

UNIVERSITATEA POLITEHNICA DIN BUCUREŞTI
FACULTATEA DE AUTOMATICĂ ŞI CALCULATOARE
DEPARTAMENTUL DE CALCULATOARE



PROIECT DE DIPLOMĂ

PoliVote - Sistem de vot peste blockchain
Iulie 2023

Gabriel Poalelungi

Coordonator științific:

Prof. dr. ing. Ciprian-Mihai Dobre

BUCUREŞTI
2023

UNIVERSITY POLITEHNICA OF BUCHAREST
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT



DIPLOMA PROJECT

PoliVote - Blockchain voting system
July 2023

Gabriel Poalelungi

Thesis advisor:
Prof. dr. ing. Ciprian-Mihai Dobre

BUCHAREST
2023

CUPRINS

1 Introducere	1
1.1 Context	1
1.2 Problema	2
1.3 Obiective	3
1.4 Soluția propusă	4
1.5 Rezultatele obținute	5
1.6 Structura lucrării	5
2 Analiza Cerințelor / Motivație	6
2.1 Motivație	6
2.2 Analiza cerințelor	8
2.3 Scenarii de utilizare	9
3 Studiu de Piață / Metode Existente	10
3.1 State of the Art	10
3.2 Tendințe viitoare	12
3.3 Abordări existente / Studiu de piață	12
3.4 Tehnologii utilizate	16
4 Soluția Propusă	17
5 Detalii de implementare	19
5.1 Aplicația Web - Componenta Front-end	19
5.1.1 Perspectiva utilizatorului normal	19
5.1.2 Perspectiva administratorului	24
5.2 Componenta Server	25
5.2.1 Înscrierea și validarea utilizatorilor	25

5.2.2	Începerea și sfârșitul procesului electoral	26
5.2.3	Comunicarea cu smart-contract-ul	26
5.2.4	Decriptarea, validarea și numărarea voturilor	29
5.2.5	Stocarea datelor	30
5.3	Componenta blockchain	31
5.3.1	Blockchain, Ethereum, Solidity și Smart-Contracts	31
5.3.2	Implementarea smart-contract-ului electoral	33
5.3.3	Reteaua blockchain - Ethereum	38
6	Evaluarea sistemului de vot	40
7	Concluzii	44
7.1	Dezvoltări ulterioare	44

SINOPSIS

Democrația reprezintă cea mai răspândită formă de guvernare de pe planetă, iar unul din principiile de bază ale acesteia este actul de a vota. Având în vedere importanța votului, este crucial ca acesta să fie organizat cu toate măsurile de siguranță necesare pentru a asigura numărarea corectă a voturilor, eligibilitatea persoanelor care pot vota și prevenirea oricărei forme de discriminare. Problema în societatea modernă este că realizarea la scară mare a unui proces de vot absolut corect este complexă și costisitoare. Angajarea de personal, păzirea urnelor de vot, închirierea spațiilor de votare, utilizarea urnelor de vot mobile, numărătoarea fizică a voturilor, dovedirea unei numărători corecte aduc din urmă atât costuri ridicate, cât și posibilități de fraudare, de incorectitudine sau discriminare. În această lucrare, ne propunem să prezintăm o soluție alternativă de vot care să asigure transparentă, siguranță și echitabilitate la costuri reduse. Soluția propusă prezintă un sistem de vot electronic care le permite persoanelor eligibile să voteze informat și convenabil, să monitorizeze progresul votului și să verifice rezultatele odată cu încheierea acestuia. Voturile sunt stocate într-o retea blockchain, făcând astfel fraudarea imposibilă, iar rezultatele fiind disponibile tuturor odată cu încheierea votului.

ABSTRACT

Democracy represents the most widespread form of governance on the planet, and one of its fundamental principles is the act of voting. Given the importance of voting, it is crucial for it to be organized with all the necessary safety measures to ensure accurate vote counting, eligibility of voters, and prevention of any form of discrimination. The issue in modern society is that implementing a large-scale, completely fair voting process is complex and costly. Hiring personnel, securing the ballot boxes, renting voting spaces, using mobile ballot boxes, physically counting the votes, and ensuring accurate tallying all entail high costs and carry the potential for fraud, inaccuracies, or discrimination. In this thesis, we aim to present an alternative voting solution that ensures transparency, security, and fairness at reduced costs. The proposed solution is an electronic voting system that allows eligible individuals to vote conveniently and securely, monitor the progress of the voting process, and verify the results upon its completion. The votes are stored in a blockchain network, rendering fraud impossible, with the results made available to everyone once the voting concludes.

