Enhancing E-voting Data Security with Blockchain Technology

1.1 Introduction

The foundation of modern democracies is voting, whether via electronic voting (e-voting) or traditional ballot-based voting (Sheer Hardwick et al. 2018). Online voting, sometimes referred to as electronic voting or e-voting, is a computerized method of casting and tallying votes (Burton et al. 2016). Although electronic voting has been used in a number of nations for local elections, there is still some reluctance about it (Benabdallah et al. 2022). For a citizen who has no control over the voting mechanism, the procedure is still extremely difficult to check and audit, much like with paper voting (Benabdallah et al. 2022). Blockchain appears to be a promising means to address these issues.

Blockchain can be defined as a decentralized, distributed, and immutable ledger technology that is used to maintain a continuously growing list of records, called blocks (Qatawneh et al. 2020) (Al-Maaitah et al. 2021) which enhance data security, openness, and transparency (Febriyanto et al. 2020). It is a new technology, with untapped potential that is being used for an increasing number of applications (Benabdallah et al. 2022).

1.2 Motivation​​

The widespread use of electronic voting technologies has changed the electoral landscape by accelerating the procedure and improving voter accessibility. Nevertheless, there are serious security flaws in these systems that compromise the integrity and validity of election results (Zhang et al. 2019). There are significant risks associated with being vulnerable to cyber-attacks, data manipulation, and hacking, which could jeopardize democratic processes at their core (Al-Maaitah et al. 2021). In the absence of strong security protocols, the consequences of these weaknesses may result in reduced confidence in electoral processes, disputed election outcomes, and undermined democratic bases.

1.3 Contribution

My contribution presents an innovative blockchain-based model that is specifically intended to strengthen data security and integrity, hence addressing the basic issues that are present in modern E-voting systems. Through the careful integration of blockchain technology, my model radically changes the view of E-voting security. By incorporating immutability, transparency, and decentralised consensus methods into the heart of electronic voting systems, this work seeks to strengthen election processes. The application of smart contracts within this infrastructure solidifies the authentication process, establishing a resilient defense against potential cyber threats and unauthorized access.

1.4 Paper Structure

This paper explores the innovative use of blockchain technology to address the inherent data security challenges of E-voting. This paper is organized as follows. We talk about the background of the study and related works in section II. In section III, we develop the proposed blockchain solution while in section IV, we discuss a thorough evaluation of the proposed blockchain model. The last section concludes the paper and summarizes key findings alongside future considerations for this solution.

2. Background and Related Work:

This section gives an overview of existing research and literature related to e-voting and blockchain technology. The papers reviewed are from IEEE Xplore, Springer, Scopus and Google Scholar where the search query included “Data Security with Blockchain Technology”, “Enhancing Data Security in E-Voting” and “E-Voting with Blockchain”.

2.1 Blockchain Concept and History

The blockchain structure is a form of append-only data structure (Malhotra et al. 2022). Taş and Tanrıöver (2020) define it as a sequence of time-stamped blocks linked by cryptographic hashes. The chain is constantly growing as more blocks are added, and each new block preserves the hash of the data from the previous block (Kamil et al. 2021). By integrating blockchain technology into the database distribution process of an electronic voting system, fraud and database manipulation can be minimised (Kamil et al., 2021). (Benabdallah et al. 2022).

In 1991, Haber and Stornetta first put up the idea of blockchain (Taş and Tanrıöver 2020). It is believed that Satoshi Nakamoto created the first blockchain-based system in 2008. In order to address the double-spend issue, Nakamoto proposed blockchain as the foundation of the Bitcoin system (Zhang et al. 2019). It's also evident that Bitcoin was the first major application of blockchain technology. An open, safe data book that is distributed globally can be compared to the idea of blockchain (Ashish Singh Parihar et al. 2021). Thus, the idea can be applied to a wide range of industries where transactions are involved, not just the cryptocurrency and banking sectors (Taş and Tanrıöver 2020). Blockchain 2.0 is born from the ability to make decentralized applications with smart contracts (Szabo 1997) (Buterin 2014). Blockchain is evolving toward 3.0 as technology improves and decentralized applications are developed (Benabdallah et al. 2022).

2.2 Related Works

The authors of this study (Cheema et al. 2020) suggested an electronic voting system linked to a public blockchain to hold city citizen records and determine a voter's eligibility to cast a ballot at a specific polling place. Voters must complete the registration process by providing their personal information. Their data is then hashed and sent back to the primary data centre for authentication. If the hashes match, the voter will be allowed to cast a ballot on election day. According to Singh et al. (2022), this system guards against intrusion into the information exchange network and provides transparency and confidence.

According to the paper by Kusters et al. (2012), they introduce the term "verifiability," which means that users of the electronic voting system, or voters, have the authority to verify that their votes were cast, counted, and that there was no vote manipulation. The "Wombat Voting System" and a variation of the "Helios Voting System" are the tested systems that Kusters et al. (2012) mention.

In their work, Hjalmarsson et al. (2018) suggested an electronic voting system built on the blockchain concept, which essentially makes use of smart contracts to enable secure, affordable elections while protecting voters' anonymity. The authors of the study have demonstrated how blockchain technology can assist lower adoption obstacles and constraints associated with electronic voting systems, hence ensuring election security, integrity, and transparency. With the help of private Ethereum blockchain, there is a chance to transmit several transactions per second towards the blockchain technology, ensuring that every component of the smart contract to reduce the burden on the blockchain network (Hjalmarsson et al. 2018).

Yi's paper, "Securing e-voting based on blockchain in P2P network," introduces novel enhancements to blockchain technology aimed at fortifying the security of electronic voting (e-voting) systems (Yi 2019). The paper suggests improvements in three main areas: first, a user credential model that uses Elliptic Curve Cryptography (ECC) for user authentication and non-repudiation; second, a synchronised voting records model built on Distributed Ledger Technology (DLT) to prevent vote forgery; and third, a withdrawal model that allows voters to change their votes before a specified deadline. Together, these developments provide a comprehensive blockchain-based electronic voting scheme that satisfies essential requirements for electronic voting. Most significantly, the plan decreases voters' worries about data security and removes the possibility of limitless reproduction in electronic voting, which increases confidence in the dependability and security of the system (Yi 2019).

In a 2016 technical study, Kyoto University researchers presented a unique electronic voting method that provides a means to use the coin transfer mechanism to achieve a "vote transfer" mechanism instead. An official coin wallet is given to each and every registered voter, including candidates. Voters are given coins by the electoral administration to use during the voting process. Then, on election day, voters briefly transfer their coin to the candidate of their choice, much like they would with any other coin trade or transfer. In addition, there is a middle unit situated in between the candidates and the voters that resumes or finishes the transfer by converting the transferred coins to a different coin-currency. So that, it is no more possible to trace the owner of any vote and this is done to protect privacy and anonymity of the voters (Çabuk et al. 2018).

3. Proposed Solution

3.1 System Design

Finding out the e-voting demands based on research and how they can be fit into the system as features is the goal of requirement analysis (especially in the case of functional requirements). A new proposed e-voting system should match different requirements; functional and non-functional (Alvi et al. 2020).

3.1.1 Functional Requirements

Functional requirements are the features that a system must have to function as expected. They differ from non-functional requirements because they are more specific. The functional requirements of this system are listed below:

1. Valid users (eligible voters) must be able to log in to the system.
2. System administrators (electoral commission staff) must be able to create elections.
3. System administrators must be able to add candidates to an election.
4. Users must be able to view candidates for an election.
5. Users must be able to vote only once.
6. A user must be able to vote for their preferred candidate in an election.
7. Users must be able to verify that their votes have been cast.
8. The system must be able to count and collate election results as the case may be.
9. Users must be able to view election results.

## 3.1.2 Non-Functional Requirements

## Non-functional requirements limit how the system accomplishes its goals (performs its functional requirements). The following are non-functional requirements of the system:

1. The voter information must be anonymous, unlinked, and untraceable from the votes cast. This is the principle of voter anonymity and secrecy.
2. Administrators cannot cancel or edit candidates in the middle of an election after it has started (electoral officers).
3. The mechanism must ensure that the voter has no way of proving that he voted for a specific candidate. This would reduce voter coercion or buying.
4. After casting votes, the system must not allow them to be modified or manipulated.
5. Controlling the system should be impossible for any organization (including the electoral commission) or individual.
6. After the election, the system must be able to audit or validate votes.
7. Elections must be precise regarding the time frame and the candidates competing.

The suggested system design has three primary layers: the blockchain, the smart contract and the user access control layer. The suggested solution uses a public blockchain, which is essential to mention. This means that anyone can be a node; thus, no trust is required among the blockchain's participants (nodes).

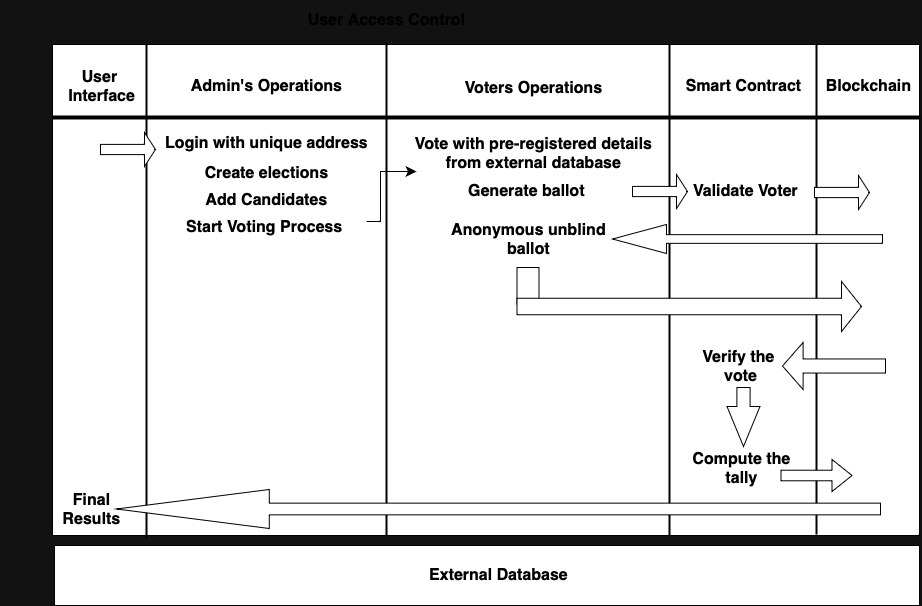


Figure 1. Pictorial representation of the system design of the proposed model.

The suggested approach does not require every registered voter to be a member of the blockchain. There is a separation between nodes and participants' devices. Similarly, not every node must be employed as a voting device.

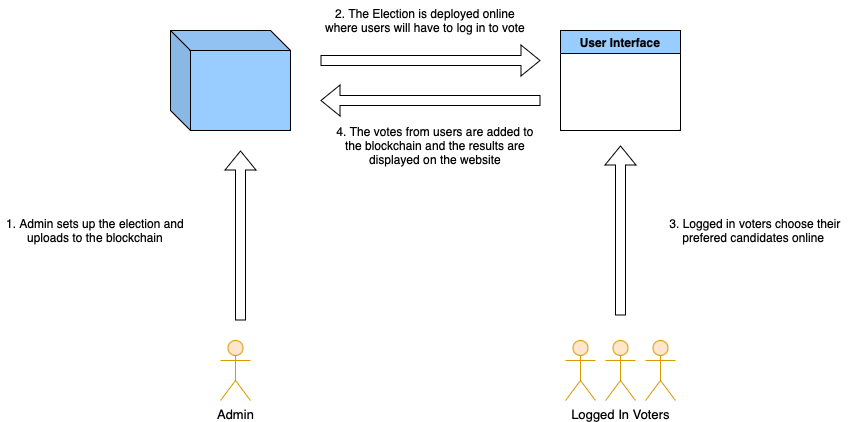


Figure 2. Pictorial representation of the concept of the proposed model.

3.2 System Components

This section refers to the components that make up the e-voting system;

3.2.1 Smart Contracts: The cornerstone of the system lies in the Ethereum smart contract - the `EVoting` contract. This contract encapsulates the rules and logic governing the election process. It includes functionalities such as candidate registration, voter participation, and user registration.

* EVoting Contract: Manages candidate information, voter statuses and admin records.
* Structures and Mappings: Define robust data structures to store candidate details, voter statuses and election statuses.
* Functions: Enable actions like adding candidates, registering voters, casting votes, managing votes, and starting, ending and creating the election.

3.2.2 Nodes**:** Nodes within the Ethereum network maintain the integrity and operation of the blockchain. These nodes contribute to validating transactions, executing smart contracts, and maintaining the decentralized network.

* Full Nodes: Store complete copies of the blockchain, participate in validating transactions, and ensure consensus.
* Mining Nodes: Engage in the process of block creation (mining) by solving complex cryptographic puzzles.

3.2.3 Blockchain: The distributed and immutable ledger holds transactional data and the state of the election process.

* Transaction History: Records actions such as candidate additions, voter participation, and admin actions.
* Smart Contract State: Stores real-time information on candidates, voter statuses, admin inputs and user registrations.

### 3.2.4 User Interface: A user-friendly interface is crucial for user engagement and participation in the voting process.

### Results Visualization: Presents election outcomes in an easily understandable format.

3.2.5 External Database: The external database serves as a repository for off-chain data storage.

* Sensitive Data Storage: An external database can store supplementary information, such as user profiles and authentication details.
* An external database can alleviate these concerns of robustness and scalability by storing non-critical or large-scale data off-chain, reducing the strain on the blockchain network.

3.3 Blockchain Structure

3.3.1 Ethereum Blockchain Architecture

Ethereum is the foundation of the electronic voting system; it is a decentralized platform that makes smart contract execution possible. The testing environment is a personal Ethereum development blockchain called Ganache. Every node in the Ethereum blockchain structure has a copy of the whole blockchain ledger. These nodes are locally mimicked within Ganache, allowing for quick testing and development cycles without requiring a lot of processing power.

