A Large-Scale E-voting System Based on Blockchain

Cosmin-Iulian Irimia
Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iasi, Romania *irimia.cosmin@gmail.com*

Adrian Iftene
Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iasi, Romania *adiftene@info.uaic.ro*

Daniela Gîfu
Faculty of Computer Science, "Alexandru Ioan Cuza" University of Iasi, Romania & Institute of Computer Science, Romanian Academy, Iasi Branch *daniela.gifu@uaic.ro*

Abstract

E-voting systems are increasingly used, considering the various facilities they offer: casting and counting votes in real time. The current voting systems are currently the target of attempted fraud and this is a major problem globally, which has not been solved even to this day. In the field of computer science, these e-voting platforms need to provide integrated security, thus enhancing the scalability and performance of the blockchain-based e-voting system. Our aim is to develop a secure internet-based voting system to maximize user participation, by allowing them to vote from anywhere. This paper proposes a system architecture based on blockchain technology along with a web interface in order to securely authenticate the voters on the platform. It should be noted in addition that these two components can be used together or separately, depending on the application's needs.

Keywords: e-voting systems, blockchain, smart contract, security requirements.

1. Introduction

In democratic societies, political parties aggregate the demands from diverse social groups and articulate public policy options to respond to them [9], [32]. Democracy, as a form of government, has given people the possibility to choose their leaders by vote [11], [15]. This method also attracted many attempts to voting system fraud, which has not been solved entirely. Many studies focused on those security issues [26], highlighting the following challenges of the electronic voting: data integrity, reliability, transparency, the secrecy of the ballot, consequences of breakdown, uneducated voters, specialized IT skills, storage of equipment, security, consequences of fraud and cost [13].

Due to Covid-19, the societies are experiencing an expansion of voting possibilities, known as a hybrid system. Here comes the current paper that aims to solve at least partially this problem, proposing an e-voting system that brings several advantages over the face-to-face one. Many experts consider the paper ballot vote to be the only reasonable way to secure this constitutional right [14]. However, this mode is vulnerable to several errors willingly or not [1]. Due to the advancement of technology, modern day voters wanted to exercise their democratic right online [6], [10]. In both cases, the election fraud has become even more vocal. Several recent examples include the vote fraud controversy in the 2019 elections in North Carolina [8] and the server wipe in the 2017 elections in Georgia [7]. Voters' lack of confidence in the authorities may reach crisis proportions [3], [14].

The main research questions of this paper intend to answer: (1) *How efficient is an e-voting system in order to have a real picture about electors' opinions*? (2) *Can the blockchain concept improve e-voting systems?*

Our research proposes the architecture of a vertical e-voting system based on Ethereum Smart Contracts, called Demochain, which adds new facilities related to the possibility to vote inside of the proposed platform. It allows head-to-tail interaction with the role of ensuring both authentication in a secure system and the process of voting, indexing, counting and validating the vote.

The paper is structured as follows: Section 2 presents a short overview of e-voting systems in order to clarify their importance and what can we do to add new system requirements, while Section 3 refers to the architecture and the design structure of the Demochain e-voting system. Section 4 briefly discusses the evaluation of the platform, through a series of usability tests in which a random group of people, both technical and non-technical, was involved. Section 5 describes a use case based on the elections of representative students in the council of a Faculty of Computer Science, using Demochain e-voting system, before drawing some conclusions in the last section.

2. An Overview of e-voting Platforms

In this section, we introduce existing solutions related to this research in the context of large-scale e-voting systems based on blockchain [34] and consider the main process with this research: elections. In order to increase the trust of the systems used in different fields, a valid solution was and, still is, blockchain. Initially, blockchain technology was used for monitoring cryptocurrency transactions [31], very vulnerable to several fraudulent activities [19], [24]. More specific, it was first used within Bitcoin, and after that was applied in all the cryptocurrencies we see today. To highlight the specificity of our system, this section addresses the following question: *what is a good e-voting system*? Thus, we recall some of the most popular methods of virtual learning platforms. The problem of safe voting is a very old one that dates back to the beginnings of the internet but the first notable attempts were in 2001 in Estonia. Nowadays, especially in the context of the current pandemic, many companies are working to implement such a vertical e-voting system.

2.1. I-Voting

Internet voting or i-Voting is a solution that simply and conveniently helps to engage people in the governance process. Since 2000, U.S has been the first country that has used this method. Afterward, there are several countries who have agreed to this method of voting and who are willing to use it in the future: Estonia, France, Switzerland, Canada, etc. [25]. Of all these, Estonia[1] remains the most relevant example, representing the leading country that uses i-voting system entirely. Let's have a look: 2005 (national elections), 2007 (parliamentary elections), 2011 (Riigikogu elections), 2015 (parliamentary elections) [17]. The main functionality is that it allows voters to cast their ballots from any internet-connected computer anywhere in the world. Of course, being an insufficient tested system, during the Riigikogu Elections of 2011, there were several incidents related to the vulnerability of the applied scheme and the legislative issues. A research program was suggested for i-vote [30]. In order to implement the vote, it was necessary to find optimum between the theoretical security of the voting scheme and the complexity of its implementation [2], [4, 5].