MULTUMIRI

Doresc să îmi exprim recunoștință și aprecierea față de Prof. dr. ing. Ciprian-Mihai Dobre care a propus această temă interesantă și de actualitate și care mi-a pus la dispoziție toate materialele și sursele de informare de care aveam nevoie în realizarea întregului proiect.

1 INTRODUCERE

PoliVote este un sistem de vot electronic ce se folosește de tehnologia blockchain pentru a oferi o cale de susținere sigură, ieftină și ușor de utilizat a scrutinelor de vot, apropiindu-se cât mai mult de idealul unui vot absolut corect, echitabil și nediscriminatoriu. Prin intermediul acestuia, orice om își poate exprima liber votul, lipsit de orice fel de constrângere și de orice teamă a fraudării rezultatelor.

Sistemul de vot se folosește de toate părghiiile tehnologiei blockchain, printre care transparenta, disponibilitatea, imuabilitatea și descentralizarea mediului de stocare pentru a oferi o modalitate cât mai sigură și simplu de utilizat pentru susținerea unor procese electorale. Din partea organizatorului, este necesară existența a câtorva persoane de încredere care să modereze scrutinul de la înscrisarea votanților până la afișarea rezultatelor, iar din partea votanților, este necesară doar o conexiune stabilă la internet și un dispozitiv electronic prin care pot accesa aplicația web.

În acest capitol, voi prezenta o scurtă descriere a sistemului, discutând punctele tari și slabe ale acestuia, ce beneficii aduce în rândul proceselor democratice, dar și ce niveluri de risc prezintă și cum pot fi acestea atacate.

1.1 Context

Exprimarea votului reprezintă pilonul principal al democrației de la inventarea acesteia în urmă cu mii de ani în Grecia Antică până în zilele noastre. Majoritatea țărilor sunt construite pe baza unor instituții ce supun la vot orice decizie ce trebuie luată spre binele acestora, votanții fiind fie cetățenii, fie membrii unor comisii care au fost alesi la rândul lor prin vot. De asemenea, procesele electorale sunt luate deseori în cadrul comunităților sau întreprinderilor unde se supun la vot atât funcții, cât și decizii.

Odată cu evoluția democrației și a creșterii volumului de persoane ce sunt nevoie să își exprime votul în legătură cu diverse decizii, apare o creștere a complexității modului în care se organizează astfel de alegeri. Cu cât este mai important scopul votului, cu atât este mai dificilă împiedicarea fraudării, iar cu cât numărul votanților este mai mare, cu atât este mai dificilă implementarea unui sistem rapid, ieftin și care oferă sansă la vot tuturor.

Au existat diferite metode de eficientizare și de reducere a discriminării alegerilor, precum prelungirea zilelor în care se poate vota, urne mobile, vot prin poștă, dar acestea aduc din urmă costuri ridicate și dificultăți în contabilitatea voturilor. De asemenea, au existat tentative ale multor țări de a implementa și folosi la scară largă sisteme de vot electronice, precum

Estonia, Brazilia, India, Olanda, Germania sau Irlanda¹, dar au întâmpinat probleme legate de lipsa transparentei și a posibilității fraudelor sau manipulării voturilor. Aceste sisteme au fost încercate începând cu anii 2000 până în prezent și, în același timp, a fost inventată și dezvoltată tehnologia blockchain, un concept ce a promis înlăturarea entităților terțe din cadrul tranzacțiilor de bunuri și servicii.

Conceptul de blockchain a apărut în anii 1990, iar dezvoltarea tehnologiei a luat ampolare începând cu 2008, odată cu inventarea *Bitcoin* [19]. În 2014, s-a dezvoltat următoarea generație de blockchain reprezentată de *Ethereum* [20]. Aceasta a fost construită pentru dezvoltarea de contracte inteligente a căror menire este înlăturarea autoritaților terțe pentru intermedierea acordurilor între alte două entități.