3.3.2 Block and Transactional Framework

A transaction is any piece of data saved on the blockchain; in this example, the transactions are votes. The electronic voting system's transactions are included in blocks, each of which is cryptographically connected to the one before it to create an unchangeable chain. These blocks form a safe and impenetrable structure because they include hashed references to earlier blocks. These blocks are effectively generated, verified, and mined in the virtual environment using Ganache, simulating the blockchain in the real world.

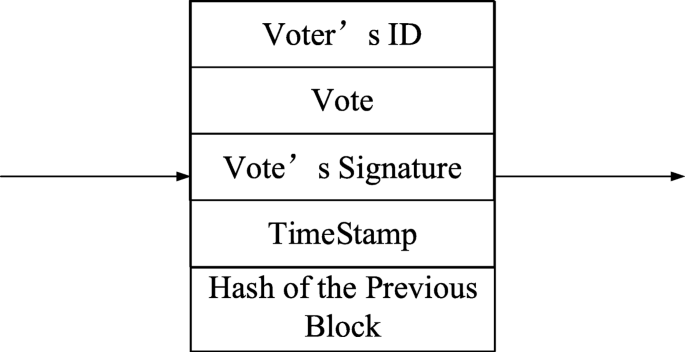


Figure 3. Pictorial representation of the block. Adapted from (Yi 2019).

3.3.3 Role of Ganache in System Development

Ganache plays a pivotal role in the rapid prototyping and testing of the proposed system. The platform's intuitive UI and local simulation functionalities facilitate the deployment, execution, and debugging of smart contracts with ease. Furthermore, Ganache offers crucial functions like transaction tracing, gas management, and fast block formation that allow the emulation of a genuine blockchain environment while keeping a developer-centric focus.

3.3.4 Smart Contracts and Consensus Integration

Smart contracts, which are self-executing Ethereum-based contracts that automate the voting process, are the brains behind the proposed electronic voting system. It is possible to create, implement, and communicate with these smart contracts in a regulated setting with the help of Ganache's integration. Ganache simulates the Proof of Work consensus mechanism, which verifies transactions and maintains the integrity of the voting process.

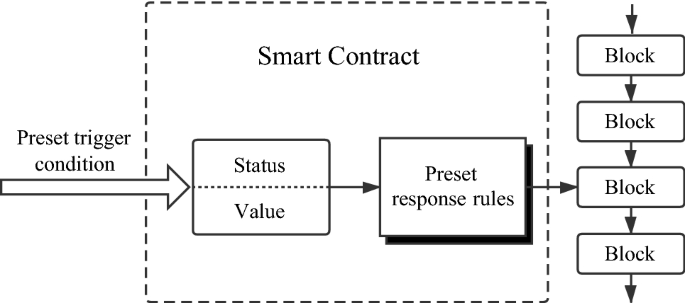


Figure 4. Pictorial representation of a smart contract. Adapted from (Zhang et al. 2019).

3.4 Consensus Protocol

3.4.1 Implementation of Proof of Work:

At the core of the proposed e-voting system's consensus mechanism lies the utilization of Proof of Work. PoW requires participating nodes to perform computationally demanding activities, or mining, within Ganache's virtual environment. Completing intricate mathematical puzzles is part of this procedure, which verifies transactions and secures the network. When a node solves a puzzle successfully, other nodes must reach consensus to validate the proposed transactions before it may add a new block to the blockchain. By using PoW, the Ganache environment provides a strong degree of network security, making it more resistant to malicious activity and unauthorized changes to the voting records.

3.4.2 Traceability of Selected Proof of Work:

The integration of PoW within Ganache not only fortifies the security of the proposed e-voting system but also aligns seamlessly with the implemented smart contracts. These Solidity-scripted contracts enforce transparency and immutability by regulating the procedures and rules of the voting process. In real-world applications, PoW's energy consumption and scalability issues must be taken into account despite its benefits. Beyond deployment, Ganache plays a critical role as a platform for rigorous testing, debugging, and iterative improvement of the smart contracts and consensus protocol. The efficiency, security, and dependability of the proposed blockchain-based electronic voting system are guaranteed by this iterative procedure.

3.5 How the system works

* User Registration and Authentication:

The voting process initiates with user registration, overseen by the electoral commission. Eligible voters are provided with unique credentials and verified to ensure their validity. Upon receiving their credentials, users authenticate themselves by logging in to access the voting platform securely.

* Election Initialization and Candidate Registration:

The candidate number, party symbol and other essential information will be added to the blockchain before the commencement of the election. Authorized administrators initialize elections within the system. Once elections are created, administrators add candidates contesting in the election, detailing their profiles and affiliations.

* Voter Ballot and Voting Process:

Voters who have registered to vote use their login credentials to view the ballot portal, which shows the list of candidates. After choosing their favorite candidate, voters confirm their selection, and the vote is encrypted to protect voter identity.

* Vote Tabulation and Result Publication:

Following the voting phase, the system uses smart contracts to collect the encrypted votes and count them according to predetermined rules. To ensure accuracy, the system counts and decrypts the votes after the voting window closes. The election's outcome is displayed in the finalised results, which are released.

* Audit and Validation of Votes:

The technology creates an audit trail after the election to confirm the integrity and correctness of the voting procedure. Election results can be independently verified by users, stakeholders, and electoral officials to make sure they match the number of votes cast.

* Governance and System Maintenance:

The platform's performance can be maintained and tracked by authorised people thanks to the system design's support for administrative functions.

* Transaction and Data Security Measures:

Strict security protocols are followed during the entire process. The unchangeability of the vote records is ensured via smart contracts and blockchain immutability, while encrypted communication channels guarantee data integrity.

4. Evaluation:

This section delves into the critical evaluation of the proposed e-voting model, aiming to scrutinize its efficacy, robustness, and practical implementation.

4.1 Theoretical Evaluation

The proposed e-voting system revolves around a meticulous integration of blockchain technology, specifically utilizing Ethereum-based smart contracts, to address the critical challenges of data security in electronic voting. Evaluating this model reveals its inherent strengths in several dimensions essential for a secure, transparent, and trustworthy voting process.

Below are issues in some of the areas which the use of smart contracts and blockchain helps solve in E-voting:

Security:

* Access Control: The contract uses a restricted modifier to limit certain functions to the contract admins. However, further access control mechanisms, like role-based permissions, could enhance security.
* Secure Data Handling: Sensitive data such as private keys are hidden from other users accounts.
* Secure Voting Process: The vote function ensures that a voter can only vote once by checking voters[msg.sender]. This is a good step toward preventing multiple votes.

Scalability:

* The system can be scalable when the blockchain system is made available online, thereby introducing larger system resources for scalability concerns.

Trust:

* The decentralized and transparent nature of the blockchain reinforces trust in the voting process, providing a tamper-resistant record of votes that can be publicly verified.

Transparency:

* Public Visibility: Key data like candidate information and voting results are publicly accessible through mapped functions like candidates and voters.
* Immutable Record: The blockchain's immutability ensures that once data is added, it cannot be altered, promoting transparency and auditability.

Functionality:

* Candidate and User Management: The contract manages candidates and users effectively.
* Event Logging: The votedEvent event logs votes, enabling tracking and transparency.
* Error Handling: The contract uses require statements for certain conditions (e.g., preventing multiple votes.

The simplicity of the contract allows for relatively fast transaction times, even though it doesn't naturally address scalability problems. Although scalability might become an issue as the user base grows, the system's architecture can handle a moderate load without compromising speed.

The interoperability of the model is good inside its ecosystem. To ensure a seamless integration, additional skills could be needed in order to interact with blockchains or other systems.

One important element is privacy, which the approach manages reasonably enough. As passwords and other sensitive data are encrypted for further security, the model generally ensures users' privacy.

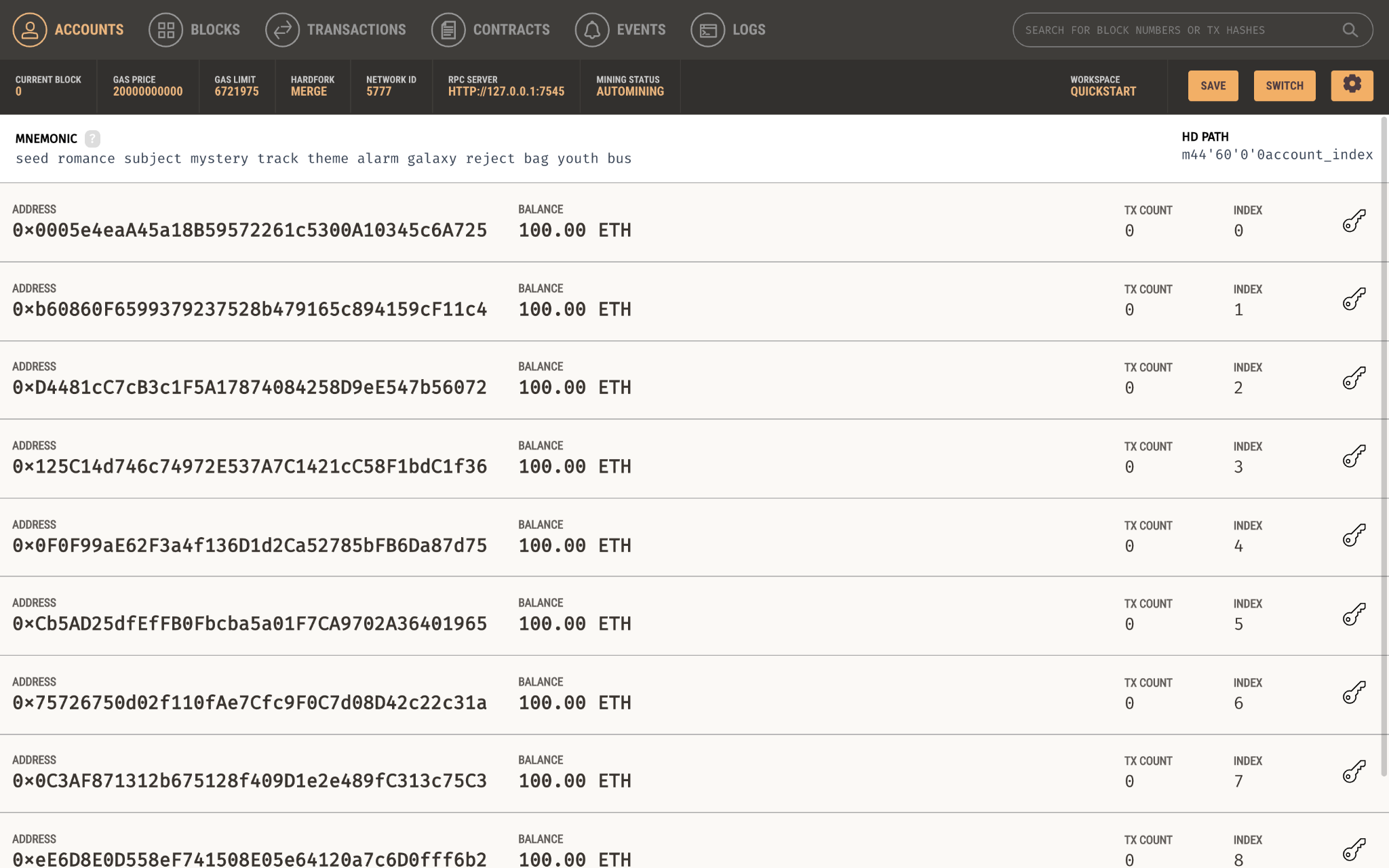
The blockchain's immutability gives the system a strong layer of trust in terms of trustworthiness. Since data cannot be changed once it is recorded, it encourages accountability and openness. This is especially important in situations such as elections, where voting process integrity is critical.

