blockchain, so it is possible to verify if only authorized people obtained the VITs.

In the *voting* phase, ABVS certified voting apps are installed in polling stations. In order to vote, the voters provide one of the obtained VITs and select their chosen candidate. The ABVS applications transfer votes to the blockchain network, where they are validated. The voters receive a validated and accepted transaction from which they generate VVPATs to leave a physical trail. The voters can also use their VITs to verify the presence of their votes in the blockchain. Furthermore, as long as the voters have their VITs, they can keep casting votes because only the newest one will be counted. It is also important to note that all votes are encrypted and can only be decrypted after the election time is over, so elections remain fair.

In the *counting and verification* phase, the election is closed and second verification is conducted, ensuring that all votes were cast by distributed VITs. After validation is over, the votes are calculated and announced.

4.2 Auditable Blockchain Voting System network

The organisation of the Auditable Blockchain Voting System network is presented in figure 5. The network consists of a single consortium made of an NEC node, which is an initial organisation that stated initiated the network, and a set of trusted organisations $Orgs = \{Org_1, Org_2, ..., Org_n\}$, which are selected before the formation of the network. Each organisation adds to the network a number of orderer-peer nodes $Nodes = \{Org_1Node_1, Org_2Node_2, ..., Org_nNode_n\}$. The network is organised into two channels: VITsDistribution Channel and Voting Channel.

VITsDistribution Channel is used connected to VITsDistribution Applications, which allows voters to obtain their voting tokens. After the ABVS network is set up, NEC generates a predefined number of VITs, which are recorded as blockchain transactions in the channels' ledger. To obtain the code, each willing voter must report to a local office, authorize and authenticate to ensure eligibility and use the application to send a request for VIT. The ABVS network will

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

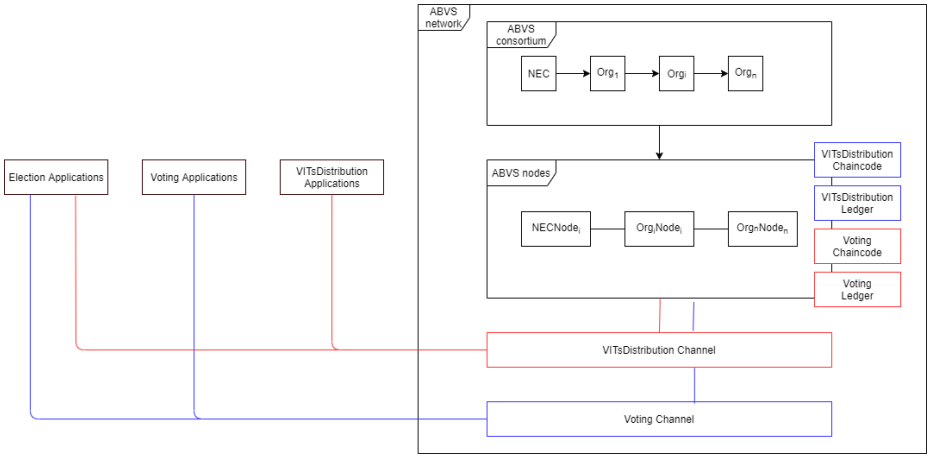

Fig. 5. Auditable Blockchain Voting System network organization

return one of the generated VITs and record its retrieval to ensure that only retrieved VITs are used in given voting.

Voting Channel is used to manage the election itself, and it contains the main business logic of ABVS. Through this channel, elections are generated, and votes are cast and recorded for further counting. This is done via dedicated client applications designated as *Voting Applications*.

Furthermore, both channels are connected to *Election Applications*, which are designed for election process administrators to manage the election process. The applications allow creating an election, generating VITs, validate elections and produce results.

4.3 Auditable Blockchain Voting System smart contracts

ABVS utilizes three main assets presented in figure 6. The *election* represents the current election and contains fields describing what the election is about (*electionGoal*), a list of candidates, and the start and end dates of the given election. A second asset consists of *VITs* and each consists of a given Election asset and a map of index-value pairs used for the given election. The final assets are *Votes*, each consisting of a given Election, VIT used for casting a vote, value of a vote and a vote location, which describes an electoral district or a university department in case of more local elections.

Each node in the ABVS system has installed chaincode, which allows its execution. There are three main smart contracts included in the ABVS chaincode, one for each asset.