The Estonian solution is simple, elegant and secure, allowing voters to log on and vote as many times as they want during the pre-voting period. With other words, a voter always has the option of changing his or her vote later. Basically, the voter logs onto the system using an ID-card or Mobile-ID, and casts a ballot. Important is the fact, the voter's identity is removed from the ballot before it reaches the National Electoral Commission for counting, thereby ensuring anonymity.

Even if the i-voting solution is a very advanced one, we have identified a general problem, already exposed in the introduction: the fact that both the data and the voting system is on a server is still a weak point that offers a single point-of-entry allowing various types of attacks. The most popular is Distributed Denial-of-Service (DDoS). We also do

[1] https://e-estonia.com/solutions/e-governance/i-voting/

not overlook the fact that it is needed for specialized readers to succeed in authentication, which brings security issues at a lower level, namely at the hardware level. Moreover, several risks are still there: (1) the possibility of malware on the client side machine that monitors the user placing their vote and then later can exchange it with another candidate; (2) an attacker can directly infect the servers though malware being placed on the DVDs used to set up the servers and transfer the votes [29].

2.2. Voatz

For public elections, we will present a private Boston-based company, called Voatz[2] [33], that building and operating another Internet voting system [28], used in high-stakes U.S. federal elections in 2018 [20]. Compared with i-voting system, this method is based on automated facial comparison of a photo of a voter's photo ID to a short selfie video, and a back end virtual ballot box in the form of a closed, permissioned blockchain. It is a smartphone app [21] that use the Jumio KYX Platform to send a photo of her/his driver's license or passport photo page plus a short live stream of the voter's face. Jumio, as a trusted identity platform for businesses, uses machine learning facial comparison software to decide whether the photo of the face on the ID matches the face in that film. If the face is matched, the voter is authenticated and her/his name and address are extracted from the photo ID and returned to Voatz as the true identification of the voter. If the face is not matched, there is apparently a backup human comparison of the photo and video.

From what has been reported, the Voatz system best meets our needs. However, here too we can notice some issues as follows: first, it is based on a closed blockchain method, which insists on security rather than transparency. Second, although the authentication is done in a similar way, by the comparison of the face with the identity card, the application does not ensure that the user is not under someone's pressure or is forced by someone to vote. Third, there is no protection of the anonymity of the vote. In general, a voter can be surrounded by other people who can see her/his vote.

The platforms presented were successfully accepted by those who tested them, and the assessment of the voters who used them demonstrated their usefulness. Even if there are several voices against blockchain technology, all these criticisms are more related to software errors [16], [18], [23]. In this context, we are thinking of the cryptographic and configuration errors, infrastructure problems, web or mobile application vulnerabilities, etc.) [27]. It is worth noting that Blockchain can help ensure privacy.

Our survey is devoted to updating state-of-the-art e-voting systems based on blockchain. In fact, below, we expose a series of system requirements which have not been reported so far.

3. Methodology

The Demochain system aims to provide functionalities for three types of actors: for the *system owner*, for *administrators* and for the *voters*. The *owner* of the system must be able to create a new voting instance, he must have access to all the voting options and have the possibility to add a new voting option, he should be able to start the voting process, must appoint and delete administrators, he can give someone access to the vote, should be able to close the voting process and finally, he can reset the system to its initial state. Like any other user in the system, he can vote, see the list of votes and the results after the voting process is closed like any other existing voter. A system *administrator* must be able to help the owner by giving the voters access to see the options, the votes and the possibility to vote. Like any other user in the system, he can vote, see the list of votes and results after the voting process is closed like any other existing voter. There are several use-cases that any e-voting application must fulfill for a *voter*, the most important would be the voting function itself. In addition, the voter must be able to view all the other votes and finally the election result after the voting process is closed.

3.1. Demochain Functional Requirements

After analyzing the common use cases that need to be solved, the main requirements of the

[2] https://verifiedvoting.org/what-we-dont-know-about-the-voatz-blockchain-internet-voting-system/

application are defined as follows:

1. To offer a *scalable*, *high-performance* solution and *easy to integrate* with different authentication systems.
2. To keep in mind the basics of high quality voting: *anonymity, transparency, security* and *easy to use*.
3. To be *platform-free* so anyone can access it.

Users have multiple features that they want but remember that the two most important are voting themselves and viewing the results. In the following sections it will be described the technical aspects of the solution and how the initial goals are met.