4.2.1 Technical Evaluation

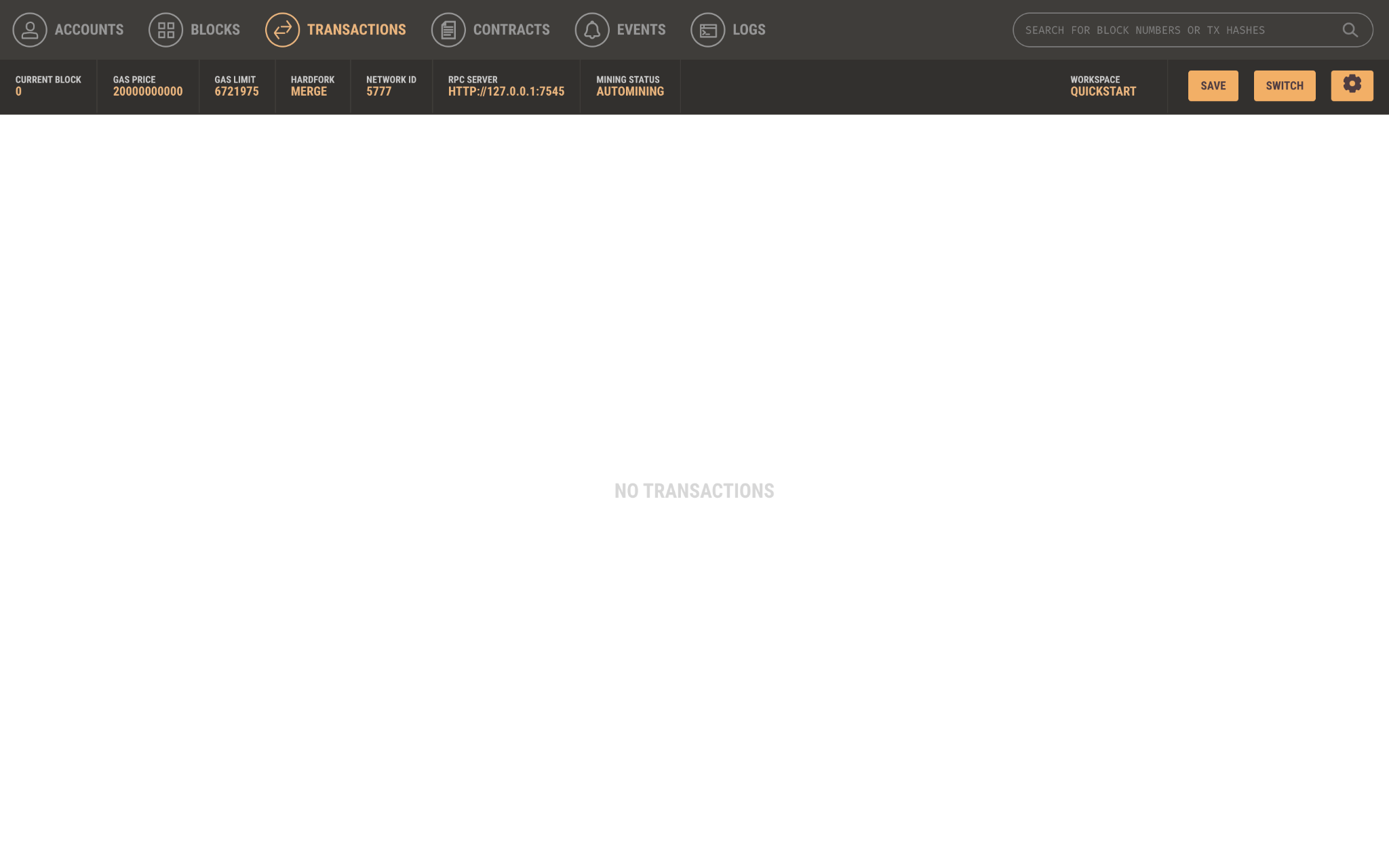
The proposed model would require certain things to be able to carry out implementation which are the following;

1. Ganache: This is the software which was used to create and manage the local blockchain ethereum network and will be used for testing purposes.
2. Remix IDE: This is the ideal environment for compiling and deploying solidity smart contracts.
3. The External Database but in this situation where the database of an electoral commission would not be available and for test purposes, we would be using a test data that has been embedded into the code for user voting. Test Data: “Last Name: Akinlua”, “Special ID: 9068340768”.

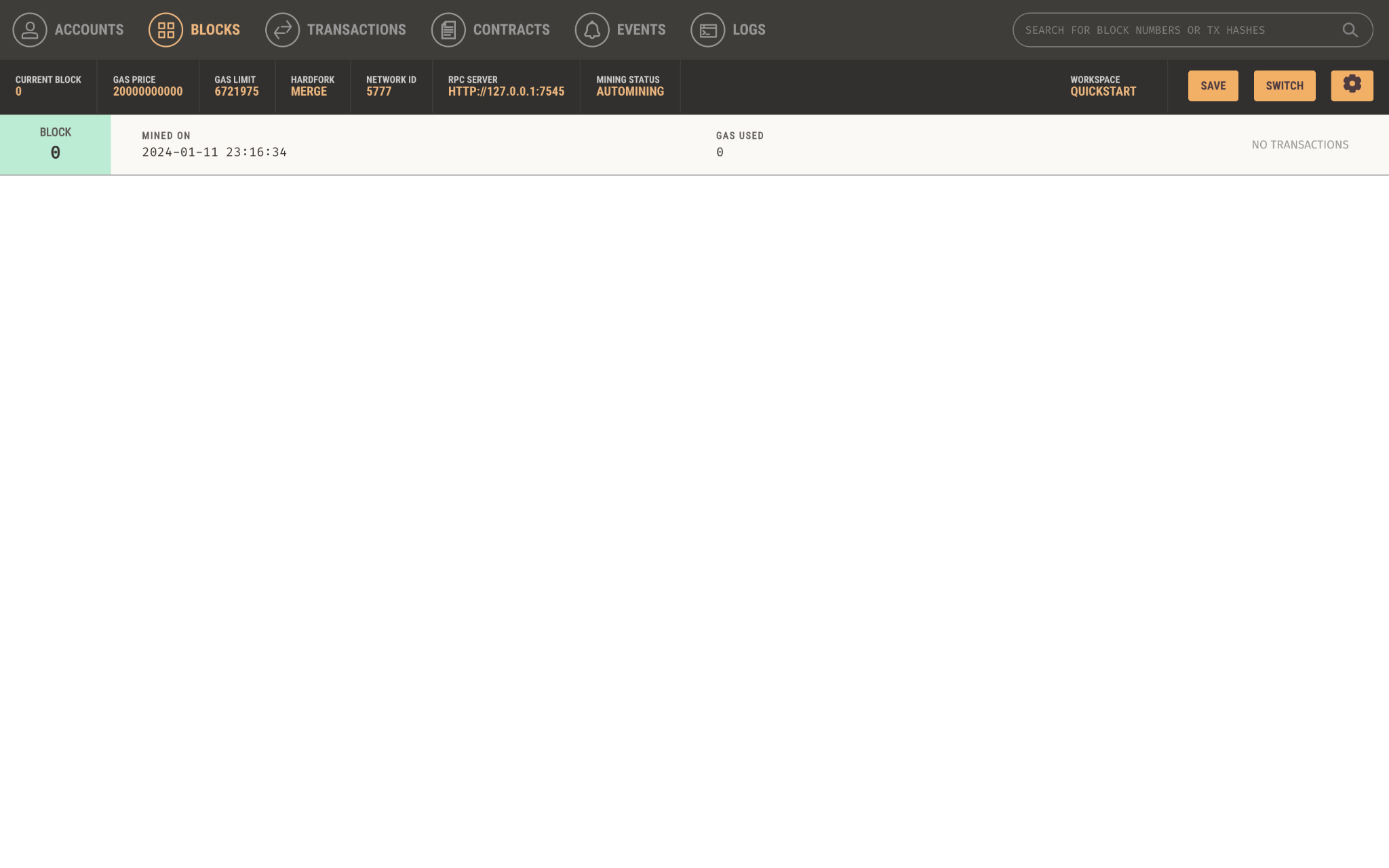
4.2.2 Diagrammatic Representations of Implementation



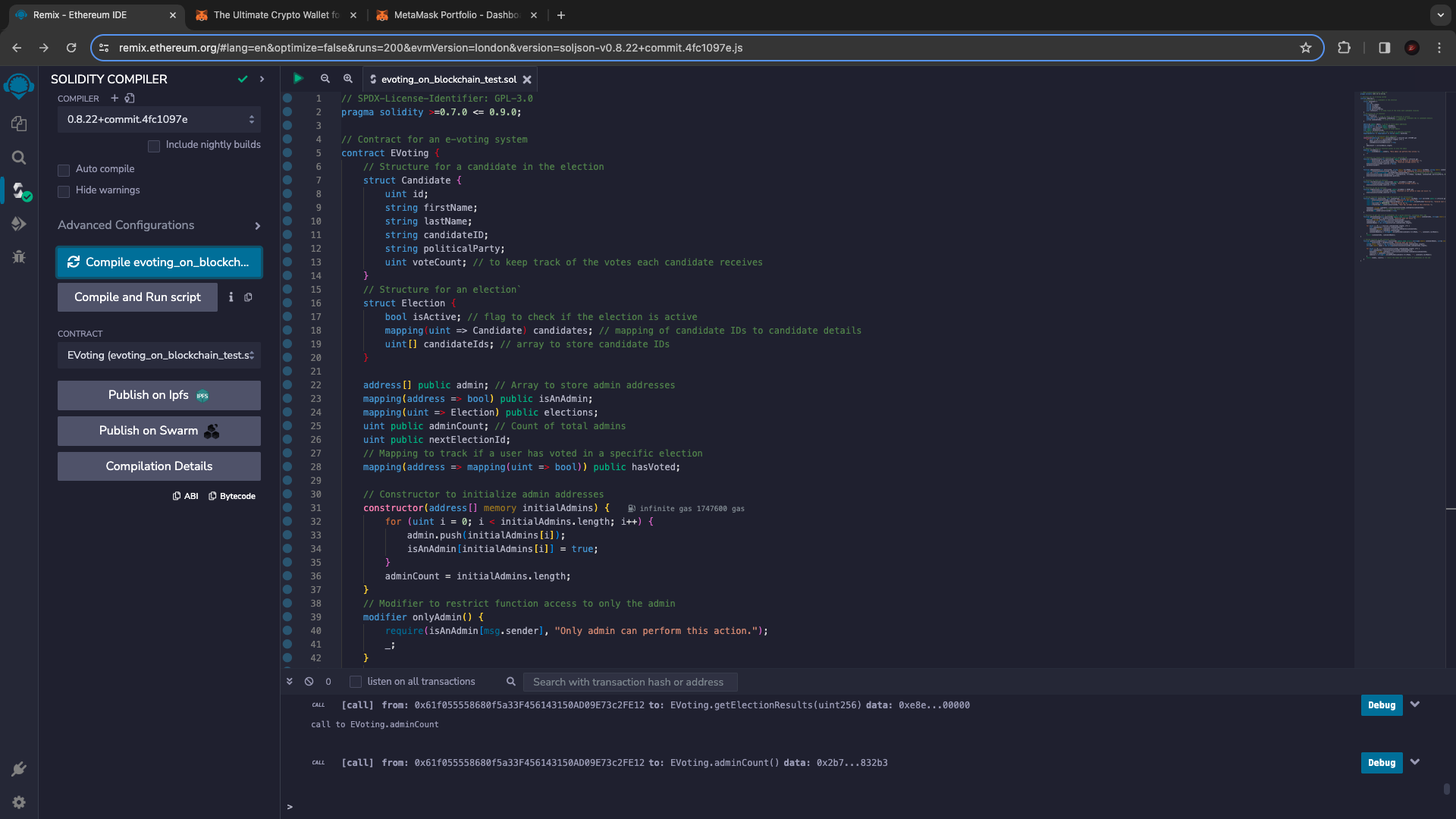
The Accounts Page of the Ganache environment with 10 accounts generated for use during deployment.



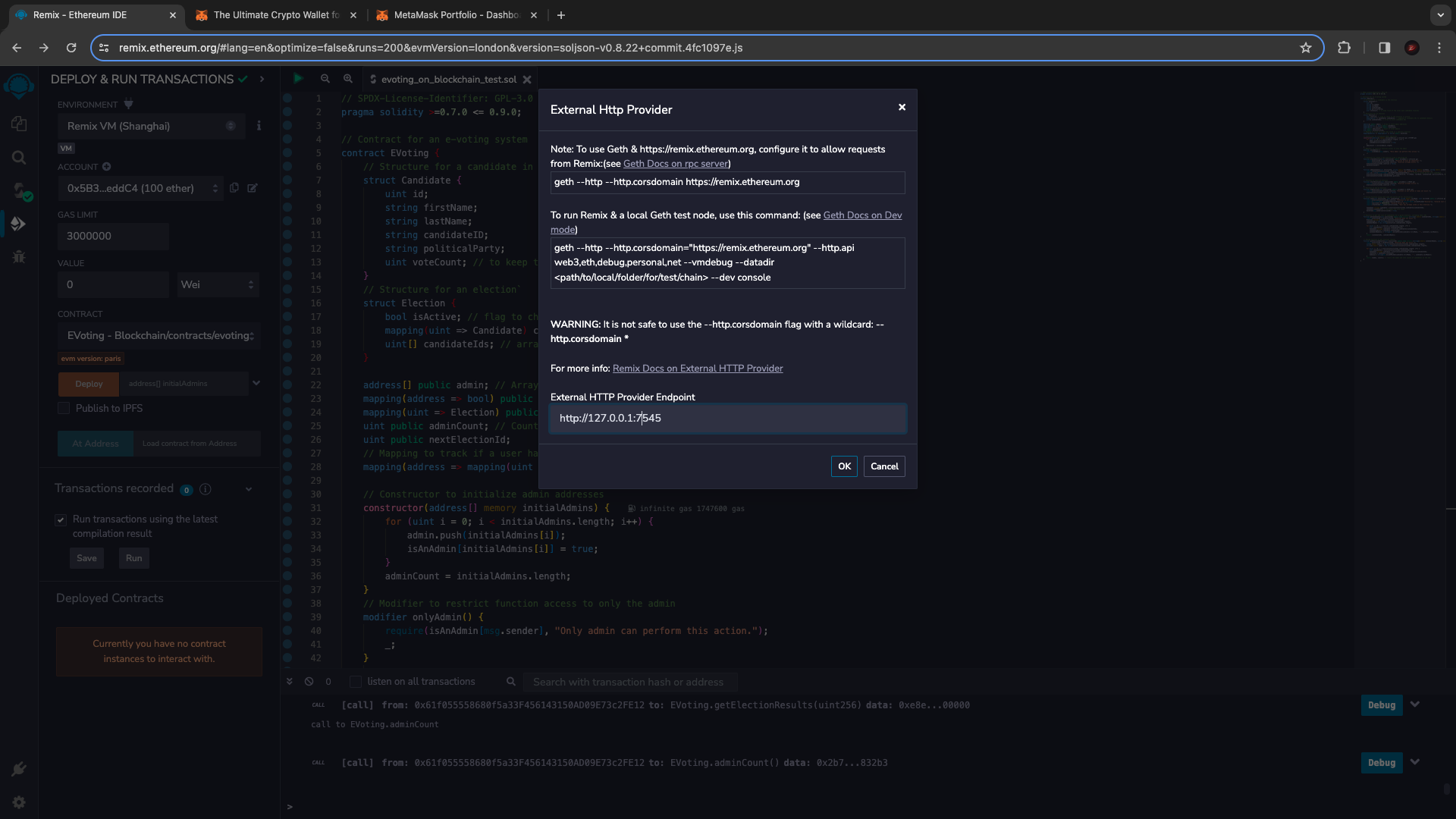
The Transactions Page of the Ganache environment to show transactions when the smart contract has been deployed.



