Technium Vol. 11, pp.32-43 (2023)
ISSN: 2668-778X
www.techniumscience.com

AQUA: A BLOCKCHAIN BASED MULTI-WINNER E-VOTING SYSTEM

Stefanos Nikolopoulos, Nikos Karanikolas, Christos Kaklamanis

University of Patras and CTI “Diophantus”, Greece
snikol@ceid.upatras.gr

Abstract. Conducting fair and secure elections has been the heart of democracy since its birth in ancient Greece. To be more precise, the act of voting has been an essential and significant component of human societies, particularly for government and municipal elections. Nonetheless, the conventional method of casting a ballot by physically visiting a voting booth raises concerns regarding the security of the process, as there have been reported incidents of tampering. To tackle these issues, we propose an online voting system that makes use of Blockchain technology. By utilizing encryption and hashing techniques, the Blockchain system guarantees the security of each vote. In our proposed system, a smart contract is triggered when an election is scheduled, thereby facilitating a secure and efficient voting process. The need to modernize voting systems has become more urgent due to the increasing frequency and volume of voters. Despite these challenges, security and fairness should not be compromised. Therefore, modernizing voting systems in terms of security is essential to ensure the integrity of elections, so that voters can have confidence in the electoral process.

To address this challenge, we introduce Aqua, a distributed system that implements a multi-winner approval-based voting scheme, following the AV rule, on the Ethereum blockchain. Aqua supports both English and Greek languages for multi-winner elections with committee sizes up to three members. It is user-friendly, cost-effective, and accessible from any device with an Internet connection.

Aqua voting is based on the AV rule, but it can be readily modified to accommodate any multi-winner voting scheme with any committee size.

Keywords. Aqua Voting System, Blockchain, Blockchain Voting, Ethereum, AV rule.

1. Introduction

With the advancement of technology in different aspects of our lives, the use of electronic devices has risen. The initial introduction of electronic devices in the voting scheme dates back to the early 1890s when Herman Hollerith invented the punch card, this simple technology has evolved over time to become today's electronic voting systems. Electronic voting (e-voting), is a term that refers to any voting process that employs electronic means to either conduct or count the votes. To be more precise, e-voting is a form of voting that uses electronic means to either aid or take care of casting and counting ballots. Depending on the particular implementation, e-voting may use standalone electronic voting machines (also called EVM) or computers connected to the Internet (online voting). E-voting has been used in various countries such as Brazil, India, Estonia, Switzerland, and Belgium. Some advantages of e-voting include faster vote counting, reduced costs, and increased accessibility for disabled voters. However, some disadvantages include security concerns such as hacking and tampering with votes.

The evolution of voting methods has transitioned from traditional public voting to the use of paper ballots, while electronic voting has progressed from punch cards to optical scanning systems at polling centers and even online voting. This evolution has been driven by the need for more efficient, secure, and accessible methods of voting. The concept of an electronic online voting system is to provide the opportunity to conduct elections via the internet which can make the voting process even more convenient and lead to greater participation since anyone could participate with a click of a button. Security, speed, and ease of use of these systems are some of the motivations for the development of online voting technology. However, security concerns are holding back the widespread adoption of such systems.

The remote nature of voting over the internet allows corrupt or malicious software to exploit security vulnerabilities in both the voters' computers and the electronic voting system itself. Hence, the need to protect against these attacks had led to the evolution of cryptographic techniques on voting systems. Some important cryptographic models used in such systems are AES, RSA, Elliptic Curve Cryptography (ECC), El Gamal, DSA and hash functions. In this way, the anonymity of the process is maintained, voters are given the opportunity to confirm their votes and it is possible to create a secure system from external attackers.

Traditional e-voting systems are based on central servers and anyone who has access to them can change the outcome of the democratic process. A vote can be considered free and fair only when there is faith in the individuals who hold positions of power. This is why blockchain technology is considered by many to be the ideal tool for creating a modern secure electronic voting system.

In essence, blockchain is a shared, immutable ledger that facilitates the process of recording transactions and tracking assets in a business network. An asset can be tangible (a house, car, cash, land) or intangible (intellectual property, patents, copyrights, branding). In addition, blockchain is a data structure that is distributed, immutable, secure, and anonymous. Data is stored in blocks and added to the chain without changing the existing data. Changes are made only through general consensus of the users. The main reason why the blockchain is so secure is because of its decentralized nature. It is not possible for any single entity to fully control it.

As modern societies increasingly embrace the practice of voting, there is a growing demand for conducting elections to form committees. This trend is reflective of a desire for more democratic and inclusive decision-making processes, where all members have a say in shaping the direction of the committee's work. By utilizing fair and transparent election processes, committees can be formed in a manner that represents the interests and preferences of all stakeholders, resulting in more effective and representative decision-making. Consequently, there has been a surge of research in the field of social choice theory, with a particular focus on the subfield of multi-winner voting. Multi-winner voting is used to elect various bodies such as parliaments, boards of directors, or even to address everyday issues like designing a dynamic poll or questionnaire. The selection process for such elections depends on the specific voting rules that are applied. In this work, we will focus on approval voting, where voters approve a set of alternatives on their ballots [11]. Additionally, we will examine multi-winner voting rules [5,10], which determine committees of alternatives as outcomes [4,9]. Specifically, we will concentrate on approval-based voting rules, with multiwinner approval voting (AV) being a well-known example. In this method, each alternative receives a point every time it is approved by a voter, and the outcome is a fixed number of alternatives with the highest score, i.e., points. We focus on approval voting since it is i) a relatively simple and intuitive voting method, ii) robust to strategic voting and manipulation, iii) flexible and can be adapted to various multi-winner voting systems, iv) used in a variety of real-world applications, from political elections to corporate board elections.