3.2. Demochain Nonfunctional Requirements

Additional to functional requirements we consider a list of nonfunctional requirements:

1. *Performance* - It is very important that the solution to be efficient, no one likes to wait, so the system must respond quickly, even if the transaction has not yet been made or the block has not been included in the system.
2. *Security* is the thing from which we practically start, so the system must certainly cover all operating scenarios and be an example to follow on the security side.
3. *Transparency* is already ensured by blockchain technology anyway [22], but the system must be transparent when it needs to be in situations where the vote is over, so we need some well-defined cases in which to ensure transparency.
4. *Scalability* - The system must be easily scalable because the voting process starts from a few dozen people and can reach hundreds of millions of votes in very large countries.

3.3. System Architecture

The proposed system is a minimalist one, with few dependencies to ensure the best possible experience but with the lowest possible chances of being affected by a vulnerability and consists of three major components, one of which is fixed and the other two are proposed as proof- of-concept and can be easily replaced in the future.

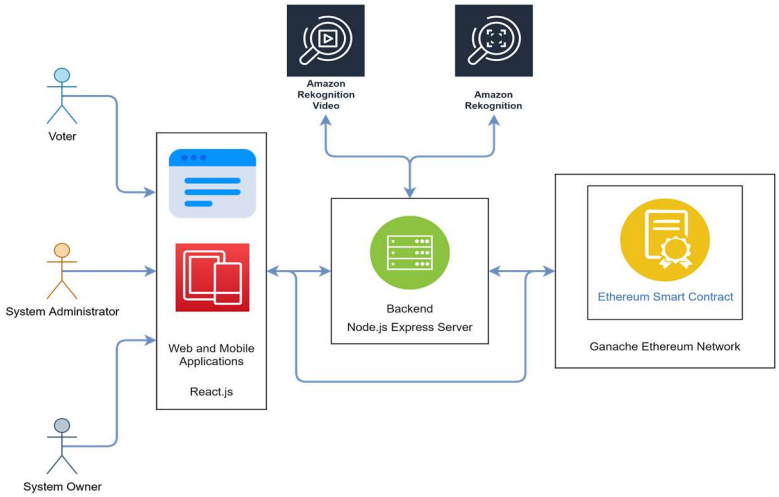

Fig. 1. Demochain architecture.

First of all, we have the *basic component*, from which the work started, and it's the smart blockchain contract using an Ethereum network[3] that will ensure the actual voting, validation and persistence of the executed transactions and then the visualization of final results. This component of the system is paramount and is not replaceable, as it can provide the basic functionalities of an e-voting system alone. In the background we have the *visual component* and the *complementary backend part*. The visual component is one that primarily ensures authentication using human face and emotions identifiers, this being easily replaceable with any other authentication system desired by a potential client whether it is also software-based authentication or even a hardware method.

[3] https://ethereum.org/en/developers/docs/development-networks/

3.4. Blockchain Smart Contract

The smart contract was coded in Solidity[4] and it is written in the most abstract way possible to ensure that regardless of changes in authentication or use-case for which this system will be used, the contract will be able to allow integration.

In order to define the contract, we defined four data structures: *Admin*, *Option*, *Voter* and *VoterDetails*.

- *Admin* is a structure that incorporates a unique identifier that will be used across the system, the name of that administrator to be used in logs, a flag that tells us if it is still active or not (if an admin does not do his job well or correctly, the owner can deactivate it and after this operation they lose all administrator rights and become normal voters) and the last field is that of permissions which currently allows only two functions: owner and moderator. The first position is that of the contract owner, the one who initialized the contract, and the second is that of the moderators who are responsible for offering the right to vote to the voters.
- *Option* is a structure that practically represents the voting proposals. These options do not necessarily have to be people, one can also conceive a vote among the options of a referendum for example. This structure also has a unique identifier, the name of that voting option, then we have the name of the party to which the option belongs (if applicable, if it does not exist, an empty string will be attached) and a voteCount that represents the number of votes accumulated by this option through the votes cast by other users. This value will start at 0 and will be incremented along the way.
- *Voter*, as its name suggests, refers to the data structure that will include users of the system, those who want to vote for a particular option. This structure has a weight field that represents a weight or a percentage of user importance for voting systems where a voter has a more important word than another voter (in the case of democratic voting, this weight is set to 1 for all voters because each voter is equal), a Boolean that indicates whether someone voted or not, then a target that represents the option voted by the user (if applicable) and last but not least a voterDetailsId which is practically a "foreign key" to the VoterDetails structure.
- *VoterDetails* is another structure assigned to the voter and it has a very well established role, to keep hidden all the user's private data from the eyes of strangers. This structure will never be shared, it is useful only for internal use for registration and authentication and contains a unique id of each voter, a uniqueIdentifier that can be the Social Security Number, but also other types of unique identifiers and last but not least, the name of the voter. We consider that these last two details must exist in any context, regardless of whether we are talking about a block-scale vote, a local, county, national or global one.

