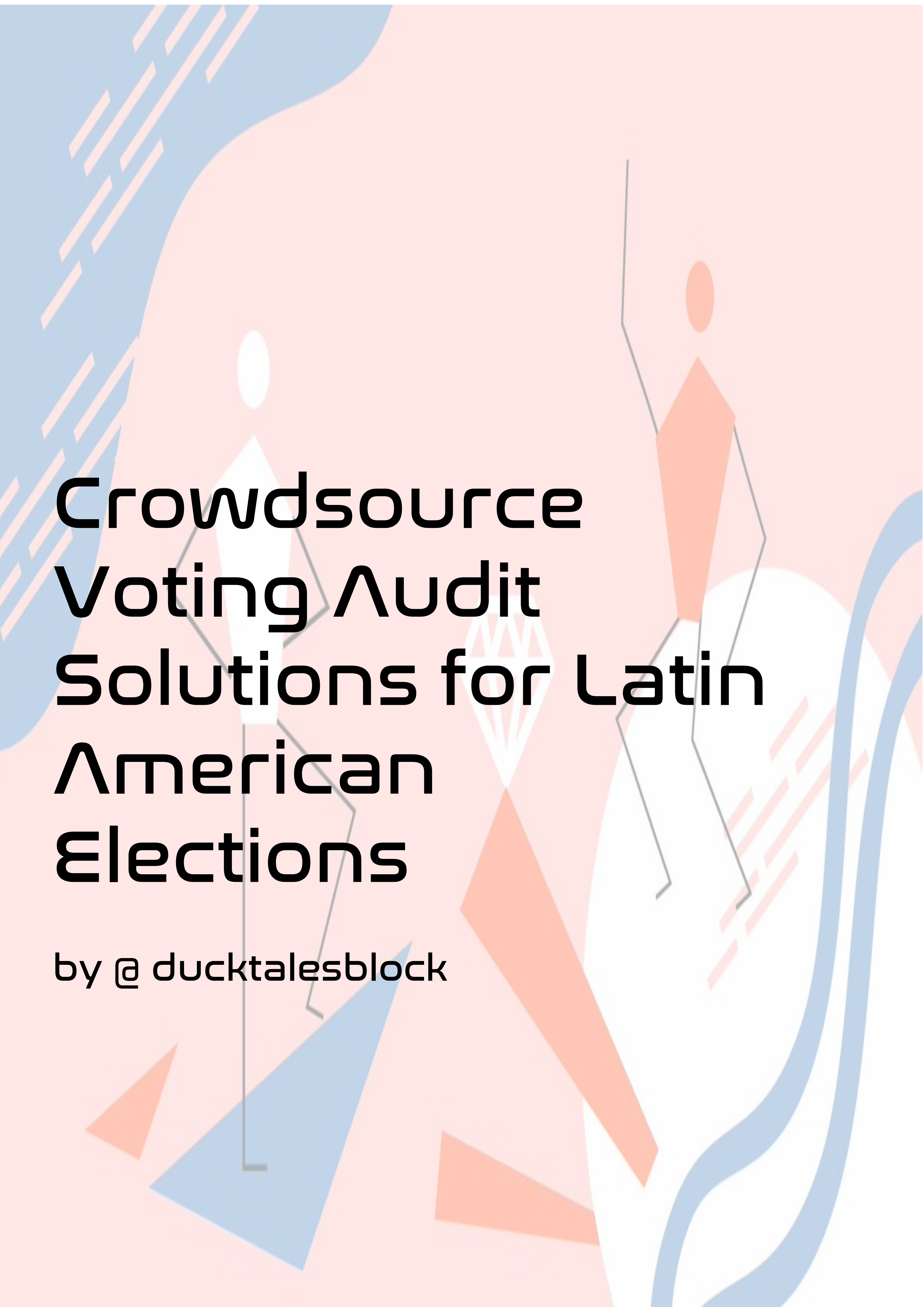


Crowdsource Voting Audit Solutions for Latin American Elections

by @ ducktalesblock

Table of contents

Introduction

The exercise of holding elections through a voting process, either using conventional voting systems (physical voting cards) or electronic voting systems, has become an event of greater social relevance, in addition to being the direct route in which the citizens or members of an organization connect and express themselves with whoever governs them. This exercise is the source and support of legitimation between the government and the governed, giving a certain sense of political stability in a nation or the different administrative structures that, through democracy, elect their representatives. ways to vote, going from emails, encrypted digital ballots, voting devices, secure FTP connections, cards, certifying authorities, using biometrics, now Blockchain, among others. The traditional voting methodologies used today have given results However, in different social sectors, electoral results are discussed with arguments ranging from the manipulation of the voter registry (impersonation, voting of unauthorized persons), inaccurate vote count, security fragility of existing systems, to the inability to carry out a comprehensive audit of the voting system, to generate confidence This document presents a proposal for electronic voting supported by Blockchain technology, the objective of which is to largely satisfy the needs expressed by social actors, combining cryptographic procedures and programming. agreements between the parties through smart contracts (Smart Contracts) .These methods seek to guarantee compliance with universal voting principles, such as: anonymity, inability to link a voter with the vote, impartiality, such as the know partial results until the end of the vote, verifiability as the ability to verify the transaction made, reliability and integrity is known as the inability to remove or change votes, and security is understood as protection against denial of service attacks or loss of information

Context

This proof of concept was approached with the purpose of satisfying the universal principles of voting systems, and especially those of reliability and integrity, so that in this section the main concepts of voting systems are presented. and of the technologies used in the Blockchain.

Electronic voting

It is the most advanced system within the electronic democracy systems, since it is 100% digital, from the authentication of the citizen to the casting of the vote. There are 2 types of electronic vote, the first is day when the voter is present at their polling station and the second when it is carried out using the internet in any location. For this work, it is assumed that the electoral process will be done with the use of voting tables. physically located in a polling station, which would be from where the citizen will cast a vote in favor of the candidate of their choice. Electronic voting forms range from the use of perforated cards, optical scanning systems and electronic voting systems. Direct Recording (DRE) to Internet ballots and telephone votes. However, they all had something in common, the different principles that must be considered for a vote to be successful. According to the following should be considered: eligibility, non-reusable, anonymity, accuracy, no bias, public, verifiable, and integrity.