1.1. Related Work

Electronic voting (e-voting) has gained considerable attention in recent years due to its potential to enhance the transparency and security of the voting process. One of the most promising technologies for e-voting is blockchain, which offers a tamper-proof and decentralized platform for recording and verifying votes. Smart contracts, which are self-executing programs that run on the blockchain, can be used to automate the voting process, and ensure that it is conducted fairly and transparently. Several researchers have proposed e-voting systems based on blockchain technology and smart contracts, aiming to address some of the challenges of traditional e-voting systems. One of the first works on the topic is the one by Hjálmarsson et al. [7], which proposes a decentralized e-voting system based on smart contracts using Ethereum blockchain. The system allows voters to register, vote and verify their votes through a web application. The system also provides real-time results and prevents data modification by third parties.

On the same direction is the work of Indapwar et al. [8], where they proposed a blockchain-based e-voting system that utilizes the Ethereum framework. The system adopts a hybrid architecture that combines permissioned, public, and shared blockchains to enable public scrutiny of the votes and ensure transparency. The system also employs smart contracts to facilitate e-voting applications, in which data blocks are appended to the ledger. Each data block contains multiple votes that are validated by the smart contract before being added to the blockchain ledger. Otherwise, the votes are rejected. A major challenge for this system is the transaction verification speed.

In the work of A. M. Al-Madani et al. [1], the authors have proposed a decentralized e-voting system based on smart contract by using blockchain technology. Their system allows voters to register and vote through a web application that interacts with a smart contract deployed on the Ethereum blockchain. The smart contract validates the votes and stores them in the blockchain, ensuring their security and immutability. The system also provides real-time results of the voting.

Another work related to smart contracts and voting is the one by Sadia et al. [12], where they propose a system that uses smart contracts to ensure privacy, security, and accuracy of e-voting. It prevents vote manipulation and provides a transparent, immutable, and verifiable process. The system has three phases: Pre-voting, Voting and Post-voting. In the Pre-voting phase, the Organizer sets the criteria for eligible voters, alternatives, and election time on the genesis block. Voters are randomly assigned to groups with different voting times. In the Voting phase, smart contracts check the eligibility and group membership of voters and control the voting time. No votes are accepted after the voting time expires. The voter’s fingerprint is used as a private key to match the eligible list and to generate a hash for identity protection. The voter selects an alternative with a unique binary string and creates a ballot with a number and a hash. The ballot is stored in a block with a sibling block that has the voter’s hashed fingerprint. The ballot block is broadcast for verification and POW, while the sibling block is kept secret. After the voting time ends, the sibling blocks are broadcasted, and the results are revealed. The system allows a short grace period for late voters.

Fusco et al. [6] designed an e-voting system called Crypto-voting that uses permissioned blockchain technology. The system consists of two linked blockchains: one for recording voting procedures and voters, and another for counting votes and providing results. The system emphasizes the anonymization of the network consensus nodes. Smart contracts handle voting procedures and results. The system enhances the efficiency of the validation phase, the allocation of the votes to the alternatives, the automatic management of electoral lists, the secure timing of voting abroad, the integration of the identification process with the voting secrecy, and the automatic and reliable technique to ensure the voting security. The system also addresses architectural issues such as privacy tools to protect the blockchain in the Cloud system.

Finally, another prominent work is the one by Alvi, et al. [2], which proposed a digital voting system based on blockchain technology using biohash and smart contract. Their system uses biometric authentication to verify the identity of voters and generate a unique biohash for each voter. The biohash is used as a private key to sign the votes and send them to a smart contract deployed on the Ethereum blockchain. The smart contract verifies the validity of the votes and records them in the blockchain, ensuring their privacy and integrity. The system also allows voters to audit their votes and check the results of the voting.

1.2. Our Work

Our work aims to address the aforementioned security concerns that may arise in electronic voting systems, and concentrates on providing security solutions by utilizing blockchain technology for electronic voting systems. The Aqua voting system is a decentralized application (DApp) based on the Ethereum blockchain, which implements a multi-winner voting system for selecting committees of size up to k=3 when approval votes are provided as input and approval voting (AV) rule is used for computation. We focus on multi-winner elections and approval voting since it has the following advantages.

- Simplicity: Approval voting is a relatively simple and intuitive voting method, where voters indicate which alternatives they approve of. This simplicity makes it accessible to a wider range of voters, and it can be easily implemented in various settings.
- Robustness: Approval voting is robust to strategic voting and manipulation, as voters are encouraged to vote for all alternatives that they support, rather than just their favorite. This can lead to more accurate representation of voter preferences and more legitimate outcomes.
- Flexibility: Approval voting is flexible and can be adapted to various multi-winner voting systems. For example, approval-based proportional representation methods can be used to ensure that different groups within a constituency are represented.
- Real-world applications: Approval voting has been used in a variety of real-world applications, from political elections to corporate board elections. This means that there is a practical value in improving the security of approval voting methods.

Our system employs a smart contract, *Aqua.sol*, to ensure greater privacy, security, and accuracy, ultimately leading to a transparent, immutable, and verifiable process that rejects all attempts for electoral fraud. The uniqueness of Aqua stems from its pioneering use of blockchain technology to facilitate multi-winner elections. To the best of our knowledge, it is the first online voting system capable of conducting multi-winner elections. In addition to being affordable and easy to use, Aqua offers the added advantage of being accessible through any Internet-connected device.

2. Preliminaries and theoretical framework

2.1. Blockchain technology and security

Blockchain is a distributed, decentralized, and immutable ledger that is used to store and transfer data. Data are stored in block form and are appended to the chain without changing the existing blocks, changes in the chain are made only through the consensus of the network participants. This ensures that the blockchain is immutable and tamper-proof. The decentralized nature of blockchain technology means that there is no central authority or single point of failure, making it resistant to censorship and manipulation.