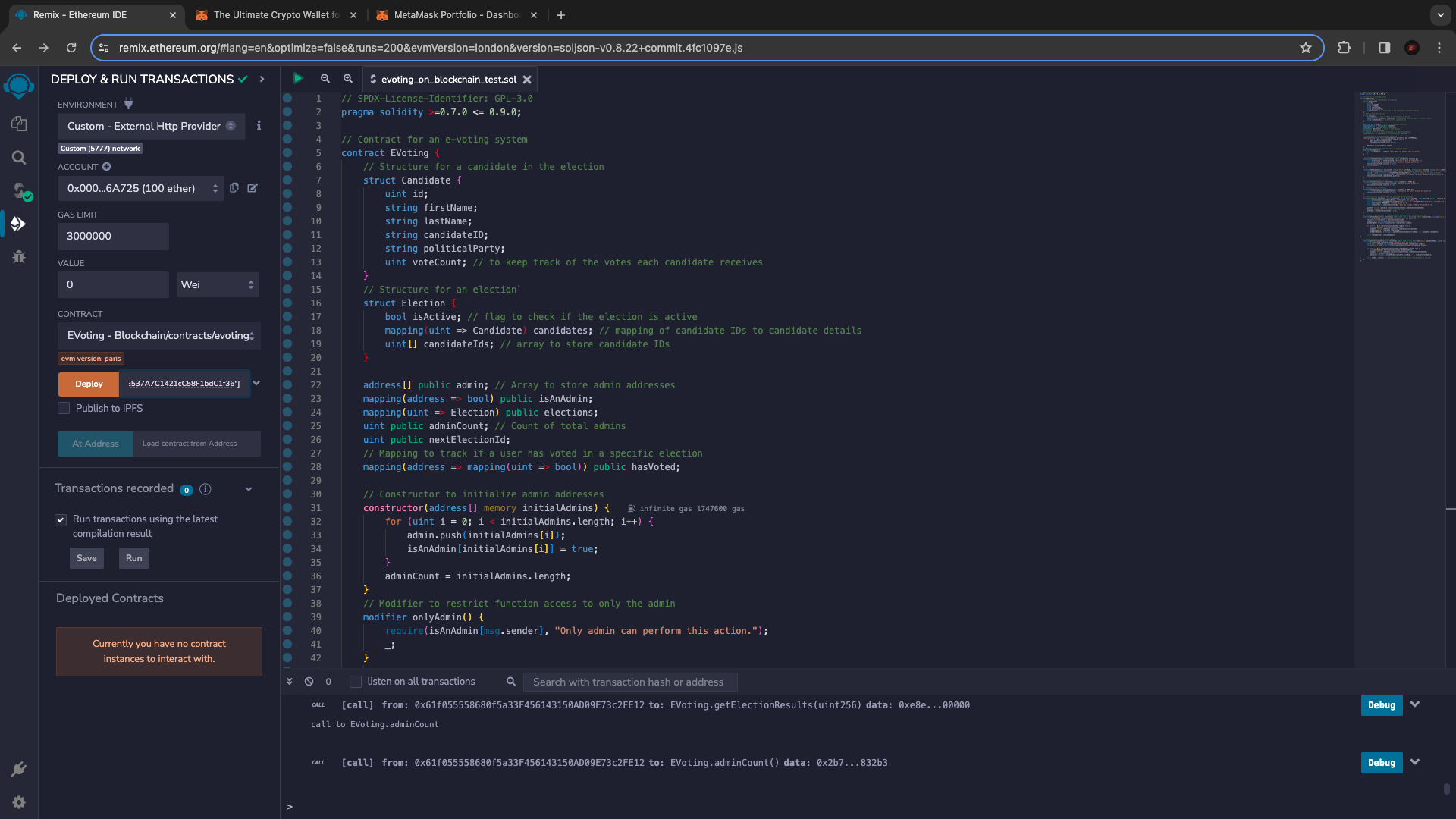
The Blocks Page of the Ganache environment to show creation of blocks when the smart contract has been deployed.



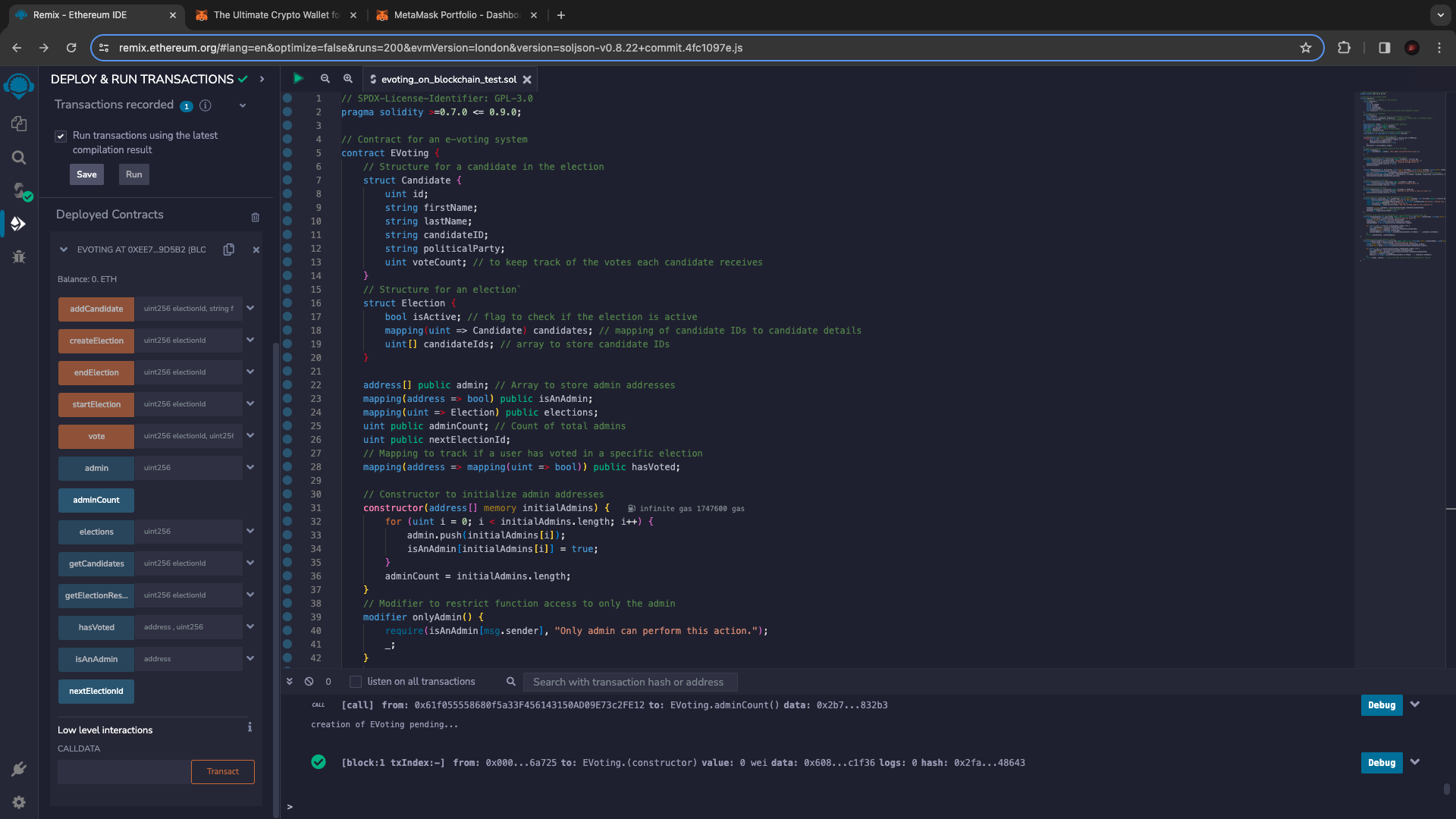
The e-voting smart contract compiled on the Remix environment and ready to be deployed to the Blockchain.



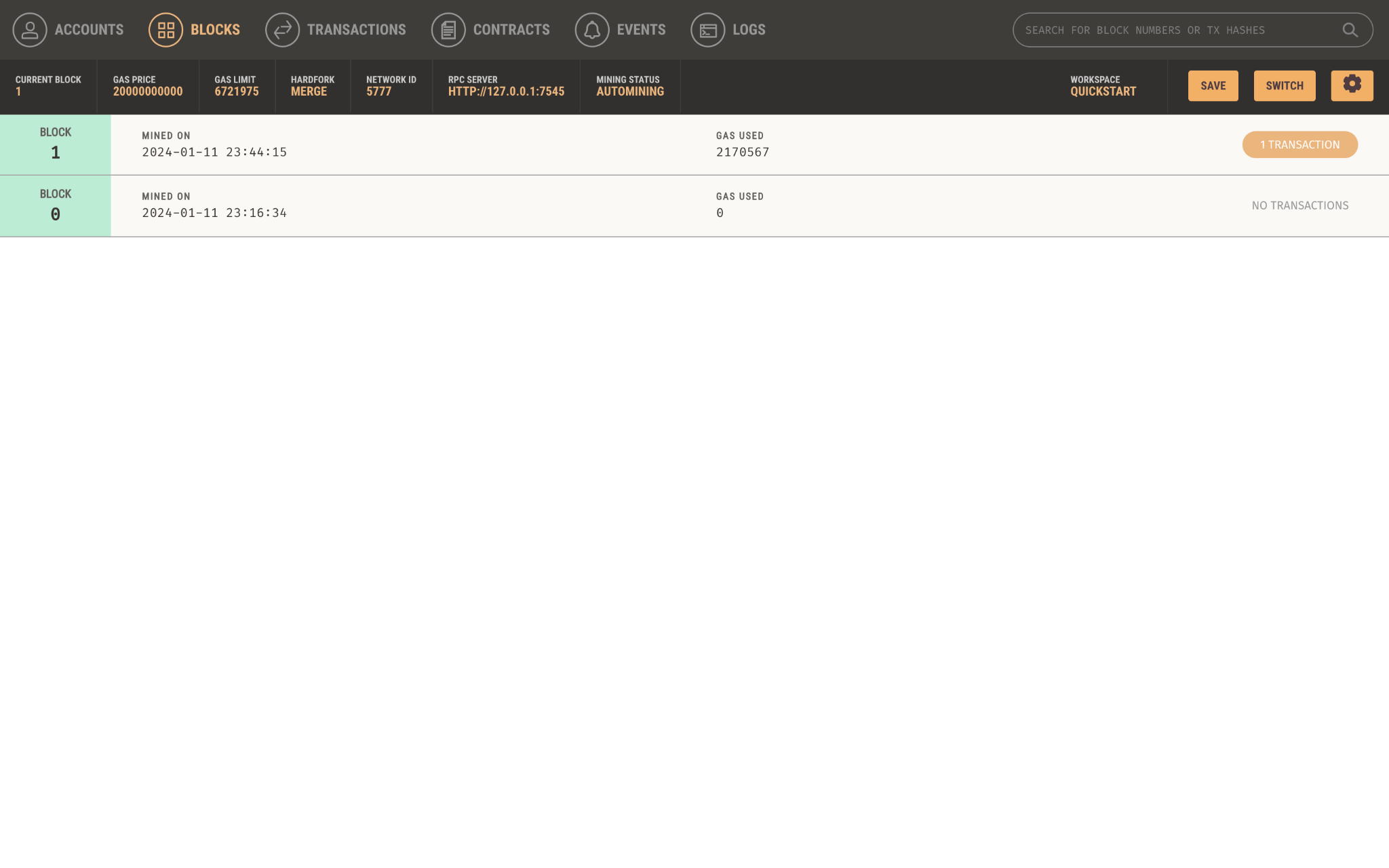
The process of connecting to the right external http provider endpoint to link Remix to the Ganache server.