Previous works

The work proposes a system that allows an electoral process to be carried out, where voters can issue tokens from their wallets to a candidate; after validation of your identity by the administrators. In addition, they use a zero-knowledge test to validate the cast ballot. In they propose to use IoT to authenticate the voter and subsequently send their vote to the blockchain, which is where it finally remains stored. Once the election day is over, they start counting the votes and cast the winner. In it is stated that this is achieved at a lower cost compared to traditional paper voting. Blockchain technology attracts more and more eyes to be used in electronic voting, and this can be seen in, which analyzes 15 works of different approaches in favor of electronic voting. Most agree that Ethereum is a promising technology for such events, and they use it in a variety of ways. To make a comparison of these works, the authors rely on the following parameters such as: authentication, platform, anonymity, voter verification, decentralized and technology used. There are countries that have implemented this system, among them are: Belgium, Brazil, United States, Estonia, Philippines, India, and Venezuela. Some are in the process of being implemented and others avoid it, such as Germany, Holland, Finland, Ireland, Kazakhstan, Norway, and the United Kingdom. In they show a voting system called Bron-coVote for universities using Ethereum and its Smart Contracts, which allow them to manage voters and vote traceability. Of the countries mentioned, Estonia decided to implement electronic voting by means of a system called I-Voting, which allows all its inhabitants to vote from any corner of the world and changing the election of the candidate if the voter so wishes, where the last the candidate who has been selected will be the one who will have their vote counted I-Voting was received with skepticism, now it enjoys great popularity and every time it is increasing, reaching an average voting time on the Internet in 2015 of 2 minutes and 36 seconds, which was calculated keeping in mind that the electoral day lasts a whole week (from 21 to February 27). Bolivia is one of the many countries that continue to vote in the traditional way with physical cards, voting tables in a specific place, juries and voters; having the same problems as other countries: electoral fraud, human errors in the processes, violation of computer attacks, centralized electoral process by autonomous entities, among others. In an electronic voting proposal is made for Bolivia using a wallet and coins as an asset to vote, for which the voter sends a coin to the candidate of his preference as a form of vote. Not of this technology, ideas continue to be investigated and proposed, among which are: certifying entities, digitally signed votes to verify their origin, secure FTP (File Transfer Protocol) sessions, symmetric encryption algorithms, among others.

Blockchain

This technology is used to build specific types of distributed databases composed of immutable data blocks, each with a list of transactions and a unique reference to its predecessor block. Blockchain technology is receiving intense and growing attention among governments. In order to make references to previous blocks, mathematical relationships of hashes are used, being the database protected cryptographically and managed by a global network of computers, where the stored information cannot be altered. The Blockchain network was born in the year In not 2008 theoretically and in 2009 one was implemented for Bitcoin. The theory of it was made known by the pseudonym Satoshi Nakamoto through the Whitepaper reported. Inside a Blockchain, everything is a node. A node refers to a person who, through a computer, with a local copy of the network and special mining software, becomes part of the network. This person is in charge of doing block mining, ensuring the integrity and transparency of the network, by participating in a mechanism called consensus. All blockchain status updates are made through transactions, using public key cryptography and private. These transactions generate a cost measured in gas, which is a measure of the computational expense by the miners to be able to write to the network. The amount of gas used in a transaction determines the reward for the miners. This gas in addition to be the economic stimulus for the participation of the miners in the network, it can also be used in the prevention of attacks on the network, since, to send many transactions with the purpose of generating a denial of services attack, this is costly for the attacker, since each time he carries out an attack he consumes gas that has a real monetary value. Network security is directly proportional to the number of active miners, so they manage it and help prevent attacks such as the 51%, where a miner takes 51% of the network's computing power, and could manipulate it at any time.

TON

Free TON is a peer-to-peer multi-blockchain system with the native TON Crystal (TON) token. The Free TON platform is based on the TON Protocol developed by Nikolay Durov and the Telegram team.

Technologies used

The following were used in the proof of concept:

- TON MAINNET:
- TON SURF plugin or something like MetaMask to implemet TON Wallet in website
- Truffle: a development suite for smart contracts, which has a debugger, compiler, and commands to display Smart Contracts.
- React JS JavaScript: library and programming language, respectively, used for FrontEnd.
- Web3JS: a library through which you can connect the FrontEnd with the Smart Contracts stored in the Blockchain and call their functions.

The FreeTON Blockchain is chosen, since it allows the programming of smart contracts for business logic, allowing the creation of customizable tokens, in addition, it is considered to be a stable Blockchain, which has a great reception for its flexibility, and is expected to be supported for a long time. The Bitcoin network is discarded due to its high transparency approach that can be detrimental to anonymity, falling into the drawbacks described in the next section. In addition, you have no control over it. Cardano is discarded because the introduction of smart contracts is recent and could contain errors, the same happens with other blockchains.

Problem statement

The traditional form of voting like cards, as a participation mechanism, receives harsh criticism for its low transparency and its high complexity in auditing the votes cast. Added to this, the distrust of the voters given the guarantees of integrity, reliability and anonymity, is increasing, since the system lends itself to sequencing votes in favor of a candidate, through third parties that interact in the counting. In addition, some of the proposals found, as described in the following section, address some of these principles to a greater extent, leaving aside others, which is why a solution is necessary to address these needs.