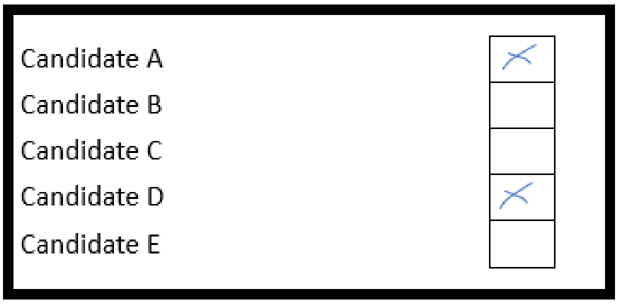
In more simple terms, as defined by Uzma, et al. [13] we call blockchain a set of technologies combining the blockchain data structure itself, distributed consensus algorithm, public key cryptography, Merkle trees and smart contracts.

Ethereum is an open-source blockchain platform on which decentralized applications (dapps) can be built using smart contracts. Smart contracts are self-executing programs that run on the Ethereum Virtual Machine (EVM). Ethereum has its own cryptocurrency called Ether (ETH), which is used as a form of payment for transactions on the Ethereum network. For every transaction on the Ethereum network, a "ticket" payment, known as gas, is required to validate the transaction [3]. While Ethereum has a lot in common with Bitcoin's blockchain, its major difference is that it was not designed only as a digital currency.

The cost of a smart contract in Ethereum blockchain is divided into two parts: deployment and execution. The deployment cost represents the complexity of the contract. The execution of a contract function is encoded as a transaction and eventually stored in blocks, therefore the gas required is limited since the block size is limited. A contract execution that costs more gas than the limit will fail. A failed transaction is still recorded as an attempted transaction and the gas spent on the execution is deducted from the original account, but otherwise has no other impact on the contract or account state.

2.2. Multi-winner voting

The approval-based model operates on the premise that each voter's preference information can be simplified to a binary choice between approved or disapproved alternatives, as depicted in Figure 1. When submitting their preferences, each voter provides a subset of alternatives they approve, effectively indicating their acceptance of those alternatives.



Candidate A ☒
Candidate B □
Candidate C □
Candidate D ☒
Candidate E □

Figure 1: An approval ballot. Here, the voter decided to approve two of the five candidates.

Formally, we denote by A the set of alternatives and we have $m = |A|$ alternatives in total. Also, we denote the committee size by k. The term committee refers to a set of exactly k alternatives. Let N denote the finite set of voters and $n = |N|$ the number of votes in a profile. An approval profile is the collection of all voters' preferences; formally it is a function $f: N \rightarrow P(A)$. We say that $V(i) \subseteq A$ is voter i's approval ballot. Further, we write $N(a)$ to denote the subset of voters that approve alternative a, i.e., $N(a) = \{i \in N : a \in V(i)\}$. Let's start by defining an "election instance" $E = (P, k)$, which includes a preference profile P and a desired committee of size k. An "approval-based committee voting rule" (ABC rule) is a method used to select committees. Namely, it takes E as input and produces one or more subsets of alternatives with size k, which are referred to as "winning committees". When $k = 1$, in a single-winner scenario, there are only a few reasonable ABC voting rules that can be used with approval ballots. One of the most natural rules is Approval Voting, which selects alternatives that are approved by the largest number of voters, with all of them being considered (co-)winners. However, in cases where $k > 1$, there is one ABC rule that extends the rationale of Approval Voting in the most natural manner. This rule is called Multi-Winner Approval Voting, abbreviated as AV. Hence, Multi-Winner Approval Voting (AV) selects the k alternatives which are approved by most voters. Formally, the AV-score of an alternative $a \in A$ is defined as $scoreAV(P, a) = |N(a)| = |\{i \in N : a \in V(i)\}|$ and AV selects committees W that maximize $scoreAV(P, W) = \sum_{a \in w} scoreAV(P, a)$.

Example 2.1: A board chooses a steering committee. Such a committee consists of three persons (k = 3) and there are six alternatives competing for these positions, $A = \{0, 1, 2, 3, 4, 5\}$. All members of the board are eligible to vote and provide approval ballots to indicate their preference. In total, 10 votes have been submitted (n = 10). The profile is:

$\boldsymbol{V(1) : \{1, 3, 5\}}$	$\boldsymbol{V(2) : \{2, 5, 4\}}$	$\boldsymbol{V(3): \{3, 4, 5\}}$	$\boldsymbol{V(4) : \{1, 4, 5\}}$
$\boldsymbol{V(5) : \{2, 3, 5\}}$	$\boldsymbol{V(6) : \{1, 3, 4\}}$	$\boldsymbol{V(7): \{2, 4, 5\}}$	$\boldsymbol{V(8) : \{5, 0, 4\}}$
$\boldsymbol{V(9) : \{3, 5, 0\}}$	$\boldsymbol{V(10) : \{4, 5, 0\}}$		

The winning committee under the AV rule, for k=3, is {5, 4, 3}.

3. Software Architecture of Aqua

Our voting system, Aqua, constitutes a decentralized application that relies on the Ethereum blockchain. Its primary function is to implement a multi-winner voting system that selects committees of size up to k=3 by utilizing approval votes as input and applying the AV rule during computation. In this section we present Aqua's software architecture and design.

The Aqua system is comprised of four distinct phases: “Initialization”, “Registration”, “Voting”, and “End”. Upon deployment of the contract, the VotingInitiator initiates a transition from the “Initialization” phase to the “Registration” phase. During the “Registration” phase, the VotingInitiator uses the public key provided by Metamask to determine the list of voters and alternatives. In the subsequent “Voting” phase, eligible voters can cast their approval votes, while in the final “End” phase, the VotingInitiator "closes" the election, computes the results, and determines the winning committee. Figure 2 shows the flow diagram of the system.