3.5. System States

Because the contract is on an Ethereum network, it has the property of immutability, which means that it cannot be changed in any way once deployed. Also, the transactions are not reversible or modifiable, therefore, the system must go through several states. To achieve this, we will need global variables and several types of data and collections to help us transition between these states.

After the contract has been deployed on the Ethereum network and then initialized, it is in the *initial state* where it can only be accessed by the owner who has full power over its functionality. In this state, the owner can add voting options, add administrators, and deactivate an administrator in case a mistake has occurred, these three operations being auxiliary but the primary operation he can perform is the transition from the initial state. After the voting process is started and we are in the *voting state*, all administrators enter into their rights, having the opportunity to give the right to vote to any user who registers in the system. Administrators also have the right to vote like any normal user. In this state of the system also intervene the users who must follow a well-established flow, namely must register in the voting program, their data will be verified by an administrator, then

[4] https://docs.soliditylang.org/

they will be able to access the "voting" method that allows them to select the desired option. After performing this operation, the voters will not be able to access any other option until the contract will pass to the next state, namely the *closed voting state*. In this state the administrators lose their attributions and can only show normal user behavior and the only two outcomes are the visualization of the votes, an operation that can be performed by any of the three actors and the operation of closing the system that can be performed only by to the owner and this will stop all communication channels of the other two actors. The last operation that can be performed is to reset the contract, and this will reset all system variables to default values in order to restart a new voting process.

The transition between the states of the system is made using only two logical variables placed in the contract that only the owner can modify but in a predetermined order, as we explained above.

3.6. Back-end Service

The backend service is basically a web server based on the Express module from Node.js that exposes a series of REST APIs[5] from where the frontend service will be able to request and provide the information necessary to interact with the system.

Based on a modular architecture, Demochain API is a collection of services and submodules accessible through a facade that will deal with the processing of input and output data, and all this exposed by a controller to be publicly accessible (see Figure 2).

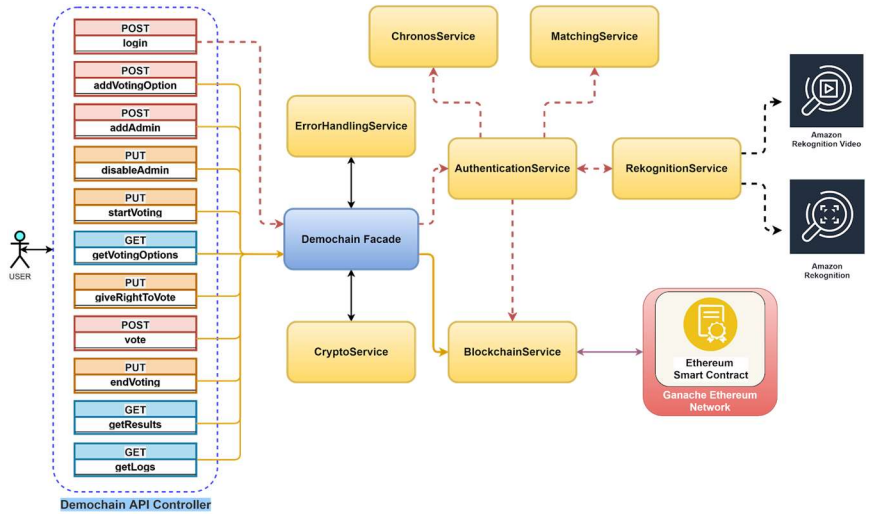

Fig. 2. Demochain API.

Basically, a normal usage scenario is this: a client calls the key exchange endpoint in order to ensure a secure, safe communication, then calls the login endpoint, in the body sending two photos in base64 or URLs from the resource it just uploaded to a bucket from Amazon AWS[6] that we will discuss when we get to the web service. The resources will be two photos, one with the user and one of his identity card. After executing this request, the controller will intercept the body and will call the specific method from Facade which will in turn call AuthenticationService. This service has a very important role, namely it is a mediator between three major components with which it interacts in a well-established order: the first time will be called RecognitionService, the service responsible for communicating with AWS Rekognition. Basically it will extract the relevant data from the body of the request, namely the resources we discussed above, it will send it to AWS for analysis and when the answer is ready, it will be returned back from RecognitionService to AuthenticationService. With this important data, AuthenticationService will determine if the person in the portrait photo is the same as the person in the ID card. If this is not fulfilled, a specific error will be answered to the client, but if this is still ok, it will be passed on in the data evaluation chain, namely: the existence of other people in the framework will be verified to establish some influences on the voter, the expression of the voter's face

[5] https://aws.amazon.com/rekognition/

will also be analyzed to determine if the user seems in a dangerous situation (he is under the threat of a weapon, he is blackmailed, etc.). All these checks are eliminatory and errors will be thrown in any negative case explained above.

Then, taking the data from the identity document using Optical Character Recognition algorithms, establishing the identity and personal data of the voter. Using the existing data we will call ChronosService which will match the CNP, the date of birth and the date of issuance of the document and after this verification the user's age will be identified and it will be determined if the user's age is appropriate and he has the right to vote.