Solution

Unlike other proposals, this one eliminates the need for third parties (such as certifying authorities), also the possible coercion of voters, and exposing a public key linked to the vote when the transaction is carried out; disadvantages that in the majority of literature do not delve. The proposal is based on the TON standard. This allows to create tokens and, in this particular case, to use them as a vote. It should be noted that the behavior of the token functions has been subtly modified in order to adapt it to the project; particularly, the way in which the balances of the candidates' accounts are consulted, in order to comply with the impartiality criterion. The authors in propose a similar form of voting but using coins, where each voter is assigned one and can send it to the candidate's wallet as support, from any place and device that supports the technologies analyzed. This method is easy and fast, however, there is a risk of finding the identity of the voter, since the voter uses his public password to vote, being registered in the transaction, and in turn, it would be linked to the wallet of the candidate; therefore, anyone who knows that key will be able to know for whom that person has voted. Another drawback is the coercion to reveal your public key, and thus, search through the blockchain transactions until you find the vote transaction. Finally, another highlight is the additional work given to the voter, since they must remember the private address to be able to vote, and, if they lose it, they will not be able to do so. In this proposal that is presented, only the address of the table and that of the candidate are used to transfer tokens instead of coins, thereby overcoming the previous inconveniences of anomity, by not having an address on which to relate the voter with your vote. To begin with, as many tokens are created as potential voters are enabled, and a certain amount of these is transferred to the tables, based on the number of voters who have been assigned to it using paper lists. This is done in order that each table is the one who authenticates and authorizes their voters to spend tokens that were transferred to them; Therefore, the voter is free to spend (issue) a single token (vote) on behalf of the polling station, so he can transfer it to the candidate's account as a token of support (vote). As a consequence, data traceability can be carried out, given that simple granularities, and furthermore, allow voter anonymity, since the table is only linked to the vote cast and the recipient candidate.

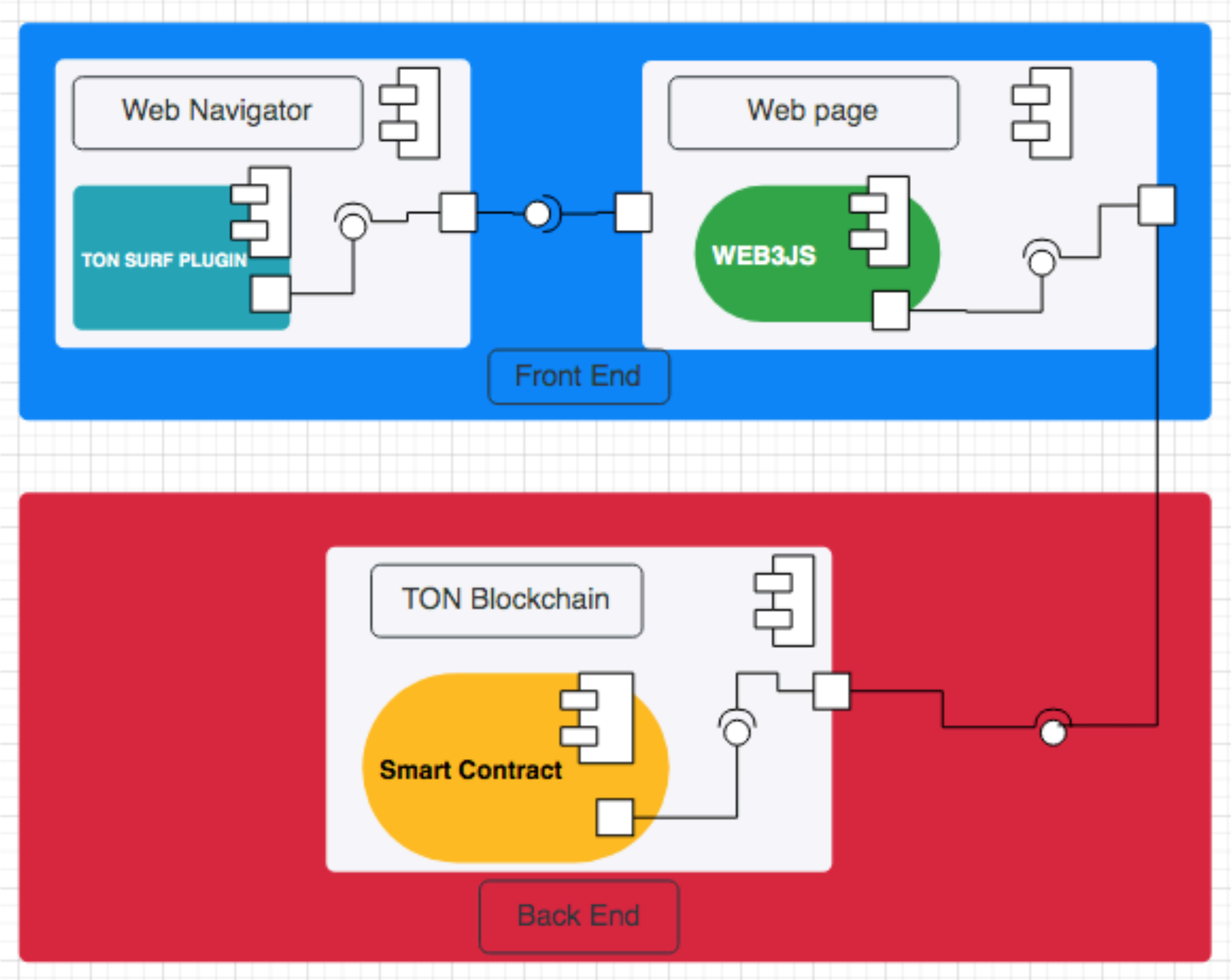


Fig. 1. Architecture decoupled from the project.