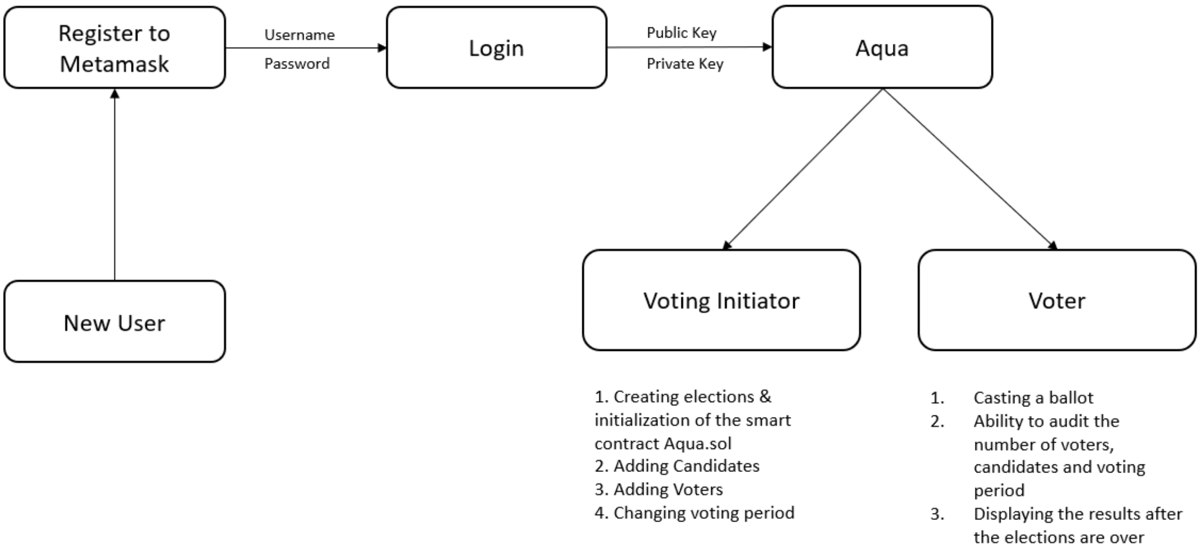

Register to Metamask
Username
Password
Login
Public Key
Private Key
Aqua
New User
Voting Initiator
Voter
1. Creating elections & initialization of the smart contract Aqua.sol
2. Adding Candidates
3. Adding Voters
4. Changing voting period
1. Casting a ballot
2. Ability to audit the number of voters, candidates and voting period
3. Displaying the results after the elections are over

Figure 2: Aqua flow diagram

4. Features of Aqua

In this section we present the various functionalities and capabilities of our multi-winner voting software system that have been implemented to ensure its effectiveness and efficiency. By detailing these features, we aim to provide a comprehensive understanding of the system's inner workings and its potential to address the challenges of traditional voting systems regarding security.

In order to illustrate the features of Aqua, we will provide an example of a real election and present its visual representation through the following series of screenshots. By using this example, we aim to showcase how our voting system operates in practice and how it can effectively address the challenges of multi-winner elections. The following demonstration will offer insights into the unique capabilities of Aqua and provide a clear understanding of how it can benefit both voters and election administrators. In this election a committee of size k=3 is elected by following the AV rule.

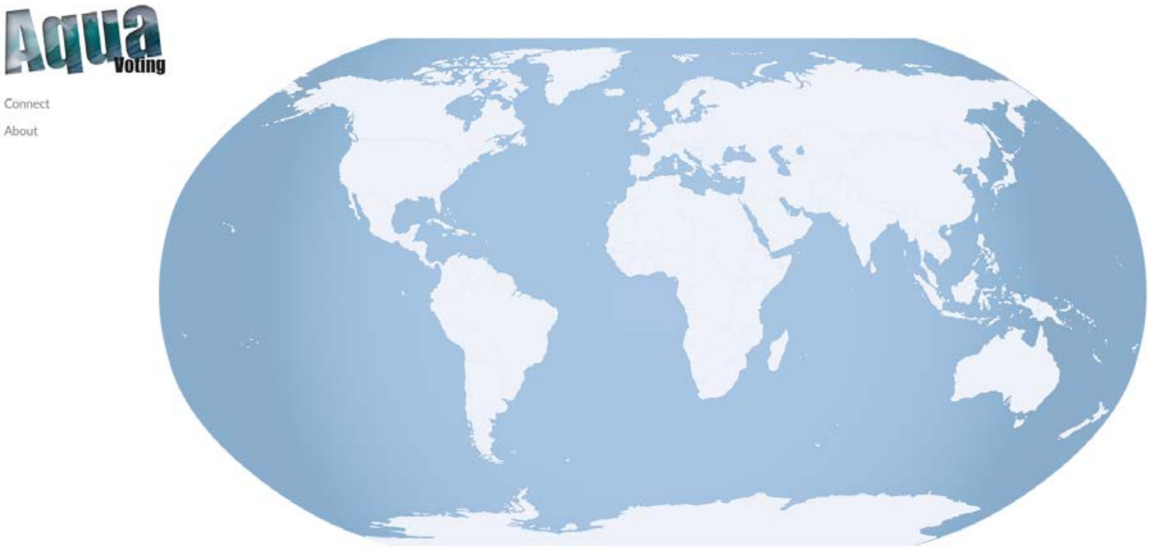
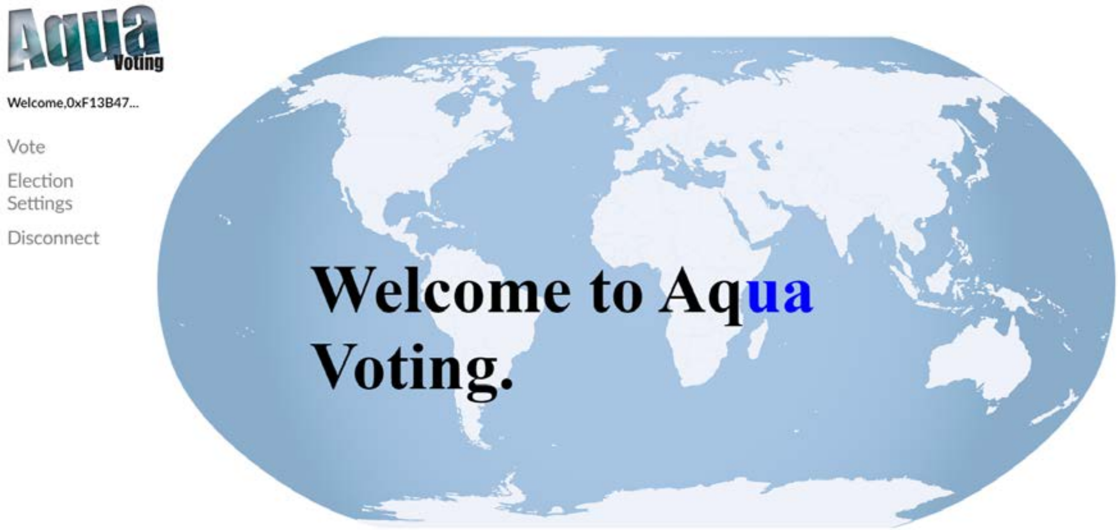