Ethereum a fost folosit la scară largă pentru aplicații de tipul *DeFi (Decentralized Finance)*, *Non-Fungible Tokens (NFTs)* sau pentru oferirea de soluții de verificare a identității, folosite cu scopul de a controla originea și identitatea datelor personale, precum documente medicale sau documente oficiale, reducând nevoia existenței unui sistem centralizat și, astfel, îmbunătățind nivelul confidențialității.

PoliVote este un sistem de vot electronic accesibil din browser care utilizează un contract intelligent peste Ethereum pentru a stoca voturile într-un mod transparent, sigur și care nu permite manipularea acestora sub nicio formă. Platforma este user-friendly și permite cu ușurință exprimarea votului de la distanță oricărui utilizator eligibil.

Proiectul a luat naștere datorită dorinței mele de a dezvolta un sistem electronic de vot ce permite desfășurarea procesului electoral la costuri reduse într-un mod cât mai aproape de idealul democrației.

1.2 Problema

Problemele principale cele mai discutate cu privire la modalitățile tradiționale de vot sunt cele ale transparentei, manipulării și costurilor voturilor datorită complexității sub care se desfășoară procesele electorale.

În prezent, se încearcă atacarea acestor probleme prin diferite modalități precum acordarea de fonduri pentru creșterea personalului responsabil atât pentru numărarea voturilor, cât și pentru supravegherea acestui proces. De asemenea, s-au creat modalități de vot la distanță, cum ar fi cel prin poștă sau implementarea urnelor mobile, pentru a permite persoanelor cu dizabilități să își manifeste dreptul la vot. Toate acestea duc la o creștere considerabilă a costurilor pe măsură ce se dorește o siguranță și transparentă mai mare.

¹<https://hackernoon.com/which-countries-are-casting-voting-using-blockchain-s33j34ab>

Pe lângă faptul că aceste metode generează creșteri ale costurilor, ele conduc și la un timp de desfășurare îndelungat și la apariția unor probleme suplimentare, cum ar fi ocuparea spațiilor care ar fi putut fi utilizate în alte scopuri, creșterea traficului de mașini sau utilizarea ineficientă a personalului disponibil, care ar fi putut fi folosit mai eficient în alte activități.

O soluție alternativă la votul tradițional este cel electronic, dar acesta vine și el cu probleme ce țin de transparentă și fraudă. Deși sistemele electronice reduc semnificativ costurile de organizare, nu garantează integritatea voturilor, transparenta cu care acestea au fost stocate și numărate și disponibilitatea acestora în cazul unei eventuale defecțiuni tehnice severe. De asemenea, sistemele de vot electronic permit manipularea votanților mult mai ușor de către persoanele rău intenționate și pot scurge informații vitale, precum rezultate partiale, ce ar putea avaria procesul electoral.

O altă problemă legată de folosirea unor astfel de sisteme electronice de vot este nivelul de cunoaștere a utilizării tehnologiei sau, invers spus, gradul de analfabetism digital. Un procent mare de populație privește cu teamă o astfel de modalitate din cauza avansului tehnologic rapid față de care nu pot ține pasul fără o informare temeinică prealabilă. Oamenii reușesc să se acomodeze cu digitalizarea proceselor instituțiilor guvernamentale, precum plătirea online a taxelor sau depunerea documentelor în format digital, sau proceselor financiare, precum controlul și transferul banilor, dar această acomodare se dovedește a fi o luptă îndelungată și trebuie găsite soluții pentru scurtarea acesteia.

Prin aplicația web PoliVote, se dorește combaterea tuturor problemelor menționate anterior atât prin utilizarea beneficiilor aduse de tehnologia blockchain, cât și prin construirea unei interfețe prietenoase cu utilizatorul astfel încât acomodarea cu aceasta să fie făcută într-un timp record.

1.3 Obiective

Scopul principal al lucrării este de a aduce o nouă perspectivă asupra evoluției procesului electoral, cu accent pe utilizarea tehnologiei blockchain ca ambasador al următorului nivel al internetului, în ceea ce privește desfășurarea scrutinelor de vot.