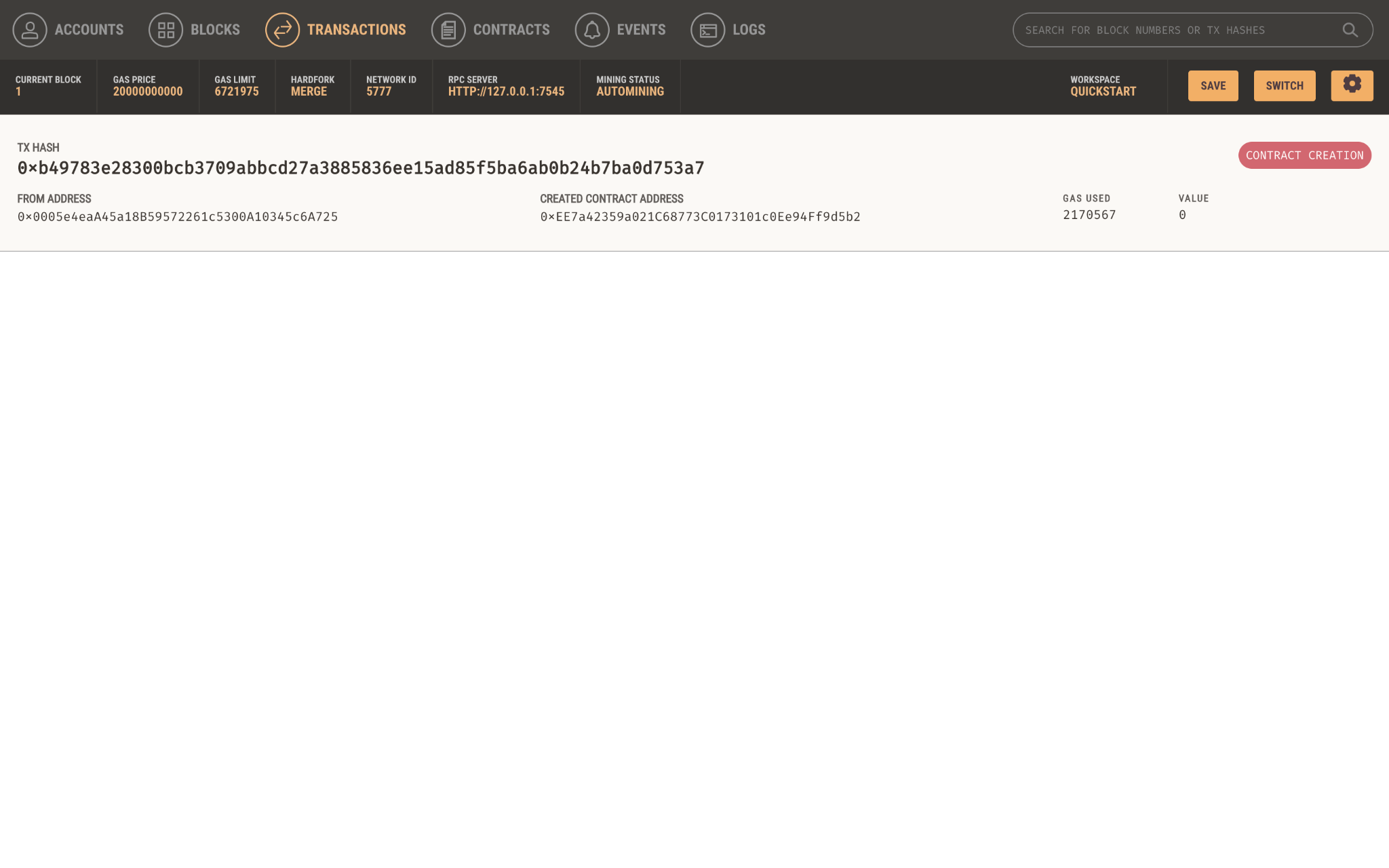
To deploy the smart contract, you need to enter the addresses of the admins of the election so only those accounts caan have admin rights.



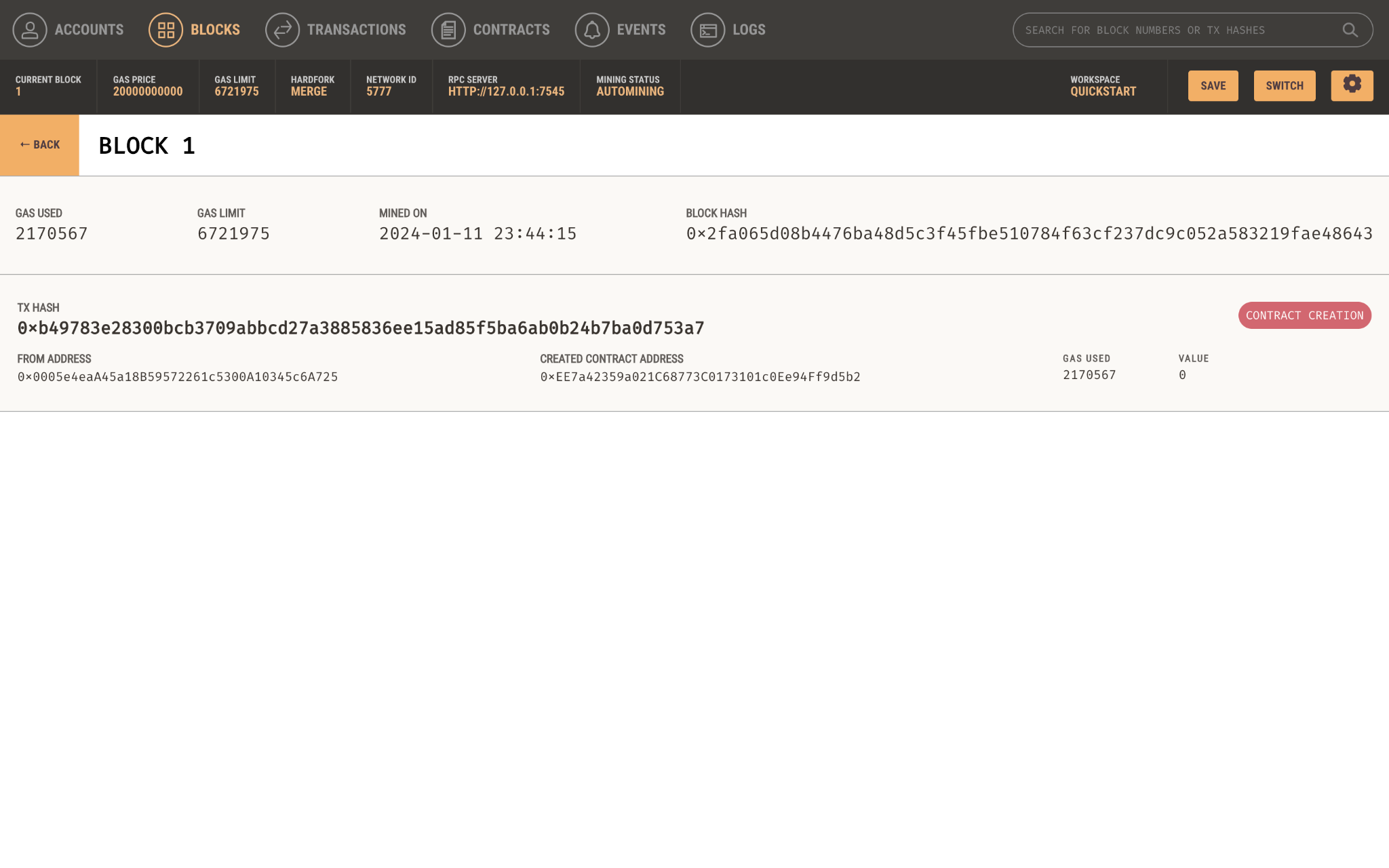
The smart contract has successfully been deployed on the blockchain as seen on the left hand side of the screen.



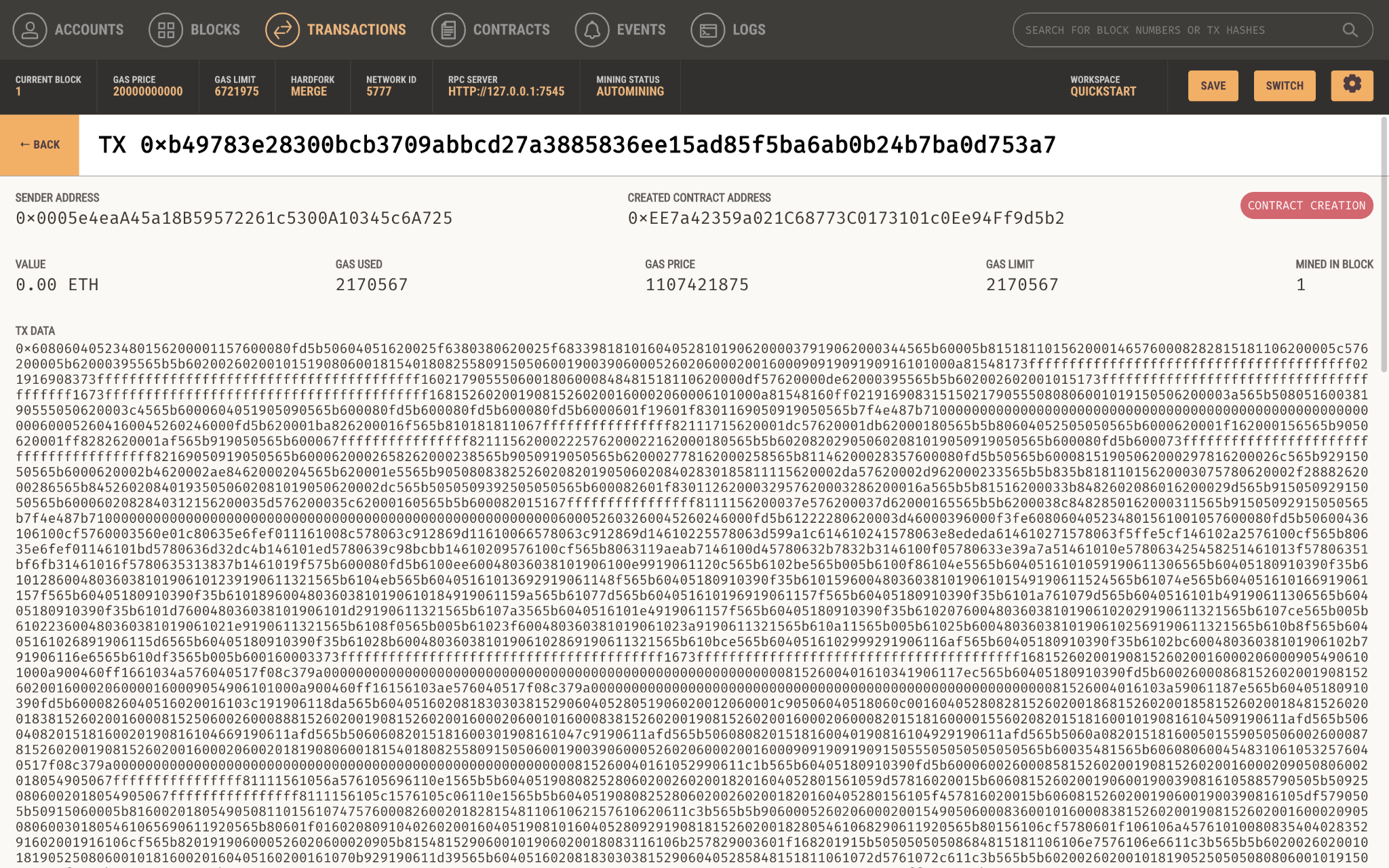
The genesis block has successfully been created on the network.



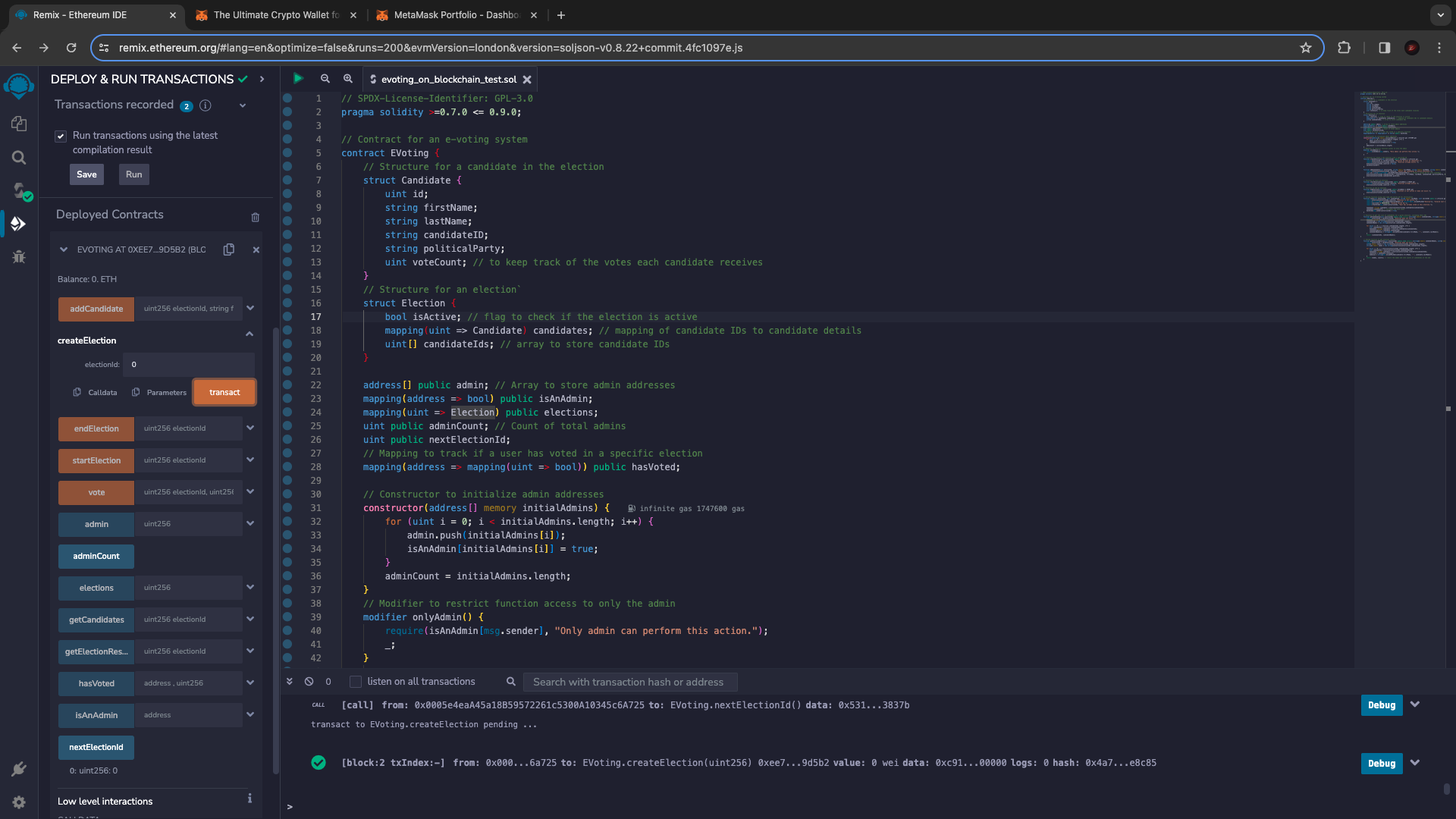
The first transaction has also been recorded on the network.