Figure 3: index.php

In Figure 3, we see the home page of Aqua, before the user logs in to Metamask. The options here are two: the “About" page which displays some information and instructions concerning the system and the "Login" page in which the user can log in to Metamask.

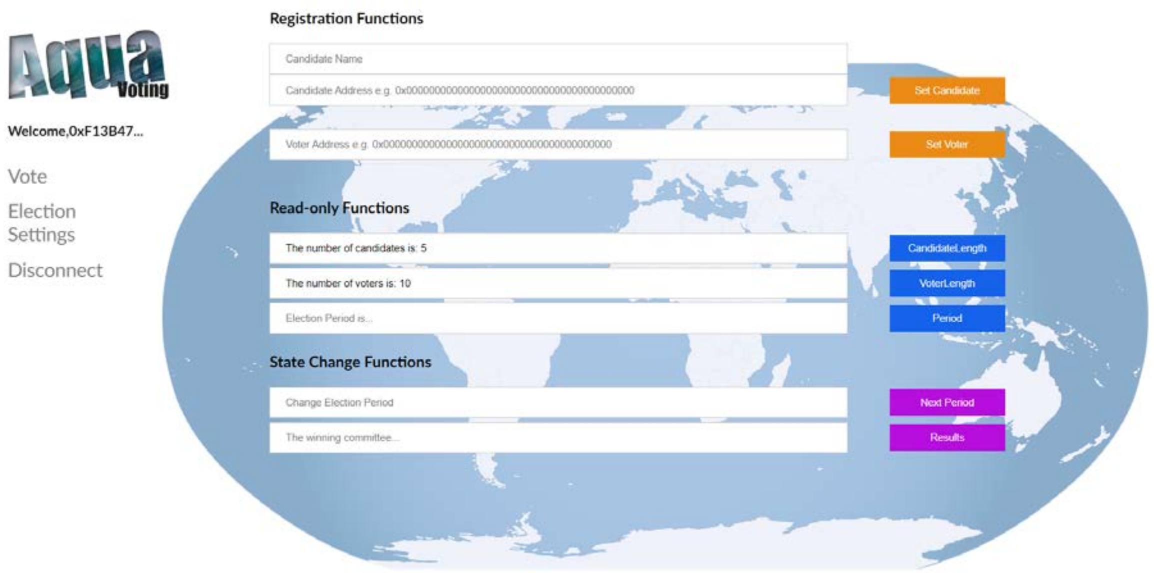


Welcome,0xF13B47...

Welcome to Aqua Voting.

Figure 4: u_index.php

Figure 4 displays the webpage that the user is directed to upon logging into Metamask. It is important to mention that the system in the upper left corner with the label "Welcome,..." reveals the user's public address. In the event that the user switches accounts, this field will automatically update to reflect the new connection to Aqua.



Registration Functions

Candidate Name

Candidate Address e.g. 0x0000000000000000000000000000000000000000

Voter Address e.g. 0x0000000000000000000000000000000000000000

Read-only Functions

The number of candidates is: 5

The number of voters is: 10

Election Period is...

State Change Functions

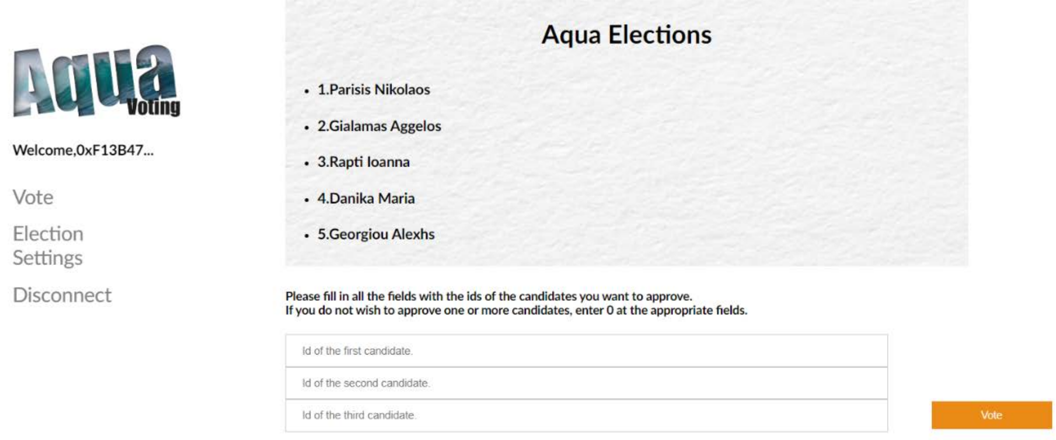
Change Election Period

The winning committee...