Election contract and election applications provide an interface, as shown in figure 7. The election contract allows administrators to:

1. query for elections,

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

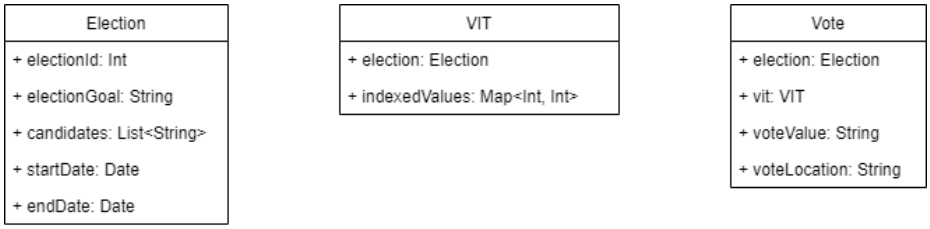

Fig. 6. Auditable Blockchain Voting System assets

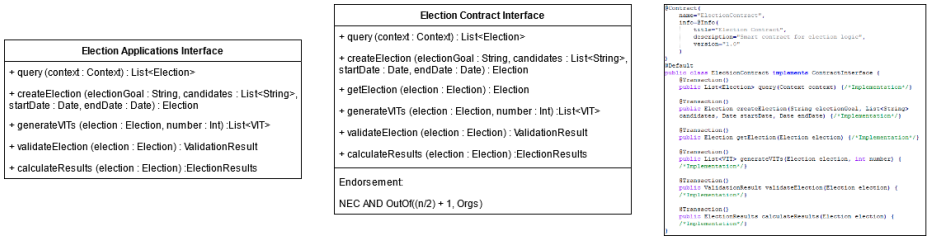

Fig. 7. ABVS election application and contract interfaces

2. create elections for specific goal, candidates and dates,
3. get an election for use in other smart contracts,
4. generate VITs for creating a specific number of VITs codes for a given election,
5. validate an election by comparing used and distributed VITs,
6. calculate results by decrypting and counting votes.

The endorsement policy for this contract requires endorsement from NEC and at least $\frac{n}{2} + 1$ of other organisations. The election applications' interface utilizes the methods from the contract interface except for the *getElection()* method, which is for internal use only.

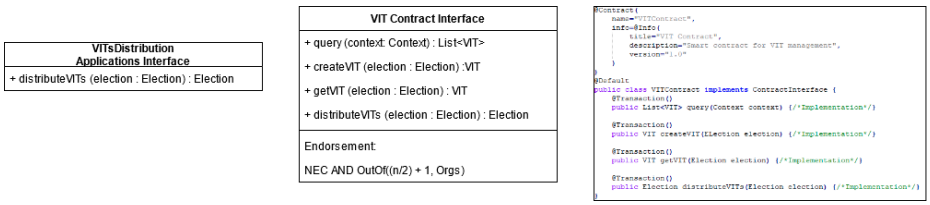

Fig. 8. ABVS VITsDistribution applications and VIT contract interfaces

VIT contract interface provides access to methods for:

1. querying the blockchain for specific VITs,

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

2. creating a new VIT for a given election,
3. getting a specific VIT for use in other methods and smart contracts,
4. distributing a VIT to voters and saving this operation.

The endorsement policy consists of a required endorsement of NEC and at least $\frac{n}{2}+1$ of other organisations. The VITsDistribution applications utilize only the *getVIT()* method, which returns a VIT to voters and logs it as a transaction (Fig. 8).

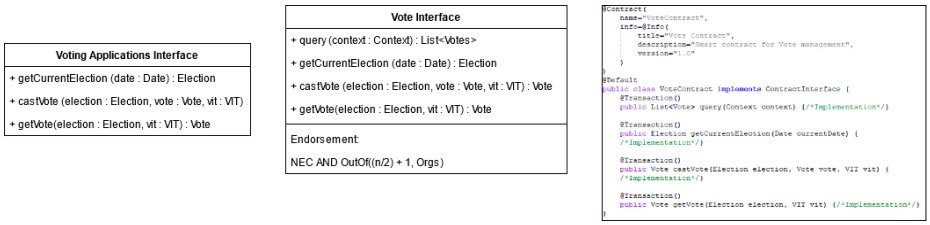

Fig. 9. ABVS Voting applications and Vote contract interfaces

Figure 9 presents interfaces provided by voting applications and Vote contract. The contract allows to:

1. query the blockchain for a specific Vote,
2. get the current election via smart contract communication,
3. cast a vote for a given election by providing a VIT,
4. get a vote by providing a VIT, which is a method for verification by voters.

The endorsement policy consists of a required endorsement of NEC and at least $\frac{n}{2}+1$ of other organisations. The voting applications provide methods for getting a the current election to facilitate the process, casting votes by providing a VIT, and getting a vote for verification by providing a VIT.