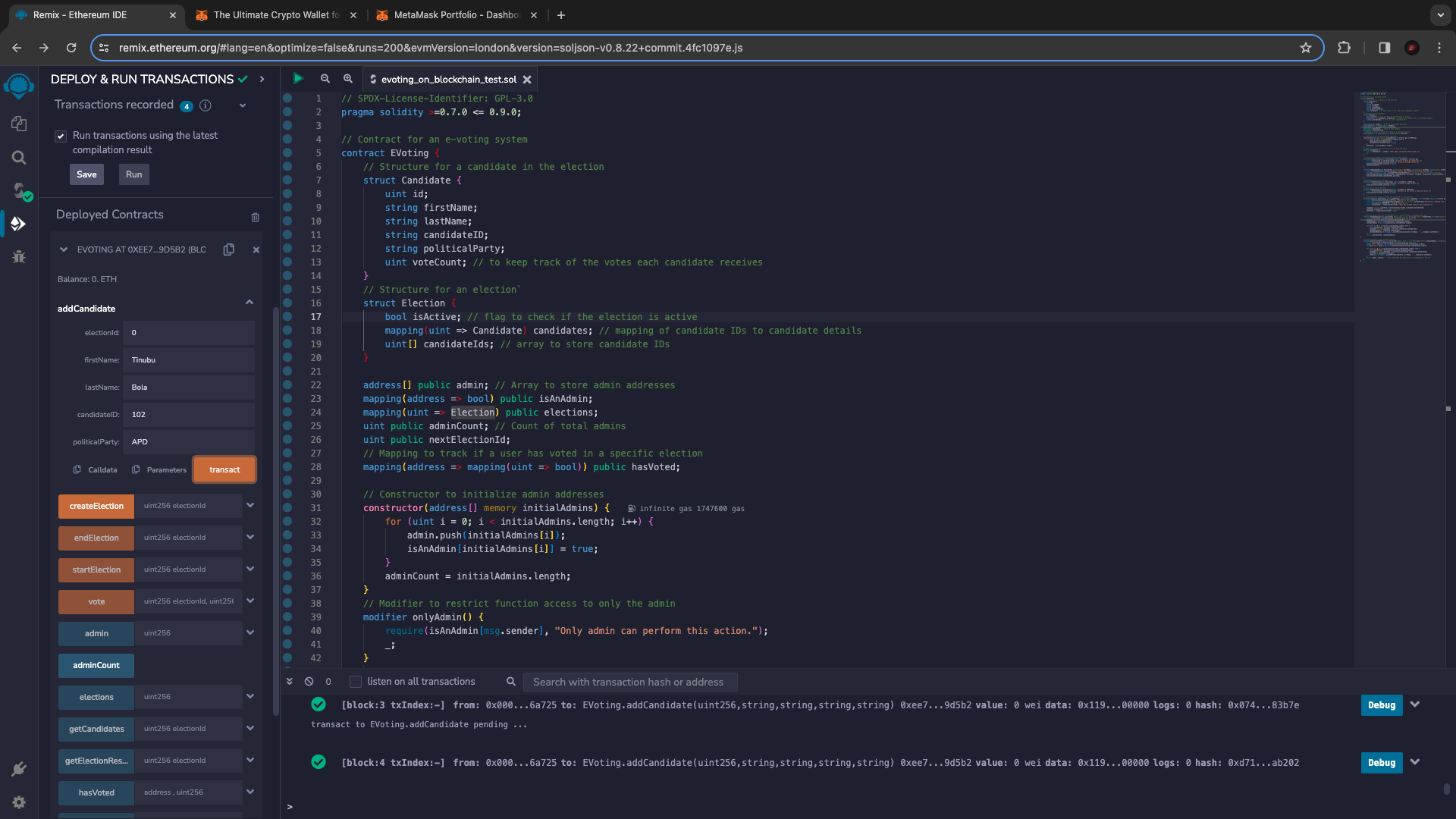
More insight to show genesis block creation.



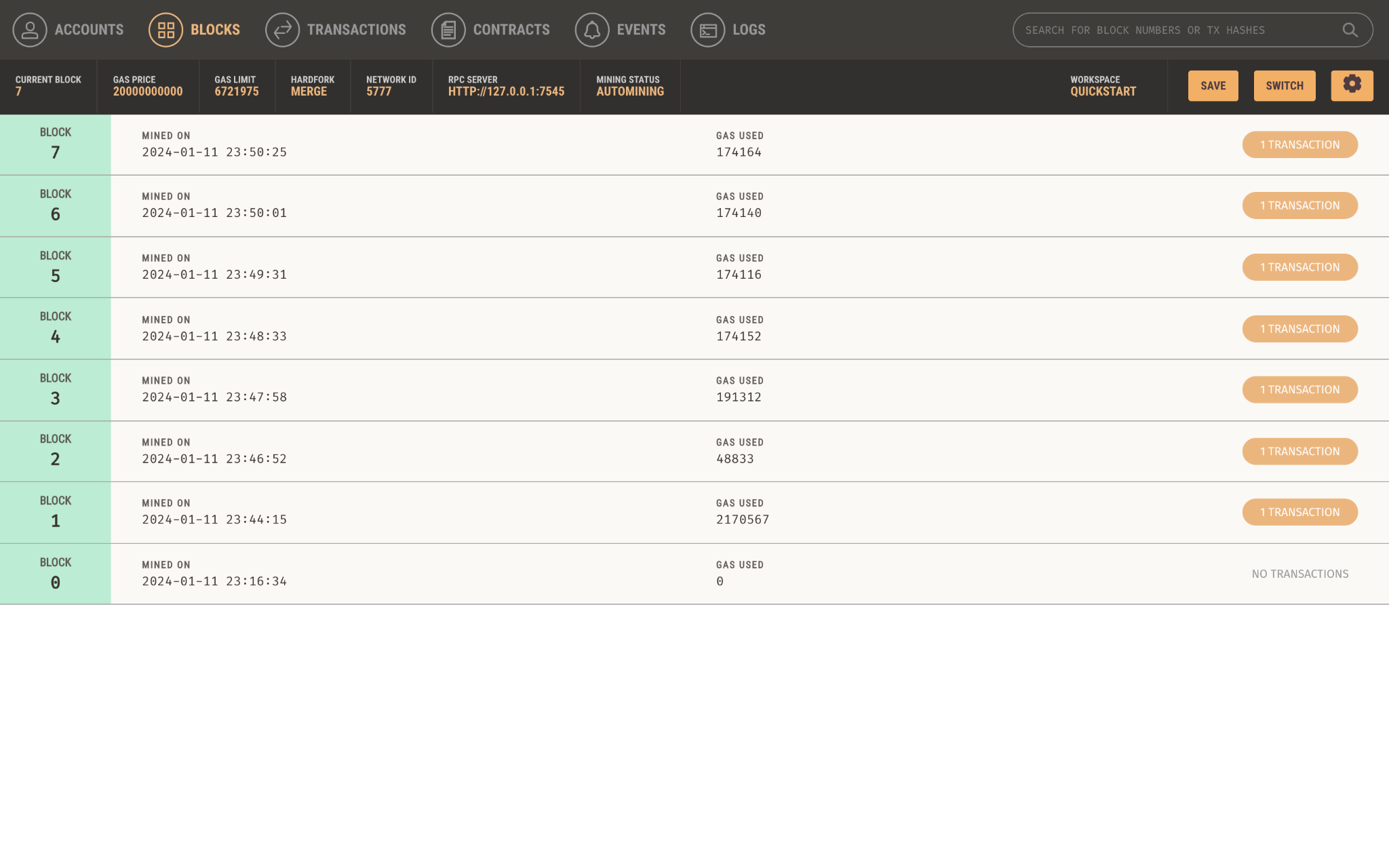
More insight to show the transaction details of the genesis block creation.



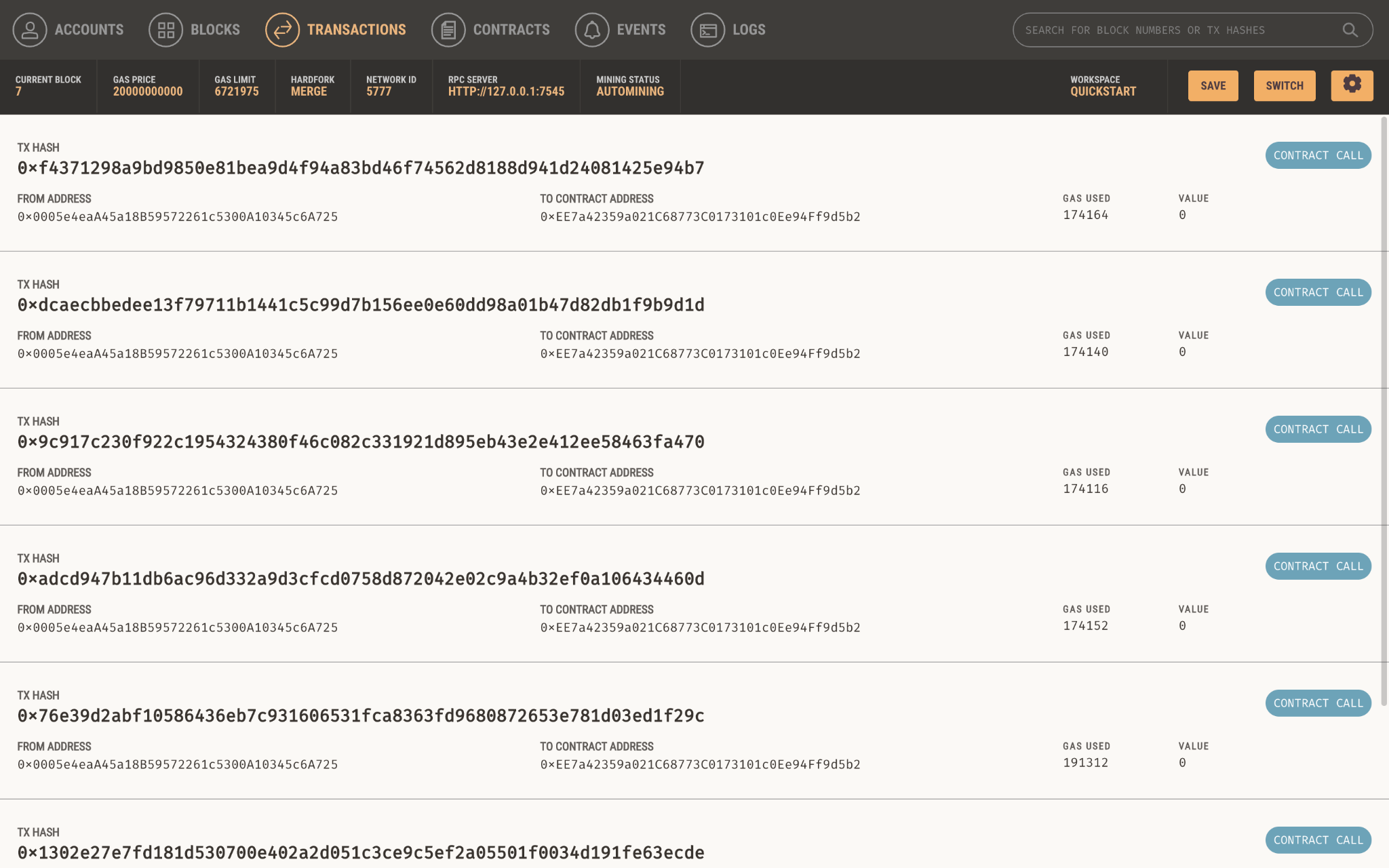
Calling the “NextElectionId” function gives the admin the Election id to use to start the election. The admin will then proceed to enter the id into the “Create Election” function field and transact.



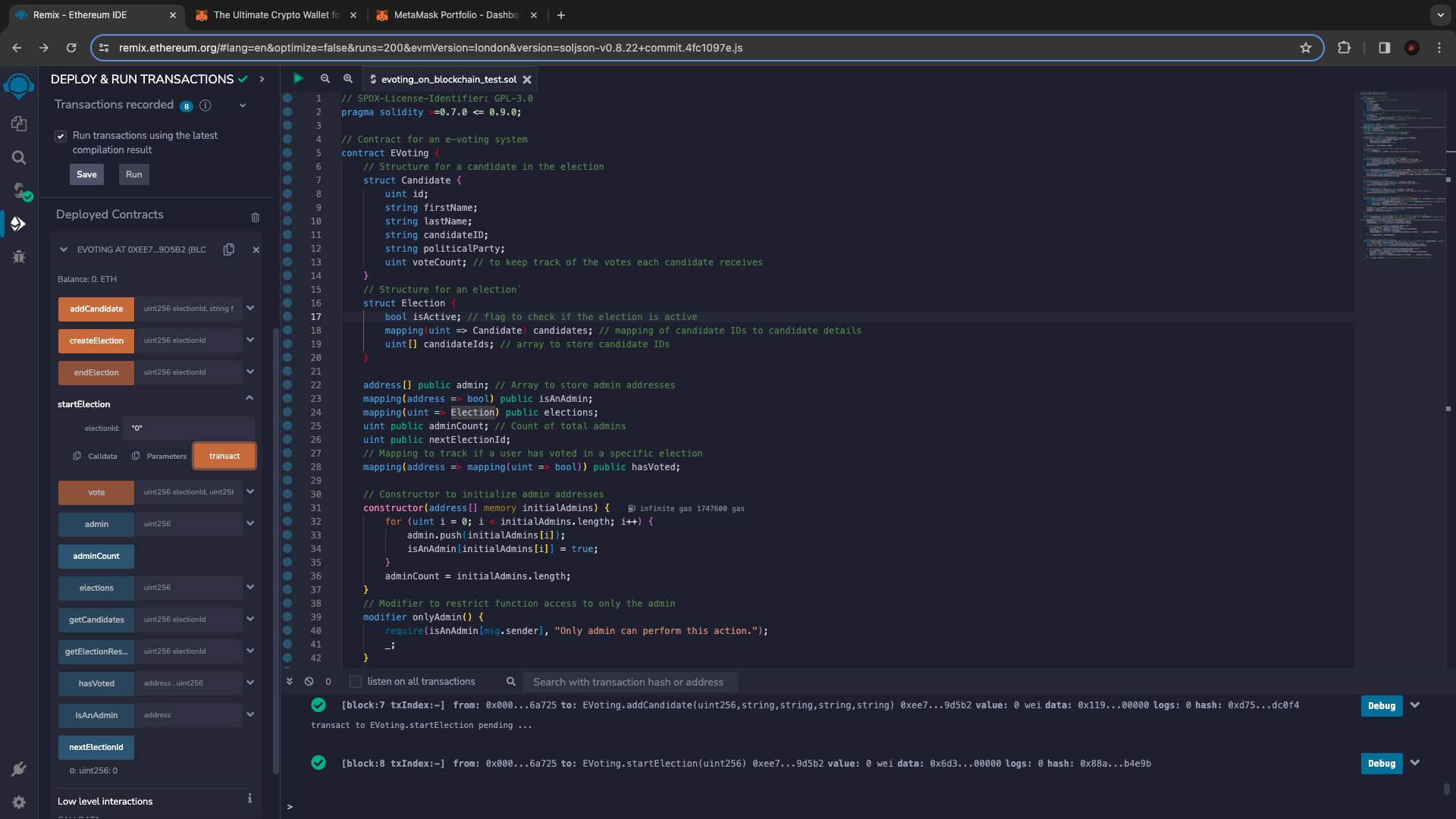
The admin then proceeds to add candidates using the “addCandidate” function. 5 Candidates were added.