Finally, after all these checks, if all the conditions are met, AuthenticationService will call BlockchainService which will generate a new address for the new user and in turn will call the registerToVote method from the contract. Finally, after the response from the contract, the address generated in response to DemochainFacade will be returned, which will encrypt it using a made-in-house algorithm and together with a newly generated public key, it will be packaged and returned to the client. This is practically the most complex operation that can be performed using DemochainAPI, all other endpoints behave identically: the client generates a request, encrypts the data with the public key received from the server and sends it to the API. When it receives this data, DemochainFacade will pass the request through CryptoService which will decrypt and recover the data and then directly through BlockchainService it will interact directly with the smart contract.

3.7. Front-end Service

The frontend application is certainly more important for users than the smart contract because it is the first layer they encounter when interacting with Demochain. Technically speaking, the application is written in React[6] and we chose this path because it offers extraordinary flexibility, is easy to write and modify, very versatile and also offers all the features we needed to be able to build what we needed. The whole application from head to tail was thought to be very minimalist but functional at the same time so as not to unnecessarily complicate the screens and to be so intuitive that it can be used by people of all ages, because the target audience is extremely varied.

The first screen is very simple, it contains a navigation bar with a logo, the title of the application and in the upper right corner a log button and in the center we have three cards with images and informative information about what the application can do, which are its benefits and how we can use it. The only option that the user can access is the login part where he will receive a prompt to provide access to the camera because it is needed at this stage. Once the camera is open, the user will record his face for a few seconds and then will be asked to take another picture of his card id to be scanned. This data will be loaded on the AWS S3, will be processed by the backend and will generate a unique token and then will be deleted so as not to compromise the user's security. However, on the login side, the system will implement an in-house handshake mechanism that will handle the secure transmission of data.

The backend will return an object that contains the user's identity if it has been successfully authenticated or an error message that will be displayed on the screen if the user has not been recognized, has no voting rights or has not submitted all the data. In case of a successful authentication, the application already works on two cases: if the user is a voter, he will have two buttons on the screen: the voting button and the button for viewing the results, we discuss them in detail immediately; but if the user is an administrator or owner, he will receive an additional control, namely a management button.

On the *voting page*, all users will see the voting options along with data about those options, be they politicians or votes of another nature. These options also have images, even a field that specifies the party from which it is negotiating if necessary. At the bottom, after the user has chosen his option, he will find a voting button that if he presses and confirms the operation, then an http call will be made to the backend that will record that vote. On the *results view page*, users will have a list of options that could be voted and next to each will be displayed various data and statistics on registered votes. Also, on this page

[6] https://reactjs.org/

we will see general statistics, the total number of votes and the winner of the elections. On the *management page* where only the administrators and the owner can reach, things are a little more complicated because it works differently depending on the user and the state of the system. If the voting has not started yet, the owner can add new voting options in a list together with the details related to them and respectively appoint new administrators, instead the admin in this phase cannot do anything. If the voting has started, the owner can still administrators but also delete them from the system if he deems it appropriate and then they have the permission to see data about users in turn and to offer them or not the right to vote; the latter can also be done by the appointed administrators. When the voting process is ready, the owner can reset the contract to its original state and the administrators lose all their rights.

3.8. Installation and Integration Phases

One of the main advantages of the proposed solution comes from the fact that a third person or even an institution could take over the Demochain system and customize it to their needs. This means that we provide our solution with an introductory tutorial for how we can use the application in its current form but also how to integrate other systems to interact with the smart contract.

4. Usability Testing

In order to realize the effectiveness and ease of use of the application we built, we ran a series of usability tests using a random group of people, both technical and non-technical. After establishing the target groups we performed usability tests and collected end-users opinions about Demochain application (both from voters and moderators), to see what can be improved or changed in the future of the application.

4.1. Methodology

We conducted a series of tests run in two shifts and these consisted of four main parts: (1) introduction to the features of the application and a small ramp-up session, (2) the actual tasks we offered to users, (3) a small interview session and (4) in the end a post-test questionnaire. We instructed the participants to think out loud and express their thoughts during the test. After the task series that we communicated verbally to the participants, we gathered their assessment of the overall experience using the QUIS (The Questionnaire for User Interaction Satisfaction) scale. In order to make sure that the whole application is tested, we divided the group of people into two small groups, one of technical persons and another non-technical person group and we offered those tasks in two turns, both voter tasks and moderator tasks. It took about 4-5 minutes for the voter tasks and around 9-10 minutes for the moderator tasks.

4.2. Participants

We collaborated for evaluation with two different groups: one technical and one non-technical.

Table 1. Participants' opinions.