Figure 5: init_voting.php

Figure 5 displays the page that appears when the user selects "Elections Settings." The system has various functions, with some being read-only functions that do not require any gas and can be used by anyone, while others can only be performed by the VotingInitiator and require gas. These functions include adding an alternative or voter, displaying the number of alternatives, voters, and the voting period, changing the period status, and calculating results. If a user accidentally or intentionally performs a function incorrectly, such as adding a duplicate voter or alternative, performing an operation without being the VotingInitiator, or running an operation during the wrong period, the system will display the appropriate error message.

This trial election consists of five alternatives and ten voters, the alternatives are A = {Parisis Nikolaos, Gialamas Aggelos, Rapti Ioanna, Danika Maria, Georgiou Alexhs}.



Aqua Elections

- 1.Parisis Nikolaos
- 2.Gialamas Aggelos
- 3.Rapti Ioanna
- 4.Danika Maria
- 5.Georgiou Alexhs

Please fill in all the fields with the ids of the candidates you want to approve.
If you do not wish to approve one or more candidates, enter 0 at the appropriate fields.

Id of the first candidate.

Id of the second candidate.

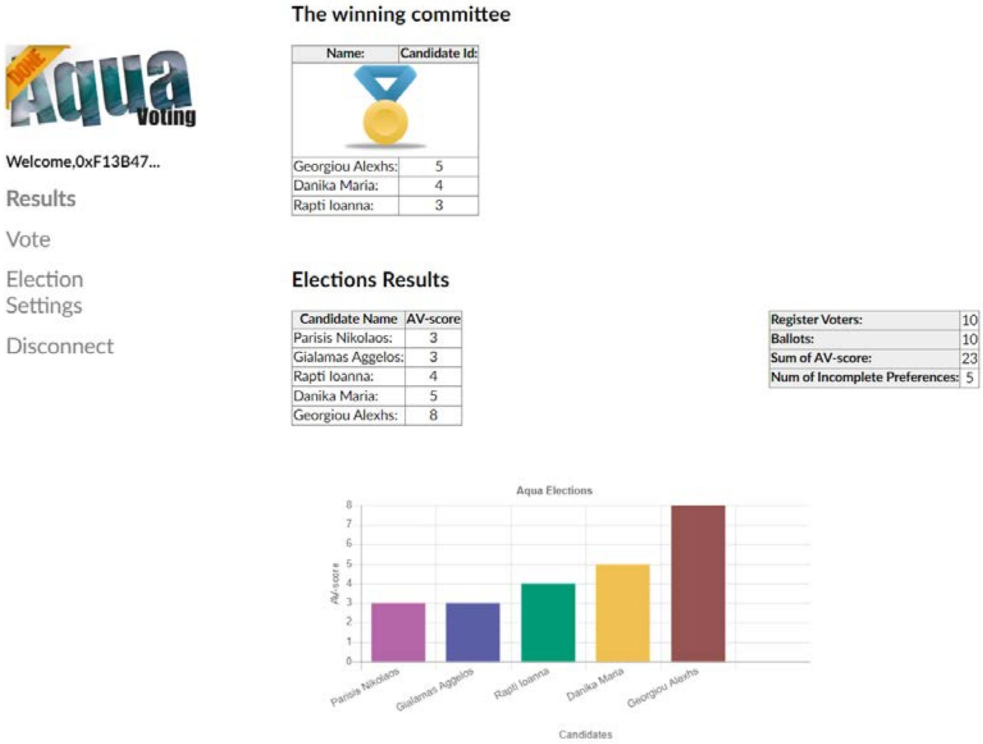
Id of the third candidate.

Figure 6: vote_voting.php

Figure 6 shows the page "Vote". This page displays the alternatives' id codes and names, and the user (voter) can complete their approval ballot. If the user chooses not to approve one or more options, they can enter 0 in the relevant fields. In case of an error, the page will display it on the user's screen.

The preferences of the voters in this trial election were:
Voter1 = {Parisis Nikolaos, Rapti Ioanna, Georgiou Alexhs}
Voter2 = {Gialamas Aggelos, Danika Maria}
Voter3 = {Rapti Ioanna, Danika Maria, Georgiou Alexhs}
Voter4 = {Parisis Nikolaos, Danika Maria, Georgiou Alexhs}
Voter5 = {Gialamas Aggelos, Rapti Ioanna, Georgiou Alexhs}
Voter6 = {Parisis Nikolaos}
Voter7 = {Gialamas Aggelos, Danika Maria, Georgiou Alexhs}
Voter8 = {Georgiou Alexhs}
Voter9 = {Rapti Ioanna, Georgiou Alexhs}
Voter10 = {Danika Maria, Georgiou Alexhs}

When the VotingInitiator wants to end the election and display the "Results" page to everyone, the VotingInitiator must change the period to "End" and perform the "Results" function. Once the function is successfully executed, the system will automatically redirect to the "Results" page.



The winning committee

Name:	Candidate Id:
Georgiou Alexhs:	5
Danika Maria:	4
Rapti Ioanna:	3

Elections Results

Candidate Name	AV-score
Parisis Nikolaos:	3
Gialamas Aggelos:	3
Rapti Ioanna:	4
Danika Maria:	5
Georgiou Alexhs:	8

Register Voters:	10
Ballots:	10
Sum of AV-score:	23
Num of Incomplete Preferences:	5

Figure 7: results.php

Finally, in Figure 7, we see the "Results" page. The results depicted refer to the election of the example. In this example the following winning committee {Georgiou Alexhs, Danika Maria, Rapti Ioanna} was computed using the AV rule.

5. Implementation Details

In this section we are going to present the implementation details of Aqua. In order to verify the proposed system, we utilized multiple technologies to implement Aqua. These technologies include Solidity [3], an object-oriented, high-level language used to create smart contracts, specifically Aqua.sol the voting contract of our system. Additionally, we employed NodeJS [14] for server-side scripting to develop our back-end, Hardhat for Ethereum development, Ethers.js to connect the front-end with the blockchain, and simple HTML-PHP, CSS, and JavaScript for the front-end development. To simulate the blockchain network, we utilized the Goerli Testnet, and for quick testing of our DApp, we used a local solution called Ganache.