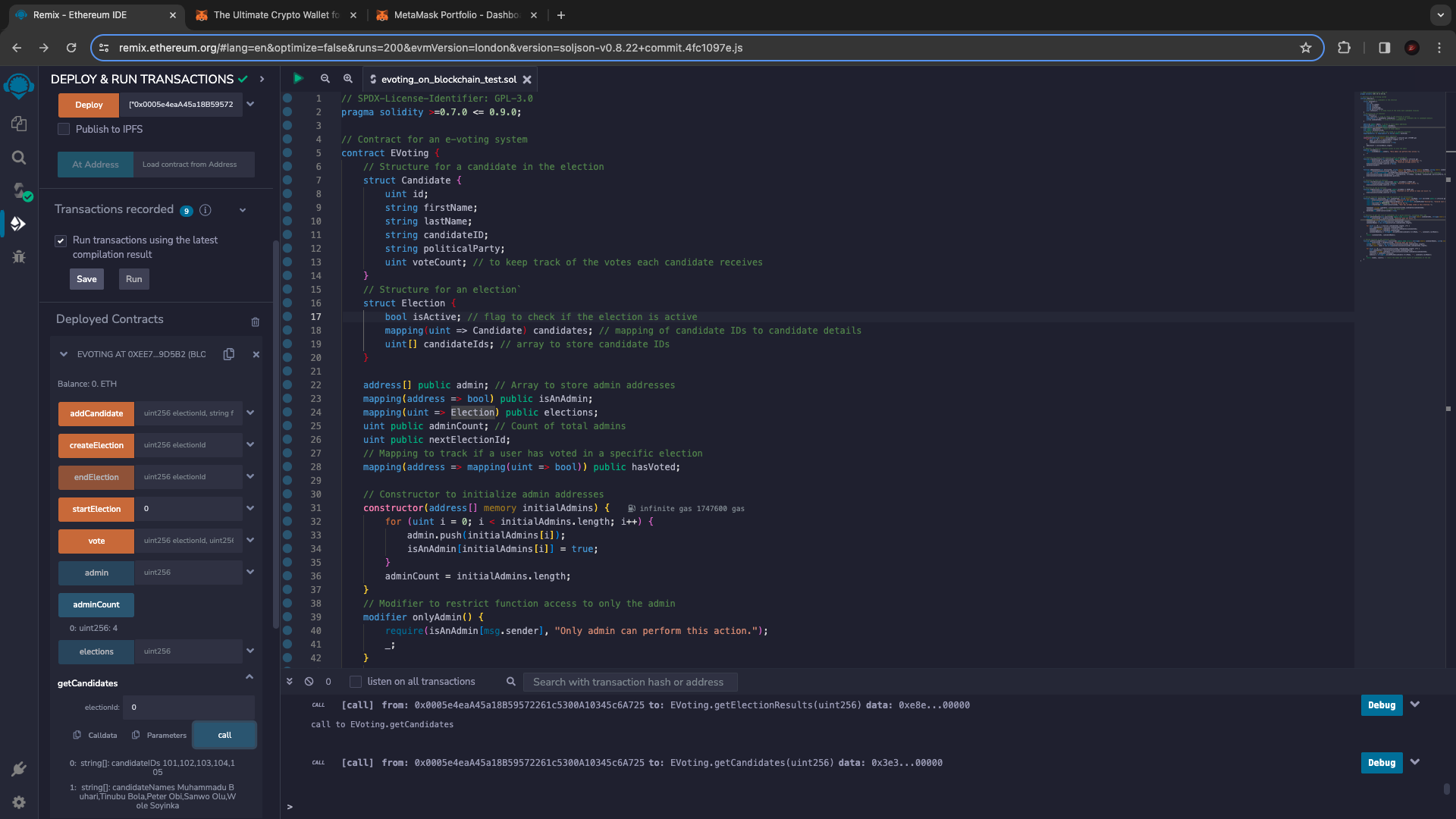
It all reflects on the blockchain as the Election was created and new Candidates are being added to the blockchain.



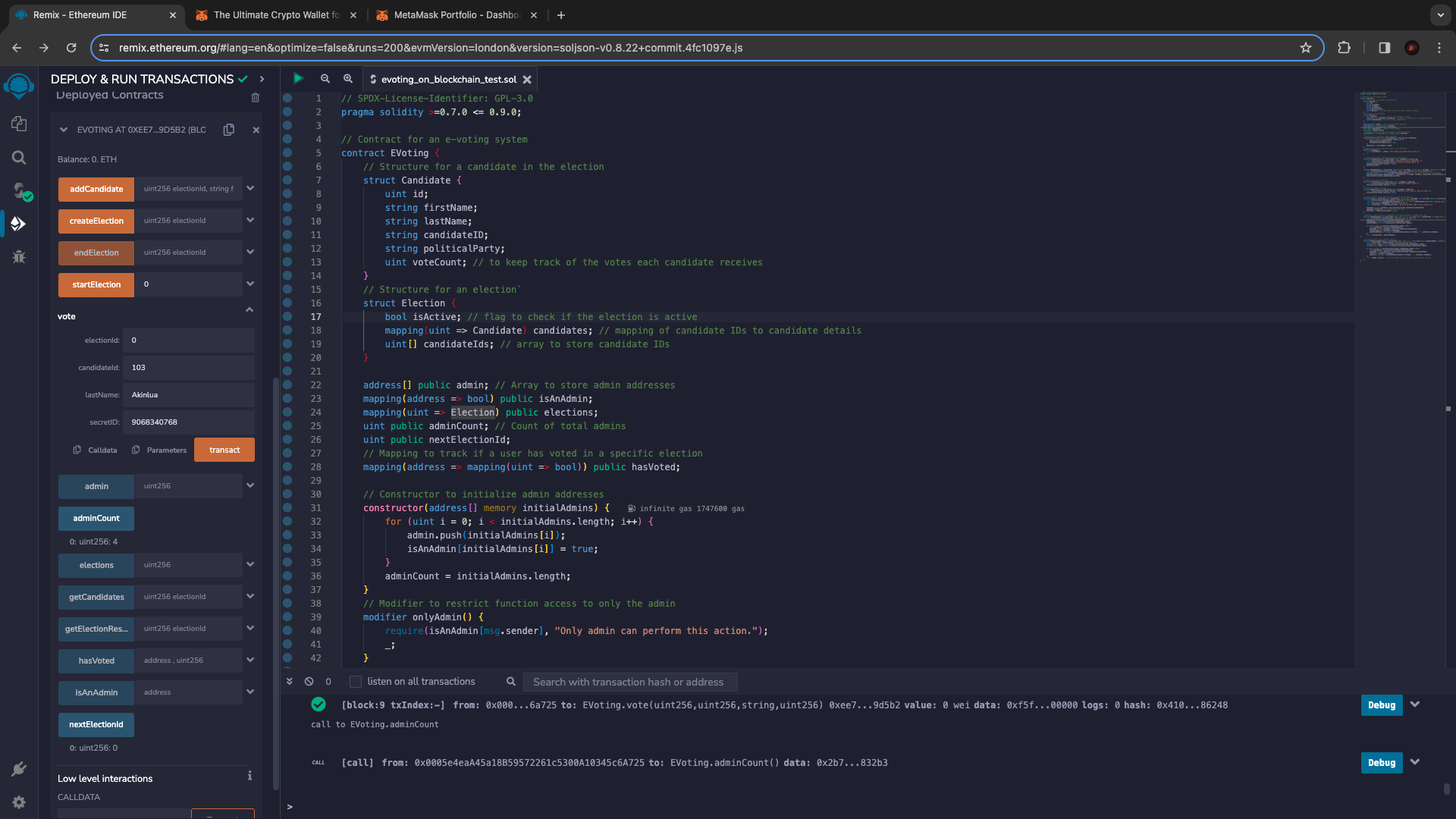
It also reflects as transactions on the blockchain as the Election was created and new Candidates are being added to the blockchain.



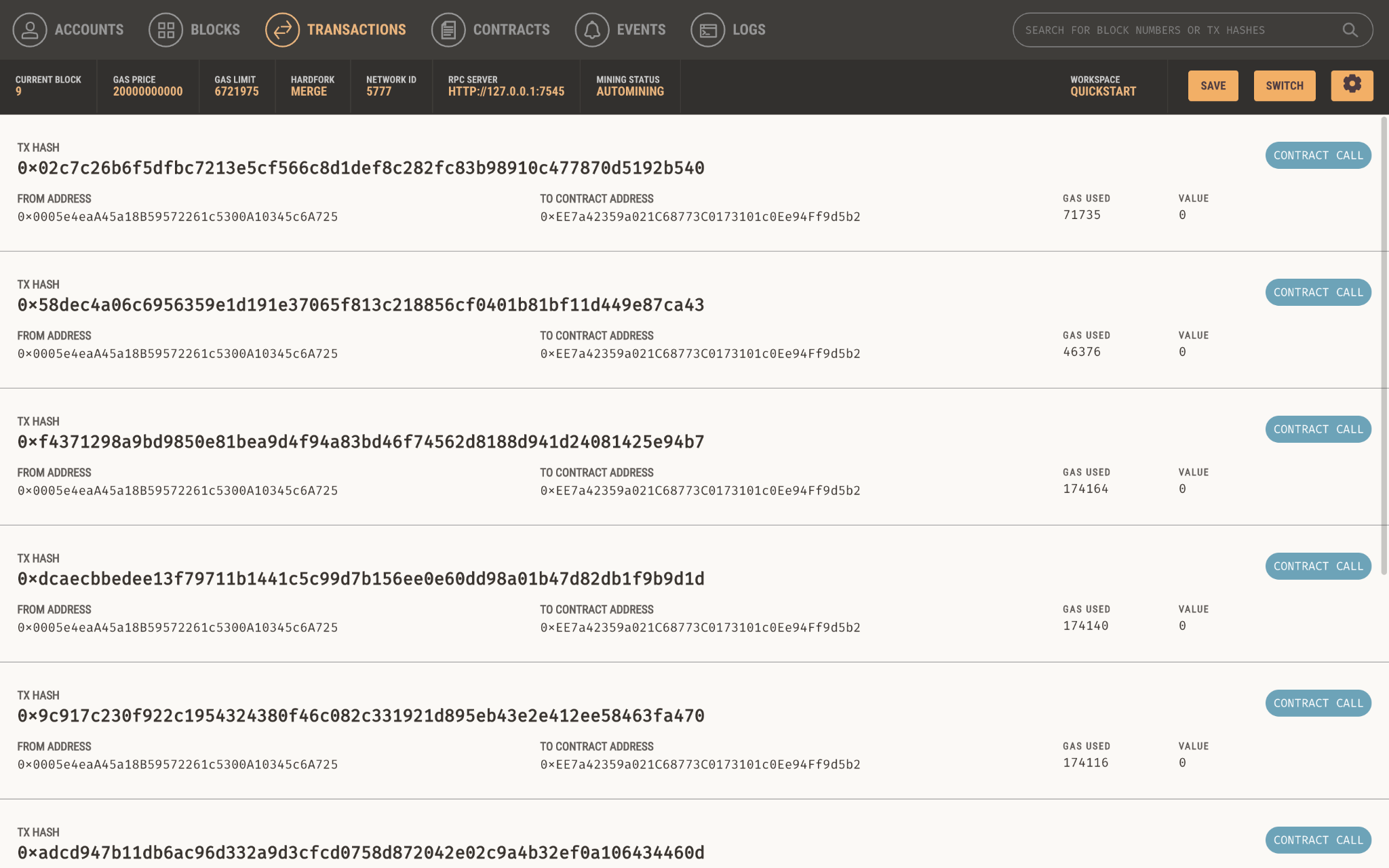
The admin then proceeds to enter the electionId into the startElection function to begin the election.



The user then requests for the candidate options by calling the “getCandidates” function and it returns the candidate ids along side their full names.



The users then vote by inputting the electionId, Candidate id of the candidate they are voting for, Last Name and their secret ID.