Opinions	Technical	Non-technical
The system is very useful and friendly (especially as the part of voting can be called from a distance and helps to know faster the final score).	x	x
Those who gave up paper voting noticed that online voting is much more efficient and rapid.	x	
Internet voting could change the sociodemographic and ideological composition of the voting electorate.		x
Internet voting can be influenced by Social Media.		x
Internet voting will play an important role in revitalizing the voters, especially the undecided.	x	x
Demochain is a scalable, simple and secure framework for an i-voting process.	x	

The first group, the technical one, is made up of 30 software developers, one of them knowing the concepts of blockchain. Their selection was random, the group being formed of 20 men (between 20 and 28 years old) and 10 women (between 21 and 26 years old).

All of them have previous experience with web applications and in use but also in writing their code. They received the Demochain application, used it and tested it for several hours.

The second group, which is non-technical, consists of 30 other people, 20 around 20-year-old students from the faculties of letters and economics, 10 around 40 and 50 years old, which are one housewife, one professor, 3 economists and 5 engineers, this matters because, it will prove later how easy to use Demochain. All of them have previous experience with web applications but none have concepts about programming. They received the Demochain application, used it and tested it for several hours and commented on it (see Table 1).

4.3. Performed tasks

In cases of voter tasks, users (both technical and non-technical) have executed the steps without having seen them executed in advance by someone. This shows in itself the ease with which users can interact with the Demochain system. For the moderator tasks, all the non-technical people needed help but they performed the tasks successfully after someone else went through the steps, presenting them but for most of the technical people these tasks were also performed very easily.

There were three tasks for each type of actor: for the *voter*, users had to register in the application, then vote for one of the two preferences offered and at the end of the vote to follow the results. For the *administrator*, the users had to login and navigate in the management menu, to view the data of a certain user in order to approve it in the system and respectively they had to offer the actual access to the vote to a certain user.

4.4. Results

From our observation during the test sessions, all the people involved were very satisfied with the application and say that they would use the Demochain application to the detriment of the classic vote if this were an option (see Fig. 3). They easily found the controls and successfully used the interface that was addressed to them. Because the application is minimal, the execution of tasks was fast and the application did not crash during the development of tests. The participants were asked to rate their experience with a note from 1 to 5, where 1 stands for confusing/frustrating experience, and 5 for clear/pleasant experience, the score obtained was 4.25.

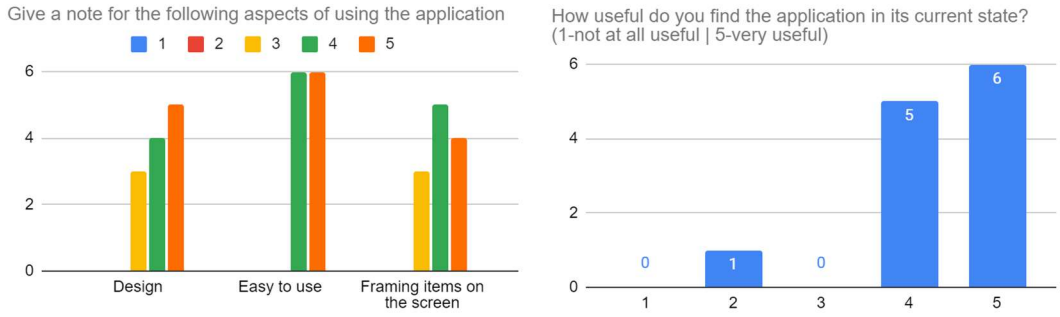

Fig. 3. Answers received by users regarding the usability of the application.

The user responses to the post-test questionnaire show that the “easy to use” and “framing items on the screen” (Fig. 3) were the most appreciated features by participants. Also, five out of six users consider Demochain to be superior to the current voting system. Users appreciated that the solution is very useful in its current state (4.3 / 5) but also offered constructive feedback such as the need for an interface that models well on mobile phones and the addition of a timer to know how long it remained until the end of the vote.

4.5. Remarks

Comparing the technical users with the non-technical ones, regarding the administration part of the application, we can say that the interface could be improved, it was not first hand for the non-technical ones to use the controls offered and this is a bit natural but with certainly we can make improvements to the system on the visual side.

Regarding the voting part, the users appreciated the ease of use but encountered small problems related to authentication where the system failed to detect a certain person due to the very low light in the room. The problem with low light authentication can be solved by two simple methods, one would be integration with other authentication methods available on modern devices (e.g., Apple's FaceID / Fingerprint Recognition / Windows Hello) or the second method in which we can improve recognition by using devices APIs to automatically increase the brightness of the screens at the time of authentication to bring rays of light on the user's face, this certainly helping the process. Also, in the informal discussions we had with all six subjects, four of them stated that the waiting time from login to identity validation is a bit long and that they would like the automatic realization of this process in future.