Github of Aqua: https://github.com/NikStef/AquaBlockchainVoting-english-version

There are *two* entities in Aqua: VotingInitiator and Voter.

VotingInitiator: Every smart contract has a unique function called constructor that runs only once when the contract is deployed on the blockchain. This function is used to define the VotingInitiator, who is the person responsible for deploying the election instance. Using Aqua, anyone can create numerous elections effortlessly, provided they have sufficient gas. Presently, the deployment cost of Aqua is estimated at 3€ based on the current gas price.

VotingInitiator can execute the following functions:

- setCandidate(): Adds an Alternative (Candidate), using alternative’s address and name, if the alternative’s address does not already exist and the voting period is “Register”.
- setVoter(): Adds a voter, using only the voter’s address, if the voter’s address does not already exists and the voting period is “Register”.
- getResults(): Calculates the results and the winning committee(s) if the voting period is “End”.
- changePeriod(): Alters the enum variable that contains the periods of the elections. While it's possible to automate this change over time, in this particular implementation, for convenience, the VotingInitiator triggers this function.

Voter: The voting protocol that describes the implementation functionalities of the voter is depicted in Figure 8. If the period is "Voting" and the voter's address is registered but has not cast a vote, their vote will be counted. However, once the vote is cast, they will not be able to vote again.

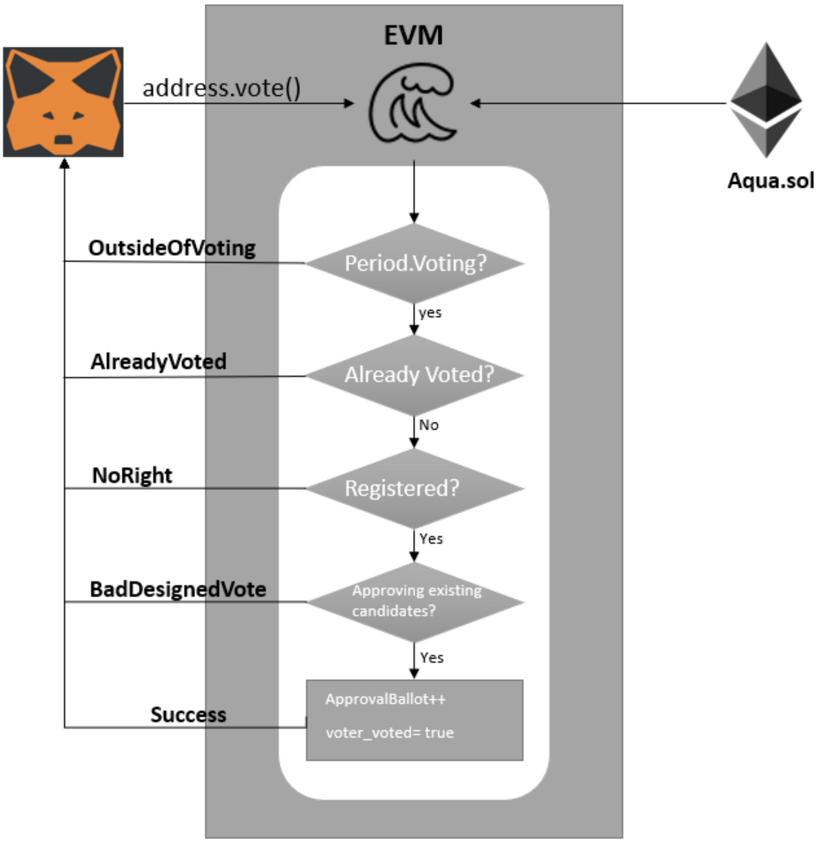
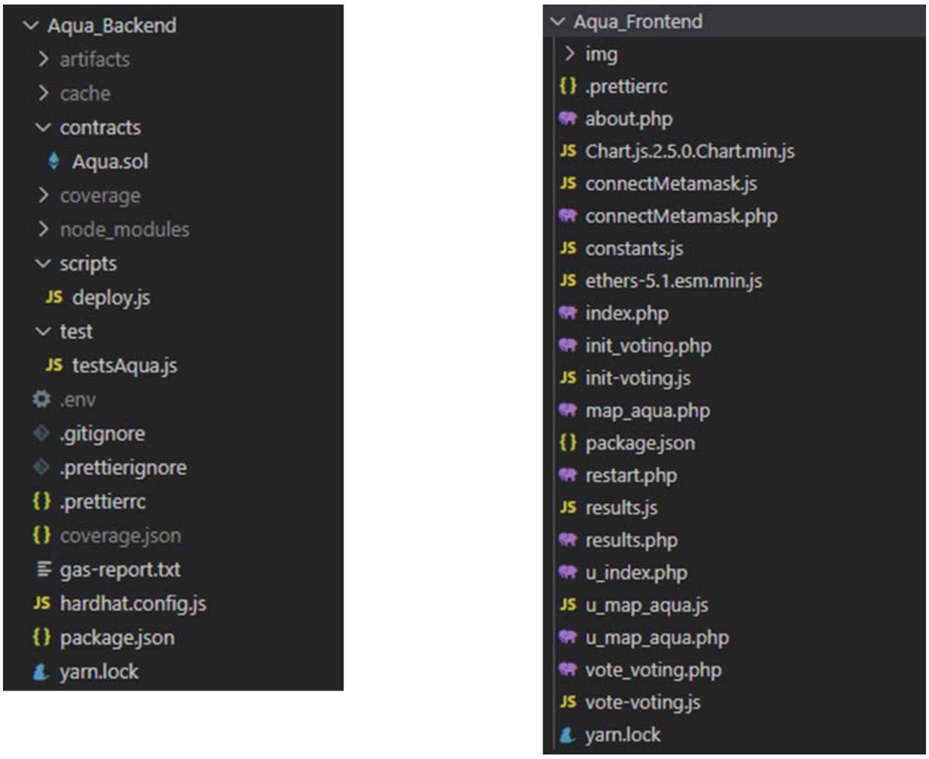


Figure 8: Voting Protocol

Apart from the functions that are exclusive to VotingInitiator and Voter entities, there exist functions that are accessible to everyone. These functions of Aqua are view functions, which means they only read data from the public blockchain. View functions do not require any gas to run.