Architectural design

Within the design, 2 architectural patterns were used, the one used by the Blockchain network and the Model View View-Model (MVVM) pattern. The first pattern will be the model (Model) of the second to store data from the DApp. The view (View) is made up of the user interfaces used to present the information to the user, and the ViewModel corresponds to the contracts, where all the business logic will be programmed. In order to know how the different components of the system that is being developed, a component diagram is presented to see the different interrelationships, as can be seen in Figure 1. This is more easily explained using Figure 2, which is an abstraction of Figure 1. Both figures show how the flow of data is worked, bearing in mind where they originate, through the actions of a user, and where they end up stored, which is in the Ethereum Blockchain. Now that the data is modeled, we proceed with the architecture. In this case, it corresponds to work with the ViewModel, starting with the smart contracts. The different smart contracts implemented can be seen in Figures 3 to 5. These are the ones that will act as the ViewModel within the MVVM architectural pattern. This logic can be programmed using some language that can be interpreted by the TVM. Before continuing, the following is clarified: there are 3 types of contracts and 3 types of actors. The first actor is the representative of the Electoral Council and is in charge of using the ElectoralProcessContract contract; the second is the representative of the voting tables, who interacts with the VotingTableContract contract during election day, and the third is the voter, who also uses this last contract. It is proposed that each of the tables located in locations have at your disposal a Smart Contract (VotingTableContract) through which you can communicate and act as ViewModel within the MVVM architectural pattern. This logic can be programmed using some language that can be interpreted by the TVM. Before continuing, the following is clarified: there are 3 types of contracts and 3 types of actors. The first actor is the representative of the Electoral Council and is in charge of using the ElectoralProcessContract contract; the second is the representative of the voting tables, who interacts with the VotingTableContract contract during election day, and the third is the voter, who also uses this last contract. It is proposed that each of the tables located in locations have at your disposal a Smart Contract (VotingTableContract) through which you can communicate with a main contract (ElectoralProcessContract) controlled by the Entity in charge of monitoring the electoral process, and in turn, they can communicate with a third contract named TokenVoteContract. The Electoral Contract Process Contract will manage matters related to the initial configuration of the electoral process (creation of voting tables, candidates, authorization to start and close voting, enable public scrutiny) being used only by the person designated by the Electoral Council. The tables and voters may use the VotingTableContract contract to authorize votes and vote, respectively TokenVoteContract, will have a record of the balances (number of votes) of each of the accounts of the different candidates and enabled tables, during the development of The advantages offered by Smart Contracts and the TON Blockchain are the possibility of carrying out transactional operations that are atomic, that is, a transaction is carried out completely or the TVM reverses the changes if a problem occurs. This can also be done manually, to undo changes when fraud attempts are detected in the consensus. Another advantage is the subscription to events that are occurring in Smart Contracts, for example, issuing an alert when the electoral process is activated, causing it to reach a FrontEnd or device, in order to mitigate fraud attempts. Finally, the Smart Contracts can be configured so that they only accept connections from a specific network domain, in this particular case, the voting stations.

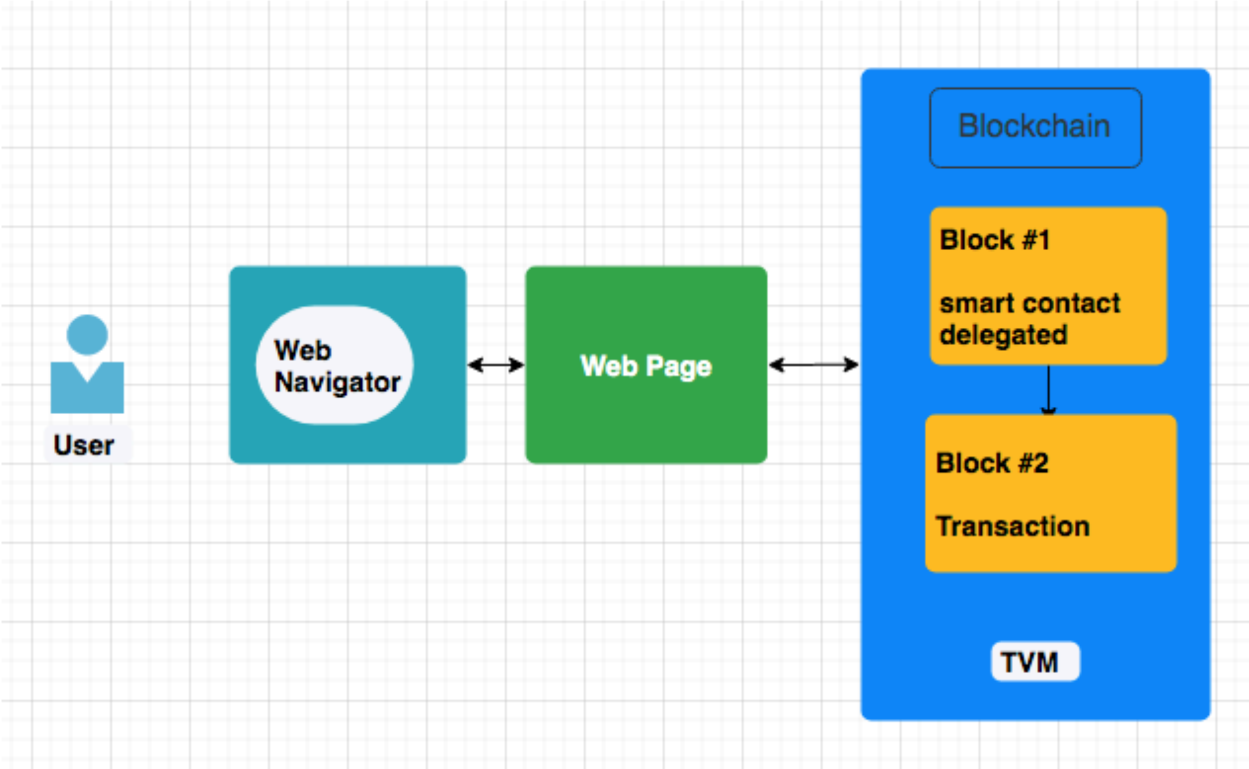


Fig. 2. Interaction of a user with the application.