In the end, the six subjects were very happy to be part of the use of such a system, everyone liked the idea of voting more often and more from home and the main attraction was the authentication that the system has. In the final discussion all the participants seemed very interested on how we did this without credentials and with great security in mind. One last thing we discussed is related to the desire of users to have this system available in the next elections and here we had a shock: 91.7% of the answers were that people want a system like this and that Demochain is considered a superior solution to the existing solutions. This shows us on the one hand the need of the people to eliminate the problems of the classic vote and on the other hand the positive appreciations for our app.

5. Using the Demochain in a Real Scenario

We perform in 2020 some experiments on the elections of representative students in the council of our Faculty of Computer Science. The problem identified in the case of these elections comes from the fact that undergraduate students (around 1300) and master's students (around 200) participate in a small number in these elections (less than 50 at the bachelor's degree and less than 10 at the master's). Next we will refer first of all to the master's students, where we also experimented with the Demochain application. Because master's students must have a representative on the faculty council, it is necessary that every two years, if not more often, we organize these elections.

Until 2019, these elections were organized at the Faculty in one of the halls during a whole day and it was necessary for the voters to physically participate in them. The elections were organized by student representatives and at least two people were required to be present in the polling station throughout the voting period. Many of the master's students are already employed in IT companies (full-time or part-time) and for this reason they do not manage to get very involved in additional activities at the Faculty.

In recent years, these elections were attended by few voters (in 2016 election year - 7 voters), (in 2018 - 5 voters), (in 2019 - 8 voters). As at least 50% of the voters did not participate, it was necessary to organize two rounds each time. Most of the time in the second round even fewer voters participated (in 2016 in the second round only two students voted). In the end of 2020 election year, it was the first time we organized the elections for master students using the Demochain application, and the statistics changed significantly.

We configured the application on the faculty server, and the students were able to vote during the day. Of the 178 eligible voters, 93 participated, and therefore a single ballot was sufficient. There was no need to print attendance lists and there was no need to block more people from being physically present in the room where the elections would have taken place. This year as in previous years, we sent them two reminder messages to participate in the vote, but this year the messages were accompanied by the web address where they could vote.

During the voting process, the same types of problems were detected as in the case of usability tests (the most common problem being related to the authentication of students in the application). After the voting process, many of the students stated that they curiously accessed the link with the application to see what this application looks like, which uses blockchain at the backend level. Most of them were satisfied with the application and appreciated the fact that they were able to participate remotely in this voting process.

6. Conclusions

An e-voting system it becomes efficient, primarily because it allows to voters to vote from anywhere, from any screen device, even that the security is still partially solved. Our experiments yielded new observations on the overall security, performance, and scalability of blockchain-based e-voting systems. The overall costs for organizing the entire voting process, using Demochain system, are lower and safer: involving a small number of people, reducing travel costs for voting, the human interaction is reduced, etc.

Experiments have shown that the application has been well received by those who have used it, the voters can vote online fast, from anywhere and the administrators can organize anytime a voting process on their own machines. The current problems are related to the authentication part where we want to work more in the future, on the one hand to improve the quality of recognition, and on the other hand to increase the speed of this process. However, we can consider that e-voting has a significant impact in the election process, involving the testing of both voters and administrators to interact with each other.

We believe that an electronic voting system is a superior one in terms of showing a clear picture about the electors' opinions through the transparency and security that it proves and we consider that it is important for such a system to be implemented globally. In addition, the blockchain through its design and protocol, the security it provides and the way the data is stored is the best current solution to improve the electronic voting systems. In short, we believe that our goal has been achieved: this paper shows that a safe and easy to use large-scale e-voting solution by anyone is tangible.

Acknowledgments

This work was supported by project REVERT (taRgeted thErapy for adVanced colorEctal canceR paTients), Grant Agreement number: 848098, H2020-SC1-BHC-2018-2020/H2020-SC1-2019-Two-Stage-RTD.

References

1. Abuidris, Y., Kumar, R., Yang, T., Onginjo, J.: Secure Large-Scale E-voting System Based on Blockchain Contract Using a Hybrid Consensus Model Combined with Sharding. In ETRI Journal, DOI: 10.4218/etrij.2019-0362 (2020)
2. Alboaie, S., Alboaie, L., Pritzker, Z., Iftene, A.: Secret Smart Contracts in Hierarchical Blockchains. In 28th International Conference on Information Systems Development (ISD2019), August 28-30, Toulon, France (2019)
3. Alvarez, R.M., Levin, I., and Li, Y.: Fraud, Convenience, and E-voting: How Voting Experience Shapes Opinions about Voting Technology, J. Inform. Technol.Politics 15, 94-105 (2018)
4. Ansper, A. *et al.*: E-voting Concept Security: Analysis and Measures, Estonian National Electoral Commitee, EH-02-02 (2010).
5. Ansper, A. *et al.*: E-voting Concept Security: Analysis and Measures, Estonian National Electoral Commitee, EH-02-01 (2003)
6. Araújo, R. *et al.*: Remote Electronic Voting Can be Efficient, Verifiable and Coercion-Resistant, in Proc. Int. Conf. Financial Cryptography Data Security (Barbados), 224-232 (2016)
7. Bajak, F.: Georgia Election Server Wiped After Suit Filed (2017).
8. Blinder, A.: Election Fraud in North Carolina Leads to New Charges for Republican Operative (2019)
9. Bull, J., Baker, J., Griffiths, V. F., Jones, J. P. G., Milner-Gulland, E.: Ensuring No Net Loss for People as well as Biodiversity: Good Practice Principles (2018)
10. Daramola, O.: Thebus, D. Architecture-Centric Evaluation of Blockchain-Based Smart Contract E-Voting for National Elections. Informatics 7, 16 (2020)
11. Delmonte, R., Tripodi, R., Gîfu, D.: Opinion and Factivity Analysis of Italian Political Discourse. In: Proceedings of The 4th edition of the Italian Information Retrieval