4.4 Auditable Blockchain Voting System security components

ABVS utilizes standard Certificate Authorities (CAs) distributing X.509 certificates to the involved entities. Thus it uses out-of-the-box distribution via Root CAs, Intermediate CAs and MSPs. However, ABVS also requires role functionality and role-based access control to the provided interfaces. To accomplish that, Attribute-Based Access Control was implemented utilizing an additional type attribute, which is added to each certificate to restrict access to various functionalities.

The methods that can be accessed by a *VoterApp* role, which represents client applications for retrieving VIT and casting votes, are: *getVIT* and *castVote*. *OrgMember* role is designed for organizations, which do not initiate elections and should not have any additional options to affect the election. Its methods are:

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

getElection, *validateElection*, *calculateResults* and all query methods. Finally, *NECAdmin* roles is designed with the same privileges as OrgMember, but can create an election and generate VITs for a given election.

Furthermore, it is important to note that all transactions remain encrypted until the end of the election, and no involved entity can perform any operation besides adding new transactions. Only voters can check their votes before the election ends.

5 Property Analysis

Eligibility is achieved in two ways. Firstly, all organizations participating in the network use public-key cryptography and are certified by the CA created by NEC, making unauthorized access to the network difficult and easily detectable. Secondly, authentication and authorization for the voters are achieved by the fact that the system was designed as a supervised system, which means that the voting takes place under official control.

$Eligibility = \{publicKey \cup certifcationCA\} \vee \{supervisedSystem\}$

Privacy is achieved by allowing supervision during the election process. The voters authenticate and authorize themselves before an election committee and then proceed to cast their vote using their VITs, which are in no way linked to specific voters besides physical ownership. The voters cast their votes anonymously from the privacy of polling stations. This has two major disadvantages. Firstly, it requires voters to travel to the polling stations, which nullifies one of the greatest advantages of electronic voting. Secondly, voters can be forced to provide their VITs to show how they voted. However, this is mitigated by the fact that ABVS allows multiple voting and counts only the last vote cast by a given voter.

$Privacy =$
$\{supervisedSystem \cup authenticatation \cup authorization \cup voterVIT\}$

Correctness/completeness is achieved because only authenticated and authorized voters are allowed to participate in an election, so only eligible votes are cast. Furthermore, the voting applications will ensure that only valid votes are transferred to the ABVS network. Finally, all transactions are validated by the ABVS network itself, so no invalid votes should be counted.

$Correctness =$
$\{authenticatation \cup authorization \cup ABVSnetworkValidation\}$

Fairness is achieved by encryption of data stored on the ABVS network through the use of private-collections functionality of Hyperledger Fabric. Moreover, access to this data is restricted to the specific role of the OrgMember, which restricts who can view the stored data. Finally, every operation in the network is saved as a transaction, so the property is further reinforced because it is relatively easy to identify eventual leaks of the data, which may affect election results.

$Fairness =$
$\{dataEncryption \cup HyperledgerFabricPrivate\_collections \cup OrgMember\_role \cup transaction\_operation\}$

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

Transparency and verifiability are achieved because of blockchain inherent properties and because each operation on the Hyperledger network is saved. Furthermore, all available operations are based on smart contracts, which are public and thus can be validated. Lastly, voters can use their VITs to view their cast votes to see if their vote was saved correctly.

$Transparency =$
$\{blockchainProperty \cup operationSaving \cup smartContracts \cup voterVIT\}$

Availability is achieved because, in essence, the ABVS is an evolution of the traditional paper-based methods and uses similar procedures, so it is no less available than its predecessors.

$Availability = \{ABVS\_properties\}$

6 Conclusions

An honest and fair democracy requires a quick and efficient election system. In order to improve the traditional paper-based voting, numerous electronic systems were designed. However, many suffer from issues with transparency, verifiability and privacy. Rapidly developed blockchain technologies may offer a solution to these still not solved problems.

Hyperledger Fabric is one of the blockchain-based platforms for developing applications on top of it. It is an open-source project by The Linux Foundation created to provide a blockchain-based solution for enterprise private and permissioned network. It is characterized by high customizability and modularity in every aspect, from identity management to smart contract validation and consensus algorithm. These reasons made this platform ideal for implementing the electronic voting system because such a system requires a specific set of settings, which may not be available in more popular platforms like Bitcoin or Ethereum.

In this paper, high-level implementation details of the Auditable Blockchain Voting System were presented. The most important components of ABVS were presented concerning various Hyperledger Fabric components and how these components interact. Models of ABVS network, smart contracts and privacy settings were shown and described. In the future, the implementation will be tested to identify the most optimal parameters and settings for a quick and scalable e-voting system. Furthermore, implementation in the MultiChain platform will be developed to verify the portability of the ABVS model.