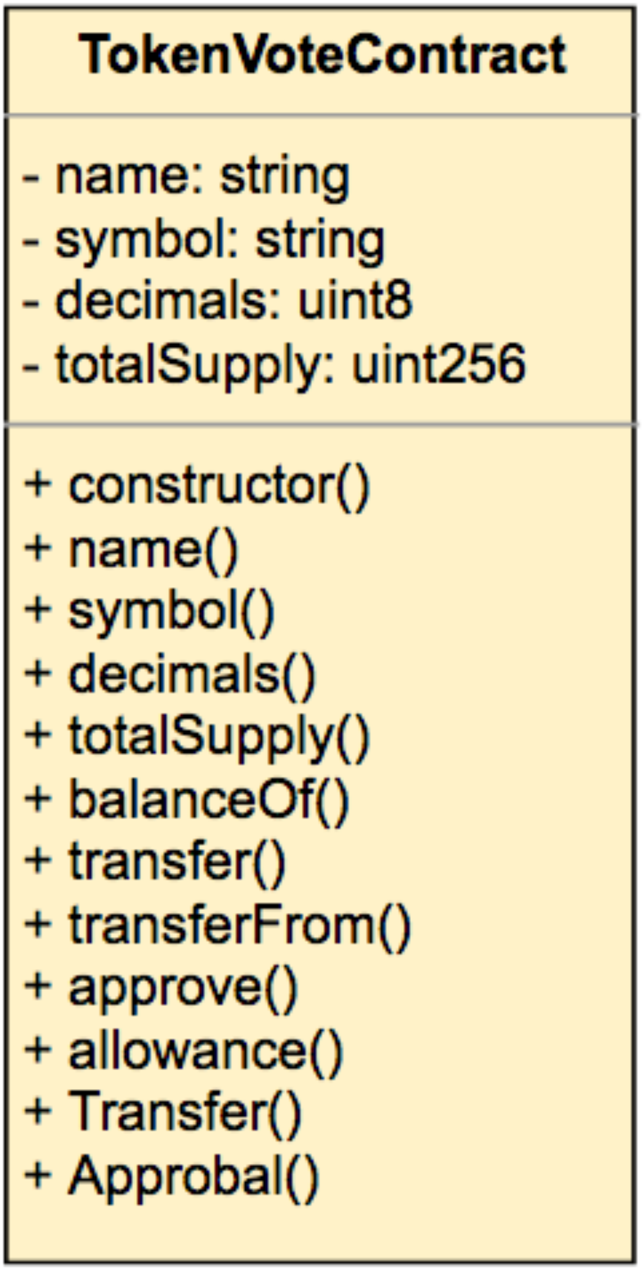


Fig. 3. Example of Smart Contract TokenVoteContract used to store the votes (tokens) of the candidates

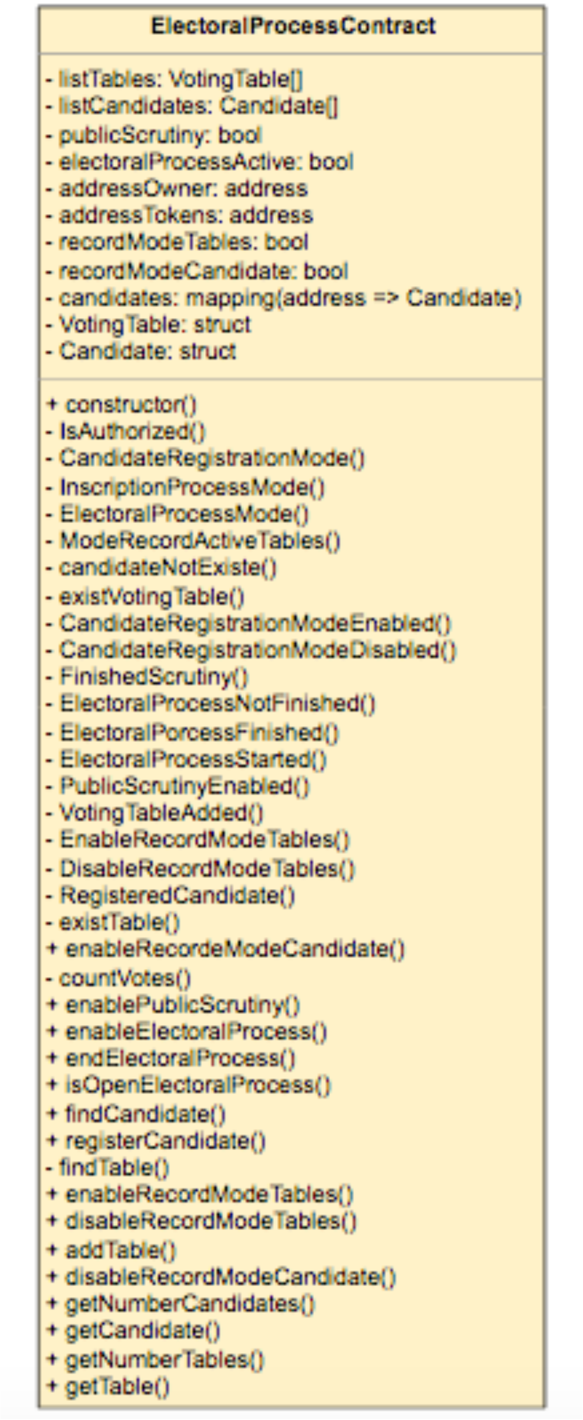
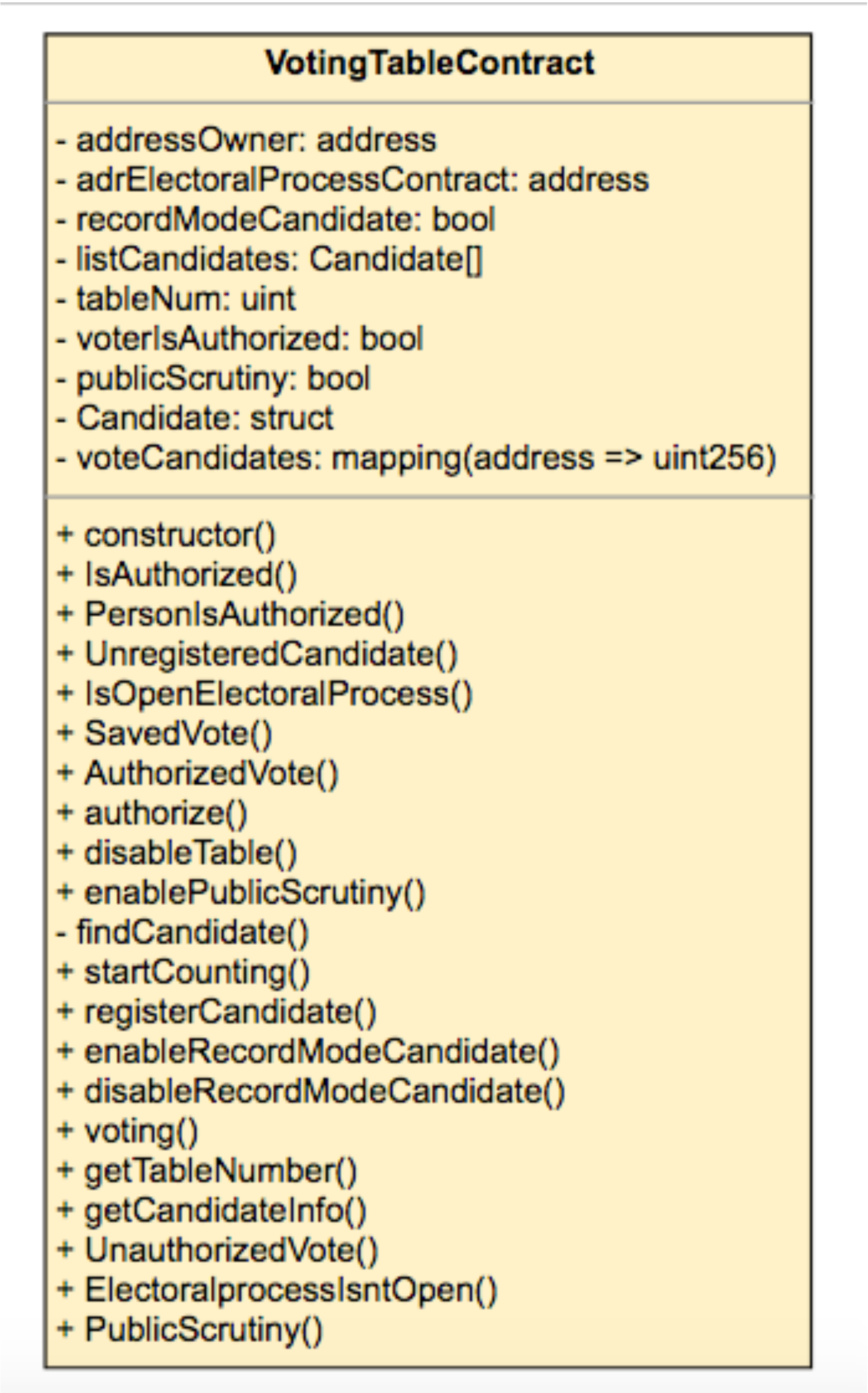