Workshop (IIR 2013), Pisa, Italy, CEUR-WS on-line proceedings series, 88-99 (2013)
12. Epstein, J.: Are all Types of Internet Voting Unsafe? - IEEE Secur. Priv. 11, 3-4 (2013)
13. Esteve, J.B., Goldsmith, B., Turner, J.: International Experience with E-Voting (2020)
14. Gibson, J.P. *et al.*, A Review of E-voting: the Past, Present and Future, Ann. Telecommun.71, 279-286 (2016)
15. Gifu, D.: The Discourse of the Written Press and the Violence of Symbols, PhD thesis. "Alexandru Ioan Cuza" University of Iaşi (2010).
16. Goodman, R., Halderman, J.A.: Internet Voting is Happening Now. (2020)
17. Heiberg, S., Laud, P., Willemson, J.: The Application of I-Voting for Estonian Parliamentary Elections of 2011. Proceedings of the Third international conference on E-Voting and Identity, DOI: 10.1007/978-3-642-32747-6_13 (2011)
18. Juels, A., Eyal, I., Naor, O.: Blockchains Won't Fix Internet Voting Security could Make it Worse (2020)
19. Karame, G.O., Androulaki, E., Capkun, S.: Double-spending fast payments in bitcoin," in Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS), Raleigh, NC, 906-917 (2012)
20. Kirby, J.: West Virginia is testing a mobile voting app for the midterms. What could go wrong? (2018)
21. Kosoff, M.: "a horrifically bad idea": Smartphone voting is coming, just in time for the midterms (2018)
22. Marr, B.: A Very Brief History Of Blockchain Technology Everyone Should Read. Retrieved 26, 8 (2018)
23. Park, S., Specter, M., Narula, N., Rivest, R.L.: Going from Bad to Worse: From Internet Voting to Blockchain Voting (2020)
24. Pérez-Solà, C., Delgado-Segura, S., Navarro-Arribas, G., Herrera-Joancomartí, J.: Double-spending prevention for bitcoin zero-confirmation transactions," IACR Cryptology ePrint Archive, 2017, 394 (2017)
25. Porup, J.: Online voting is impossible to secure. So why are some governments using it? Retrieved 22 8 (2018)
26. Ryan, P.Y.A., Schneider, S., Teague, V.: End-to-End Verifiability in Voting Systems, from Theory to Practice. IEEE Secur. Priv. 13, 59–62 (2015)
27. Saad, M., Spaulding, J., Njilla, L., Kamhoua, C, Shetty, S., Nyang, D., Mohaisen, A.: Exploring the Attack Surface of Blockchain: A Systematic Overview (2019)
28. Specter, M.A., Koppel, J., Weitzner, D.: The Ballot is Busted Before the Blockchain: A Security Analysis of Voatz, the First Internet Voting Application Used in U.S. Federal Elections (2018)
29. Springall, D., Finkenaur, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., Halderman, J.A.: Security Analysis of the Estonian Internet Voting System (2014)
30. Tammet, T., Krosing, H.: E-valimised Eesti Vabariigis: võimaluste analüüs (E-voting in Estonia: feasibility study), Analysis ordered by Estonian Ministry of Transport and Communications. In Estonian (2001)
31. Taş, R., Tanrıöver, Ö. Ö.: A Systematic Review of Challenges and Opportunities of Blockchain for E-Voting. In Symmetry, 12(8), 1328; DOI: 10.3390/sym12081328 (2020).
32. Wantchekon, L.: Policy Deliberation and Voting Behavior: Evidence from a Campaign Experiment in Benin. Princeton University: unpublished (2013)
33. Weiss, M., Halyard, M.: Voatz. Harvard Business Review. Case Study (2019)
34. Zhang, S., Wang, L., Xiong, H.: Chaintegrity: Blockchain-enabled large-scale e-voting system with robustness and universal verifiability. Int. J. Inf. Secur. 19, 323–341 (2020)