In a few words:

- getState(): returns the voting period
- getCandidateLength(): returns the number of alternatives
- getVoterLength(): returns the number of voters
- getTotalVotersVote(): returns the number of registered voters who voted
- getInformation(): depending on input alternative's id returns all information of them, only if period is "End"
- getCandidateName(): returns only the name of an alternative depending on the input id
- getNumOfIncompleteVotes(): returns the number of votes that are not "complete", i.e., do not approve exactly k=3 different candidates.
- getTotalOfAVScore(): returns the sum of AV-score for all the alternatives
- getWinnersbyId(): returns the ids of the winners
- isResults(): returns the results variable, indicating whether the voting results have been calculated



```
Aqua_Backend
  artifacts
  cache
  contracts
    Aqua.sol
  coverage
  node_modules
  scripts
    deploy.js
  test
    testsAqua.js
  .env
  .gitignore
  .prettierignore
  .prettierrc
  coverage.json
  gas-report.txt
  hardhat.config.js
  package.json
  yarn.lock
```

```
Aqua_Frontend
  img
  .prettierrc
  about.php
  Chart.js.2.5.0.Chart.min.js
  connectMetamask.js
  connectMetamask.php
  constants.js
  ethers-5.1.esm.min.js
  index.php
  init_voting.php
  init-voting.js
  map_aqua.php
  package.json
  restart.php
  results.js
  results.php
  u_index.php
  u_map_aqua.js
  u_map_aqua.php
  vote_voting.php
  vote-voting.js
  yarn.lock
```

Figure 9: Directories for back-end and front-end of Aqua

6. Conclusions & Future Work

In this paper, we propose a decentralized multi-winner voting system based on Ethereum Blockchain, which utilizes the AV rule. Our system, named Aqua, represents a novel integration of blockchain technology with multi-winner voting systems and has the capability to replace existing multi-winner voting systems and enhance the overall security of such elections. A decentralized multi-winner voting system based on Ethereum Blockchain, like Aqua, has the potential for future development in elections that utilize various approval rules, such as Approval Chamberlin-Courant, Greedy Monroe, and

Phragmén's Sequential Rule, among others. It can also be adapted for multi-winner elections that employ the ranking-based model, where voters provide as input complete rankings of the alternatives. Moreover, the user interface and the visualization of the results can be customized and adjusted to meet the specific needs of individuals or organizations.

References

[1] A. M. AL-MADANI, A. T. GAIKWAD,V. MAHALE and Z. A. T. AHMED. Decentralized E-voting system based on Smart Contract by using Blockchain Technology. In *Proceedings of the 2020 International Conference on Smart Innovations in Design, Environment, Management, Planning and Computing (ICSIDEMPC)*, Aurangabad, India, pp. 176-180, 2020.

[2] S. T. ALVI, M. N. UDDIN and L.ISLAM. Digital Voting: A Blockchain-based E-Voting System using Biohash and Smart Contract. In *Third International Conference on Smart Systems andInventive Technology (ICSSIT)*, Tirunelveli, India, pp. 228-233, 2020.

[3] A. M. ANTONOPOULOS and G. WOOD. *Mastering ethereum: building smart contracts and dapps*. O'reilly Media, 2018.

[4] H. AZIZ, M. BRILL, V. CONITZER, E. ELKIND, R. FREEMAN and T. WALSH. Justified representation in approval-based committee voting. *Social Choice and Welfare*, 48(2), 461–485, 2017.

[5] P. FALISZEWSKI, P. SKOWRON, A. M . SLINKO, and N. TALMON. Multiwinner voting: A new challenge for social choice theory. In U. Endriss (Ed.), *Trends in computational social choice*, AI access, pp. 27–47, 2017.

[6] F. FUSCO, M. ILARIA LUNESU, F. EROS PANI and A. PINNA. Crypto-voting, a blockchain based e-voting system. In *Proccedings of the 10th Int. Conf. on Knowledge Management and Information Sharing*, vol 3, pp 223-227, 2018.

[7] F. Þ. HJÁLMARSSON, G. K. HREIÐARSSON, M. HAMDAQA and G. HJÁLMTÝSSON. Blockchain-Based E-Voting System. In *Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, San Francisco, CA, USA, pp. 983-986, 2018.

[8] A. INDAPWAR, M. CHANDAK and A. JAIN. E-voting system using Blockchain technology. *Int. J. of Advanced Trends in Computer Science and Engineering*, 9, No.3, pp. 2020.

[9] D. M. KILGOUR. Approval balloting for multi-winner elections. In J. F. Laslier & M. R. Sanver (Eds.), *Handbook on approval voting* (pp. 105–124). Springer, 2010.

[10] M. LACKNER and P. SKOWRON. *Multi-winner voting with approval preferences*. Springer Nature, 2023.

[11] J. F. LASLIER and M. R. SANVER (Eds.). *Handbook on approval voting*. Springer, 2010.

[12] K. SADIA, M. MASUDUZZAMAN, R. KUMAR PAUL, and A. ISLAM. Blockchain based secured e-voting by using the assistance of smart contract. In *Proceedings of the Int. Conf. on Blockchain Technology (SpringerIETE)*, pp 161-176, 2019.

[13] J. UZMA, M. J. A. AZIZ, and Z. SHUKUR. Blockchain for electronic voting system—review and open research challenges. *Sensors*, 21, no. 17: 5874, 2021.

[14] J. WILSON. *Node.js 8 the Right Way : Practical, Server-Side JavaScript That Scales*. The Pragmatic Bookshelf, 2018.