ElectoralProcessContract
- listTables: VotingTable[]
- listCandidates: Candidate[]
- publicScrutiny: bool
- electoralProcessActive: bool
- addressOwner: address
- addressTokens: address
- recordModeTables: bool
- recordModeCandidate: bool
- candidates: mapping(address => Candidate)
- VotingTable: struct
- Candidate: struct
+ constructor()
- IsAuthorized()
- CandidateRegistrationMode()
- InscriptionProcessMode()
- ElectoralProcessMode()
- ModeRecordActiveTables()
- candidateNotExiste()
- existVotingTable()
- CandidateRegistrationModeEnabled()
- CandidateRegistrationModeDisabled()
- FinishedScrutiny()
- ElectoralProcessNotFinished()
- ElectoralPorcessFinished()
- ElectoralProcessStarted()
- PublicScrutinyEnabled()
- VotingTableAdded()
- EnableRecordModeTables()
- DisableRecordModeTables()
- RegisteredCandidate()
- existTable()
+ enableRecordeModeCandidate()
- countVotes()
+ enablePublicScrutiny()
+ enableElectoralProcess()
+ endElectoralProcess()
+ isOpenElectoralProcess()
+ findCandidate()
+ registerCandidate()
- findTable()
+ enableRecordModeTables()
+ disableRecordModeTables()
+ addTable()
+ disableRecordModeCandidate()
+ getNumberCandidates()
+ getCandidate()
+ getNumberTables()
+ getTable()

Fig. 4. Smart Contract Electoral Process Contract used to manage the electoral process



VotingTableContract

- addressOwner: address
- adrElectoralProcessContract: address
- recordModeCandidate: bool
- listCandidates: Candidate[]
- tableNum: uint
- voterIsAuthorized: bool
- publicScrutiny: bool
- Candidate: struct
- voteCandidates: mapping(address => uint256)

+ constructor()
+ IsAuthorized()
+ PersonIsAuthorized()
+ UnregisteredCandidate()
+ IsOpenElectoralProcess()
+ SavedVote()
+ AuthorizedVote()
+ authorize()
+ disableTable()
+ enablePublicScrutiny()
- findCandidate()
+ startCounting()
+ registerCandidate()
+ enableRecordModeCandidate()
+ disableRecordModeCandidate()
+ voting()
+ getTableNumber()
+ getCandidateInfo()
+ UnauthorizedVote()
+ ElectoralprocessIsntOpen()
+ PublicScrutiny()

Fig. 5. Smart Contract VotingTableContract, used to authorize and cast votes.