References

1. F. Lehoucq: Electoral Fraud: Causes, Types, and Consequences. Annual Review of Political Science. Vol. 6, no. 1, pp. 233–256 (2003)
2. J. Willemson, Bits or paper: Which should get to carry your vote?. Journal of Information Security and Applications. Vol. 38, pp. 124-131 (2018)
3. C. De Faveri, A. Moreira and J. Araújo: Towards security modeling of e-voting systems. Proc. of IEEE 24th International Requirements Engineering Conference Workshops (REW), Beijing, China (2016)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51

4. M. Pawlak, J. Guziur and A. Poniszewska-Maranda: Voting Process with Blockchain Technology: Auditable Blockchain Voting System. Advances in Intelligent Networking and Collaborative Systems. INCoS 2018. LNDECT, Vol. 23, pp. 233–244 (2018)
5. Hyperledger: A Blockchain Platform for the Enterprise: Hyperledger Fabric. Linux Foundation, 4 September 2019. [Online]. Available: https://hyperledger-fabric.readthedocs.io/en/latest/. Accessed 15 Semptember 2019.
6. S. Caarls: E-voting handbook: Key steps in the implementation of e-enabled elections. Council of Europe, November 2010. [Online]. Available: https://www.coe.int/t/dgap/goodgovernance/Activities/E-voting/E-voting%202010/Biennial_Nov_meeting/ID10322%20GBR%206948%20Evoting%20handbook%20A5%20HD.pdf.
7. T. Dimitriou: Efficient, Coercion-free and Universally Verifiable Blockchain-based Voting. Computer Networks. Vol. 174 (2020)
8. K.M. Khan, J. Arshad and M.M. Khan: Investigating performance constraints for blockchain based secure e-voting system. Future Generation Computer Systems. Vol. 105, pp. 13–26 (2019)
9. National Democratic Institute. Common Electronic Voting and Counting Technologies. [Online]. Available: https://www.ndi.org/e-voting-guide/common-electronic-voting-and-counting-technologies. Accessed 22 January 2018.
10. Council of Europe – Committee of Ministers: Recommendation CM/Rec(2017)51 of the Committee of Ministers to member States on standards for e-voting, 14 June 2017. [Online]. Available: https://search.coe.int/cm/Pages/result_details.aspx?ObjectID=0900001680726f6f.
11. Council of Europe – Committee of Ministers: Recommendation Rec(2004)11 of the Committee of Ministers to member states on legal, operational and technical standards for e-voting, Strasbourg: Council of Europe Publishing (2005)
12. D. Drescher: Blockchain Basics: A Non-Technical Introduction in 25 Steps, 1 ed., Frankfurt am Main: Apress (2017)
13. N. Gaur, L. Desrosiers, V. Ramakrishna, P. Novotny, S.A. Baset and A. O'Dowd: Hands-On Blockchain with Hyperledger. Building decentralized applications with Hyperledger Fabric and Composer, Birmingham: Packt Publishing Ltd (2018)
14. D. Ongaro and J. Ousterhout: In Search of an Understandable Consensus Algorithm (Extended Version), 20 May 2014. [Online]. Available: https://raft.github.io/raft.pdf. Accessed 10 August 2020.
15. S. Xiao, X.A. Wang, W. Wang and H. Wang: Survey on Blockchain-Based Electronic Voting. INCoS 2019. Advances in Intelligent Systems and Computing. Vol. 1035, pp. 559–567 (2019)
16. D. Kirillov, V. Korkhov, V. Petrunin, M. Makarov, I.M. Khamitov and V. Dostov: Implementation of an E-Voting Scheme Using Hyperledger Fabric Permissioned Blockchain. ICCSA 2019. LNCS, Vol. 11620, pp. 509–521 (2019)
17. Q. He and Z. Su: A new practical secure e-voting scheme. Information Security Conference (SEC 1998) (1993)
18. S. Zhang, L. Wang and H. Xiong: Chaintegrity: blockchain-enabled large-scale e-voting system with robustness and universal verifiability. International Journal of Information Security, https://doi.org/10.1007/s10207-019-00465-8 (2019)
19. Agora Technologies: Agora_Whitepaper_v0.2.pdf," 2015. [Online]. Available: https://agora.vote/Agora_Whitepaper_v0.2.pdf. Accessed 20 April 2018.
20. B. Adida: Helios: Web-based Open-Audit Voting. Proc. of 17th USENIX Security Symposium (2008)

ICCS Camera Ready Version 2021
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-77961-0_51