To show that the blockchain has received the user’s vote as a transaction on the blockchain.

4.2.3 Smart Contract

// SPDX-License-Identifier: GPL-3.0

pragma solidity >=0.7.0 <= 0.9.0;

// Contract for an e-voting system

contract EVoting {

// Structure for a candidate in the election

struct Candidate {

uint id;

string firstName;

string lastName;

string candidateID;

string politicalParty;

uint voteCount; // to keep track of the votes each candidate receives

}

// Structure for an election`

struct Election {

bool isActive; // flag to check if the election is active

mapping(uint => Candidate) candidates; // mapping of candidate IDs to candidate details

uint[] candidateIds; // array to store candidate IDs

}

address[] public admin; // Array to store admin addresses

mapping(address => bool) public isAnAdmin;

mapping(uint => Election) public elections;

uint public adminCount; // Count of total admins

uint public nextElectionId;

// Mapping to track if a user has voted in a specific election

mapping(address => mapping(uint => bool)) public hasVoted;

// Constructor to initialize admin addresses

constructor(address[] memory initialAdmins) {

for (uint i = 0; i < initialAdmins.length; i++) {

admin.push(initialAdmins[i]);

isAnAdmin[initialAdmins[i]] = true;

}

adminCount = initialAdmins.length;

}

// Modifier to restrict function access to only the admin

modifier onlyAdmin() {

require(isAnAdmin[msg.sender], "Only admin can perform this action.");

\_;

}

// Function to create a new election with a specified ID

function createElection(uint electionId) public onlyAdmin {

require(electionId == nextElectionId, "Election ID must be sequential.");

require(!elections[electionId].isActive, "Election already exists.");

elections[electionId].isActive = false;

nextElectionId++;

}

function addCandidate(uint electionId, string memory firstName, string memory lastName, string memory candidateID, string memory politicalParty) public onlyAdmin {

require(!elections[electionId].isActive, "Cannot add candidates to active elections.");

uint id = uint(keccak256(abi.encodePacked(candidateID))); // Create a unique ID for each candidate

elections[electionId].candidates[id] = Candidate(id, firstName, lastName, candidateID, politicalParty, 0);

elections[electionId].candidateIds.push(id);

}

// Function to start an election

function startElection(uint electionId) public onlyAdmin {

require(!elections[electionId].isActive, "Election already active.");

elections[electionId].isActive = true;

}

// Function to end an election

function endElection(uint electionId) public onlyAdmin {

require(elections[electionId].isActive, "Election is not active or does not exist.");

elections[electionId].isActive = false;

}

// Voting function using last name and secret ID for verification

function vote(uint electionId, uint candidateId, string memory lastName, uint secretID) public {

require(elections[electionId].isActive, "Election is not active.");

require(keccak256(abi.encodePacked(lastName)) == keccak256(abi.encodePacked("Akinlua")), "Invalid last name."); // Using pre saved data for login as I do not have access to database system

require(secretID == 9068340768, "Invalid secret ID.");

require(!hasVoted[msg.sender][electionId], "User has already voted in this election.");

Candidate storage candidate = elections[electionId].candidates[candidateId];

candidate.voteCount++; // Increment the vote count

hasVoted[msg.sender][electionId] = true;

}

// Function to get the list of candidates for a given election, including their IDs

function getCandidates(uint electionId) public view returns (string[] memory candidateIDs, string[] memory candidateNames) {

require(electionId < nextElectionId, "Election does not exist.");

Election storage election = elections[electionId];

candidateIDs = new string[](election.candidateIds.length);

candidateNames = new string[](election.candidateIds.length);

for (uint i = 0; i < election.candidateIds.length; i++) {

uint candidateId = election.candidateIds[i];

Candidate storage candidate = election.candidates[candidateId];

candidateIDs[i] = candidate.candidateID;

candidateNames[i] = string(abi.encodePacked(candidate.firstName, " ", candidate.lastName));

}

return (candidateIDs, candidateNames);

}

// Voting function to get election results

function getElectionResults(uint electionId) public view returns (string[] memory candidateNames, uint[] memory voteCounts) {

require(electionId < nextElectionId, "Election does not exist.");

uint[] memory counts = new uint[](elections[electionId].candidateIds.length);

string[] memory names = new string[](elections[electionId].candidateIds.length);

for (uint i = 0; i < elections[electionId].candidateIds.length; i++) {

uint candidateId = elections[electionId].candidateIds[i];

Candidate storage candidate = elections[electionId].candidates[candidateId];

counts[i] = candidate.voteCount;

names[i] = string(abi.encodePacked(candidate.firstName, " ", candidate.lastName));

}

return (names, counts); // return the names and vote counts of candidates at the end

}

}

5. Conclusion

The paper's key findings emphasize the substantial enhancement of data security in E-voting systems through the implementation of blockchain technology. By addressing the fundamental flaws in conventional voting procedures, our strategy provides a transparent, decentralised, and safe alternative. The study emphasises how important blockchain technology is for safeguarding voting integrity during elections against fraud and other manipulation.

The report does, however, admit certain difficulties, such as scalability and integration into current electoral systems. Prospective avenues for investigation encompass refining user interfaces, enhancing blockchain for increased transaction volumes, and striking a balance between voter anonymity and transparency. This work lays foundations for the advancement of safe, blockchain-based democratic processes.

References

Adida, B., 2008. Helios: Web-based open-audit voting. *In*: *Proceedings of the 17th USENIX Security Symposium* [online]. 335–348. Available from: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85026581986&partnerID=40&md5=59d8fa46f5b98989f9133320b97d7336.

Al-Maaitah, S., Qatawneh, M. and Quzmar, A., 2021. E-Voting System Based on Blockchain Technology: A Survey. *2021 International Conference on Information Technology (ICIT)*.

Ali, T. and Murray, J., 2016. *Real-World Electronic Voting: Design, Analysis and Deployment: CRC Series in Security, Privacy and Trust* [online]. CRC Press. Available from: https://www.crcpress.com/Real-world-Electronic-Voting-Design-Analysis-and-Deployment/Hao-Ryan/9781498714693 [Accessed 19 Dec 2023].

Aljarrah, E., Elrehail, H. and Aababneh, B., 2016. E-voting in Jordan: Assessing readiness and developing a system. *Computers in Human Behavior*, 63, 860–867.

Alvi, S. T., Uddin, M. N. and Islam, L., 2020. *Digital Voting: A Blockchain-based E-Voting System using Biohash and Smart Contract* [online]. IEEE Xplore. Available from: https://ieeexplore.ieee.org/abstract/document/9214250/authors#authors.

Anane, R., Freeland, R. and Theodoropoulos, G., 2007. e-Voting Requirements and Implementation. *The 9th IEEE International Conference on E-Commerce Technology and The 4th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (CEC-EEE 2007)*.

Ashish Singh Parihar, Prasad, D., Gautam, A. and Swarnendu Kumar Chakraborty, 2021. Proposed End-to-End Automated E-Voting Through Blockchain Technology to Increase Voter’s Turnout, 55–71.

Benabdallah, A., Audras, A., Coudert, L., El Madhoun, N. and Badra, M., 2022. Analysis of Blockchain Solutions for E-Voting: A Systematic Literature Review. *IEEE Access*, 10, 70746–70759.

Burton, C., Culnane, C. and Schneider, S., 2016. vVote: Verifiable Electronic Voting in Practice. *IEEE Security & Privacy*, 14 (4), 64–73.

Buterin, V., 2014. Ethereum white paper: a next generation smart contract & decentralized application platform. *First version*, 53.

Çabuk, U. C., Adıgüzel, E. and Karaarslan, E., 2018. A Survey on Feasibility and Suitability of Blockchain Techniques for the E-Voting Systems. *IJARCCE*, 7 (3), 124–134.

Cheema, M. A., Ashraf, N., Aftab, A., Qureshi, H. K., Kazim, M. and Azar, A. T., 2020. *Machine Learning with Blockchain for Secure E-voting System* [online]. IEEE Xplore. Available from: https://ieeexplore.ieee.org/document/9283806.

Dorri, A., Kanhere, S. S. and Raja Jurdak, 2016. Blockchain in internet of things: Challenges and Solutions. *arXiv (Cornell University)*.

El Madhoun, N., Hatin, J. and Bertin, E., 2020. A decision tree for building IT applications. *Annals of Telecommunications*, 76 (3-4), 131–144.

Febriyanto, E., Triyono, Rahayu, N., Pangaribuan, K. and Sunarya, P. A., 2020. Using Blockchain Data Security Management for E-Voting Systems. *2020 8th International Conference on Cyber and IT Service Management (CITSM)*.

Hanifatunnisa, R. and Rahardjo, B., 2017. Blockchain based e-voting recording system design. *2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)* [online]. Available from: https://ieeexplore.ieee.org/document/8272896/.

Hjalmarsson, F. P., Hreioarsson, G. K., Hamdaqa, M. and Hjalmtysson, G., 2018. Blockchain-Based E-Voting System. *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)* [online]. Available from: https://ieeexplore.ieee.org/document/8457919/.

Hossain, S. S., Arani, S. A., Rahman, Md. T., Bhuiyan, T., Alam, D. and Zaman, M., 2019. E-voting system using Blockchain technology. *Proceedings of the 2019 2nd International Conference on Blockchain Technology and Applications*.

Kamil, M., Bist, A. S., Rahardja, U., Santoso, N. P. L. and Iqbal, M., 2021. Covid-19: Implementation e-voting Blockchain Concept. *International Journal of Artificial Intelligence Research*, 5 (1).

Kusters, R., Tomasz Truderung and Vogt, A., 2012. Clash Attacks on the Verifiability of E-Voting Systems. *33rd IEEE Symposium on Security and Privacy, S and P 2012*, 395–409.

Lin, I.-C. and Liao, T.-C., 2016. A Survey of Blockchain Security Issues and Challenges. *International Journal of Network Security* [online], 19 (5), 653–659. Available from: http://dx.doi.org/10.6633%2fIJNS.201709.19(5).01.

Luo, Y., Chen, Y., Chen, Q. and Liang, Q., 2018. *A New Election Algorithm for DPos Consensus Mechanism in Blockchain*[online]. IEEE Xplore. Available from: https://ieeexplore.ieee.org/abstract/document/8634684/.

Malhotra, M., Kumar, A., Kumar, S. and Yadav, V., 2022. Untangling E-Voting Platform for Secure and Enhanced Voting Using Blockchain Technology. *SPRINGER*, 51–72.

Manuskin, A., Mirkin, M. V. and Ittay Eyal, 2020. Ostraka: Secure Blockchain Scaling by Node Sharding. *arXiv (Cornell University)*.

Park, S., Specter, M., Narula, N. and Rivest, R. L., 2021. Going from bad to worse: from Internet voting to blockchain voting. *Journal of Cybersecurity*, 7 (1).

Qatawneh, M., Almobaideen, W. and AbuAlghanam, O., 2020. Challenges of Blockchain Technology in Context Internet of Things: A Survey. *International Journal of Computer Applications*, 175 (16), 13–20.

Ryan, P. Y. A., Bismark, D., Heather, J., Schneider, S. and Zhe Xia, 2009. PrÊt À Voter: a Voter-Verifiable Voting System. *IEEE Transactions on Information Forensics and Security*, 4 (4), 662–673.

Sheer Hardwick, F., Gioulis, A., Naeem Akram, R. and Markantonakis, K., 2018. *E-Voting With Blockchain: An E-Voting Protocol with Decentralisation and Voter Privacy* [online]. IEEE Xplore. Available from: https://ieeexplore.ieee.org/abstract/document/8726645.

Shorman, A. and Qatawneh, M., 2018. Performance Improvement of Double Data Encryption Standard Algorithm using Parallel Computation. *International Journal of Computer Applications*, 179 (25), 1–6.

Singh, S., Wable, S. and Kharose, P., 2022. A Review Of E-Voting System Based on Blockchain Technology. *International Journal of New Practices in Management and Engineering*, 10 (04), 09-13.

Srikrishnaswetha, K., Kumar, S. and Rashid Mahmood, Md., 2019. A Study on Smart Electronics Voting Machine Using Face Recognition and Aadhar Verification with IOT. *Lecture Notes in Networks and Systems*, 87–95.

Szabo, N., 1997. Formalizing and Securing Relationships on Public Networks. *First Monday*, 2 (9).

Taş, R. and Tanrıöver, Ö. Ö., 2020. A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting. *Symmetry*, 12 (8), 1328.

Vujičić, D., Jagodić, D. and Ranđić, S., 2018. *Blockchain technology, bitcoin, and Ethereum: A brief overview* [online]. IEEE Xplore. Available from: https://ieeexplore.ieee.org/abstract/document/8345547.

Wouter Bokslag and Manon de Vries, 2016. Evaluating e-voting: theory and practice. *arXiv (Cornell University)*.

Yi, H., 2019. Securing e-voting based on blockchain in P2P network. *EURASIP Journal on Wireless Communications and Networking* [online], 2019 (1). Available from: https://jwcn-eurasipjournals.springeropen.com/articles/10.1186/s13638-019-1473-6.

Zamani, M., Movahedi, M. and Raykova, M., 2018. Rapid-Chain: Scaling Blockchain via Full Sharding. *Canada ACM Reference Format: Mahdi Zamani* [online], 18. Available from: https://dl.acm.org/ft\_gateway.cfm?id=3243853&ftid=2008927&dwn=1.

Zhang, S., Wang, L. and Xiong, H., 2019. Chaintegrity: blockchain-enabled large-scale e-voting system with robustness and universal verifiability. *International Journal of Information Security*.

Zhang, W., Yuan, Y., Hu, Y., Huang, S., Cao, S., Chopra, A. and Huang, S., 2018. A Privacy-Preserving Voting Protocol on Blockchain. *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*.