Decentralized application or DApp

As a result of the design proposed in the previous sections, the development of the application presented in Figures 6 and 7 is achieved. In the complementary material that is related in Section VI, it is possible to visualize the different roles that are part of the electoral process, such as also the options to consult the final results, or by table depending on the case. Through the web page of Figure 6, the voter can choose their preferred candidate, and once the table authorizes their vote, they can send it as support to the chosen candidate. This authorization is given through another interface using a button to authorize vote, a functionality that allows the voter to spend a token from the balance of the table and send it to a candidate for support. Through Figure 7, we will have total control over the electoral process, allowing it to be managed. Once the electoral day has concluded, a public page is available, where any interested party will be able to know the results of the electoral process. This can also be done by using another of the pages that is implemented, enabling queries of the votes obtained by each candidate per table, in order to check the final results, complying with the principle of public and verifiable.

Results and discussion

Using the proof of concept as a starting point to have reliable data, and thus make correct judgments when evaluating these technologies in later sections, a table is made in which the different expenses in terms of gas and crystal (known as TON Crystal) are presented, as indicated, gas represents the computational cost required to execute a transaction and be added to the Blockchain , said cost allows to establish the economic value of executing each function of a Smart Contract.

Discussion

Blockchain is a viable promise to be used in small electoral contexts, unless the duration of the electoral day is extended, since mining compromises its performance and scalability. Compared to other works such, this present proposes a Similar electronic voting form, avoiding linking the public key to the vote. In addition, the task of the voter to save the pair of keys is removed to be able to vote in each electoral period, seeking to mitigate the inconveniences due to the loss of their keys. In the works cited in the present, there are a variety of proposals, however, the way to implement them is not mentioned, and they do not provide the developed software, unlike the present one. This proposal does not allow the coercion of voters, and they can vote protected in the voting points. Otherwise if they voted from home, especially in hostile areas.

Implementation in Latin America

Blockchain has shown to have great adaptation flexibility, so it can be implemented in supply chains, digital identity, transparency in government contracting processes, product traceability, money transfers without intermediaries, electronic voting. Unique. This can be done through a company's own infrastructure or provided by a third party in the cloud. In order to have better control over the transactions carried out on this technology, it is necessary for each government to issue well-defined policies for regulatory purposes.

Conclusions

The consensus procedure, the absence of a central authority, added to the replication of information in each node, allows them to judge unequivocally if the new block to register in the Blockchain does not maliciously alter it, this promises the immutability of the votes stored, highlighting the viability of the technology in small electoral contexts. In addition, like everything in the public Blockchain, anyone can perform manual vote counting and corroborate with the final results, although it should be clarified that the application does not allow partial vote counting during election day, only when enables public scrutiny (in consideration of the principle of impartiality) by the Electoral Council. It is necessary to work on decentralized identities for different voters, so this is considered as future work. Wherever there is a way, under the restrictions of the basic voting principles that were raised in previous sections, to deliver a digital identity to the voter, without compromising their anonymity.

Appendix: App

Environment configuration Voting App

Voting App is based on decentralized technologies, and in this particular case, on TON Blockchain; technologies that are not manipulated directly by browsers for the reason that they are not designed for this. In order to make it possible to communicate with the Blockchain that runs in memory, the pertinent configurations are carried out so that this can be carried out successfully.

Necessary software to learn about Voting App

This section is an explanation for people who want to implement this application on a personal computer, it contains information to prepare a work environment for people who are not involved in software development (in a next section we will talk about those who seek to modify the code of this). It will be divided into sections to obtain particular information in a faster way.
So far, the name of the project and the necessary network configurations have been filled out so that something like Truffle for TON, and something like Metamask as I named TON SURF PLUGIN and Web3JS (technologies discussed later) can successfully connect to the main Blockchain. Therefore, it is necessary to add to the above, the fact that be able to create as many accounts as needed for testing, for example.

It is necessary to be careful when doing so, since these changes cannot be modified once they are created, and it is enough to have configured as it appears in the Proceeding with the explanation of the cited image, user will have the amount of crystal (false) that each of the accounts will have as initial balance, this serves to be able to carry out transactions, it is not recommended to leave it at 0, since you will not be able to use the application and errors will jump. In the next field, you have the total number of accounts that you want to create depending on the needs of the project. Finally, the option to auto-generate MNEMONIC must be enabled. The twelve words generated by this option allow the wallet to be imported by other wallet managers using these words, in this case, they will be imported by TON SURF PLUGIN. Once these settings have been made, user will press the button in the upper right-hand corner labeled SAVE WORKSPACE

Main interface after configuring the Blockchain network. There will be a bar with the number of blocks added to the network, the gas default settings, network identifier, the RPC server (communication protocol), the Automatic option, the name. of the current workspace, change workspace (SWITCH), modify some options of the workspace (gear icon) and the list of accounts that we assigned previously.

Browser

Now it is time to select a browser and it may be one of your preference, in this case, Mozilla Firefox should be selected. Caution should be exercised with the selection of the browser since it must be compatible with TON SURFT plugin.

Additional software

I guess we should use the necessary tool to be able to deploy contracts. Truffle is a suite, which allows the development of smart contracts, compiling, testing and deploying them to a Blockchain network. It is for these reasons that it is necessary to be able to work with the DApp (which will be discussed later) that was developed.

Additional data for developers.

Those people who wish to modify the proposed code must have knowledge of these particular topics of each of the following technologies:

- ReactJs: particularly working with components, states, props, react-router, manipulation of the browser DOM, manipulation of the ReactJs virtual DOM , life cycle of a component, import and use of installed packages with NPM, JSX syntax, JavaScript, component rendering, inheritance, modules, anonymous functions, callbacks, ECMAScript-6 syntax, ECMAScript-5, JSON, OOP with JavaScript, among others.
- ReactJS –JavaScript
- NodeJS
- NPM
- Bootstrap: buttons, lists, containers, multimedia objects, margins, paddings, among others.
- Blockchain: Blockchain types, block, nounce, transaction, mining, node, confirmation, header of block, hash, identifier, address, consensus, proof of work, 51% attack, EDCSA, crypto, public key, key

• Contracts using Solidity: compiler, compiler versions, OOP, Solidity, states, calls, sends, transactions, functions, anonymous functions, mappings, arrays, visibility modifiers (public, external, internal, private), modifiers, events, function accessibility modifiers (view, pure), address, global variables (msg), payable, gas, limit gas, tokens, public states, packages, import contracts, calls to external contracts, return states , transfer, emit, public keys, own address, standards, convention rules, documentation, constructor, memory, contract deployment, storage, data types, ABI, among others.
• DApps
• Web3Js: providers, calls, sends , among others.
• Something like Truffle or Truffle modifaction: compile, migrate, test, console, migrations, among others.
• Visual Studio Code, Atom, Sublime text
. • IDE Remix

Using Voting App

Now that the respective environment is configured to be able to work with the web page (DApp), i proceed to explain how to use it, and this is what the following section is about. It should be clarified that the authentication of people is not worked in this project, Due to its complexity, it provides material for a complete thesis.

Registering voting tables

Before being able to register voting tables, contracts must be enabled for this mode, in order to allow the registration of tables only, this allows to have a better control to avoid authorizing more tokens to a table, for example, when they are voting. This helps to make the process a little more enjoyable for users and transparent for others interested in the election day. The enabling of the Register of voting tables, begins by entering the main page of the application and accessing the CEU role option.
The different options are only used by the representative of the Electoral Council (representative of the CEU). This, through the options presented, will be limited to enabling voting stages, registering tables, candidates, authorizing votes to the tables and disabling these stages, mainly, although it can also list tables, candidates and the results of the day electoral. In order to enable the registration of voting tables, the option Start table registration is used, which can be seen at the beginning of the work area calls TON SURF PLUGIN, which is the one who manages the account balances to indicate an approximate value of how much this

transaction costs,it is a matter of the CEU representative to authorize plugin to make the expense or not. If user accept, a floating notice will be displayed on the right side of the screen indicating the result of said transaction (authorization).

Authorizing polling station registration (CEU). Notification about the start of registration of voting stations. Own source Continuing with the topic of this section, we proceed to describe how the different voting tables can be registered, so the option - Add voting table. By clicking on this option, the corresponding form is automatically loaded so that the representative can register as many tables as necessary, in order to have control of the tables that are enable and subsequently be able to audit the votes obtained by each candidate at each table.

This only contains the code and the account of the table. To register the accounts of the tables, the following clarification must be made, these, unlike the accounts of the candidates, are not random. These addresses as mentioned above, correspond to the addresses of the deployed contracts and that are in the TON Blockchain, otherwise, the table will not be able to cast votes, as you will not be able to authorize your votes. Once the polling station is registered, the Voting App will display a notification of the registration status.

Otherwise, it may display warnings in orange or errors in red. For the warnings it can be when some special mode is not enabled, for the red color at least that it does not have sufficient funds, it is not authorized. When user are sure that user have registered all the necessary tables, user will be able to close the polling station registration mode to avoid conflicts in the system if a previous mode that user wish to enable is enabled, to do this, on the left side user can select Manage electoral process, this will immediately take user to the page, where user must select the option Finish table registration.

Now that user has all the registered tables, user can list them using the List voting tables option.

It should be noted that it is not necessary to terminate the polling station registration mode, user can list them without any problem with the mode activated.

Registering candidates

In order to register candidates, like the tables, the Voting App must also be enabled in this mode. For this, user must enter the main page of the Voting, enter the CEU role. On this page user must uses the option Start candidate registration When this option is pressed, TON SURF PLUGIN asks to confirm the transaction.

Performing the opinion query

In order to carry out the Opinion Consultation or Election Day, the votes must be authorized at the different voting tables, for this, the main page of the Voting App must be entered and select enter as. On this page, user must go to the option Authorize votes per table that is on the left side. This action will display the following user interface.

In this section user must select the table with the code to which user wants to authorize votes. When selecting the table, to make sure it is the correct table, therefore, the table address is presented on the side. Then enter the number of votes that is equal to the number of registered voters.

Election day now that the tables have the respective votes. To do this, change to the option Manage electoral process, which will change to the page. Once on this new page, user must choose the option Start process electoral, which has the purpose of configuring the contracts to be able to vote, since this option is blocked by default like the rest of the modes. As usual, TON SURF PLUGIN asks to confirm the transaction.

Then, the voting table must go to the main page of the Voting App and enter through the Table role, once user enters will be able to view the page.

This authorizes the voter to cast a vote using the table's account, so their choice of candidate cannot be linked to it. With this confirmation of successful authorization, the voter must be asked to go to the cubicle where they will have an interface to vote, and entering through the Vote option.

Once the electoral day concludes, in terms of time, whoever is in charge of managing the day (assumed the role of CEU), must enter the main page of the Voting App and enter through the role CEU. Once user is on the page mentioned, user must manually end the electoral process, for which, you must choose the option End electoral process.

Consulting results

Once the scrutiny is public, it can be consulted from 3 perspectives which will be presented in this section.

Results for CEU

Whoever is managing the electoral process or who assumes the CEU representative, for this particular step of the case study, can consult the results by entering the main page of the Voting App and entering through the CEU role. Once on this page, user must select the option Consult results, which is where the different candidates will be listed with their respective votes.

General public results

Another way to consult the results and that is of interest to all those people who participate in some way in the electoral process, as well as any other type of list of which is available for the different users of Voting App, this option does not represent any charge in ether unlike the transactions that had been made in previous cases, this is because records are not altered within the system.

Apart from these two forms presented, there is a last way by which the results can be consulted, clarifying that it was intentionally placed in the Voting App through graphical user interfaces.

Auditing results

Auditing the results implies consulting the number of votes that a candidate has had in each of the tables that are available, this is done in order to audit the data and thus generate a traceability of the different results of each candidate in the final scrutiny. This is in terms of the management of Voting App, below are some notes that may be useful to evaluate the technology.

Contacts

Telegram: @ducktalesblock

FREETON Address:
0:c0efbaaa82ea20862cc7b49fdd57288275eae56ff92832c5c948a37943888691