De asemenea, vor fi evidențiate și următoarele obiective:

- Dezvoltarea unui sistem de vot electronic de încredere ce poate suporta un proces electoral de mare anvergură, demonstrând fiabilitate și scalabilitate.
- Asigurarea confidențialității datelor utilizatorilor prin anonimizarea voturilor, neexistând o legătură între aceștia și, mai mult decât atât, asigurarea utilizatorului cum că votul său a fost înregistrat în mod corect și predictibil.
- Evidențierea disponibilității voturilor stocate în rețeaua blockchain și transparenta tranzacțiilor efectuate.
- Imposibilitatea fraudării sau manipulării voturilor prin controlarea temeinică a tranzacțiilor

efectuate și păstrarea în secret a rezultatelor intermediare.

- Numărarea corectă și transparentă a voturilor, precum și punerea la dispoziție a acestora pentru orice entitate care dorește să calculeze rezultatele finale.
- Dezvoltarea unei interfețe user-friendly ce permite utilizatorilor cu minime cunoștințe digitale să își exprime votul.

1.4 Soluția propusă

PoliVote își propune să rezolve o serie de probleme ce au afectat în diferite moduri orice sesiune de alegeri electorale la scară mare folosind pârghiile oferite de tehnologia blockchain, în special platforma Ethereum. Dacă înainte era necesară o autoritate terță în care se punea toată încrederea pentru a asigura un scrutin corect și nepărtinitor și care asigura integritatea voturilor, acum această nevoie este satisfăcută de folosirea contractelor inteligente ale Ethereum-ului care execută instrucțiunile specifice unui proces electoral doar sub anumite condiții și fară compromisuri.

Chiar dacă sistemul de vot propus pare a fi aproape de un caz ideal, acesta vine la pachet cu diverse riscuri și provocări. În primul rând, utilizarea rețelei oficiale Ethereum poate reprezenta o sursă de costuri mari în cazul utilizării ei pentru un scrutin de vot la scară largă, deoarece costul tranzacțiilor este nu numai dependent de tipul instrucțiunilor executate, dar și de cererea și oferta din rețea. Astfel, există șansa creșterii taxelor de procesare datorită volumului mare de tranzacții concomitente. În al doilea rând, cheia de decriptare a voturilor trebuie ținută secretă cu cea mai mare precauție până la terminarea procesului electoral. Autoritatea organizatorică este responsabilă de moderarea momentelor votului și de asigurarea faptului că nimeni nu poate calcula rezultatele finale decât după eliberarea acestei chei. În acest sens, trebuie aleasă o comisie de maximă încredere ce detine competențele necesare pentru o astfel de responsabilitate. Nu în ultimul rand, utilizatorii ce vor să își exprime votul pot manifesta o temere într-o măsură sau alta față de accesarea unei astfel de platforme. În acest sens, s-a încercat pe cât posibil ca platforma să fie cât mai prietenoasă, iar explicațiile despre cum funcționează să fie cât mai ușor de înțeles.

Există posibilitatea ca unele persoane să fie constrânse să voteze pe un candidat anume sau chiar să fie rău intentionate și să voteze în numele altor utilizatori. Pentru această problemă, s-a implementat anonimizarea utilizatorilor, astfel încât un vot să nu poată fi potrivit cu cel ce l-a exprimat, iar utilizatorul să primească doar o confirmare a votului, nu și valoarea acestuia. De asemenea, autoritatea organizatorică validează înscrierea utilizatorilor, iar lista finală este transmisă contractului intelligent, eliminându-se astfel posibilitatea efectuării acțiunilor malicioase precum adăugarea voturilor invalide sau votarea în numele altor persoane.

1.5 Rezultatele obținute

Implementarea soluției propuse a dus la obținerea unor rezultate mulțumitoare ce întăresc încrederea în utilizarea unui astfel de produs în cadrul unor procese electorale de orice anvergură.

Contractul intelligent al sistemului de vot este unul securizat, nu prezintă scurgeri de informații, funcționează după un set definit de pasi și după un set strict de reguli și condiții și nu prezintă erori de procesare. În cazul serverului care se ocupă de moderarea momentelor votului și validarea înscrerii utilizatorilor, este capabil să proceseze o masă mare de cereri de înscrisie fară să comită vreo greșală ce ar putea duce la o invaliditate a scrutinului, este rezistent la încercări de manipulare a momentelor votului și este capabil să calculeze rezultatele finale cu precizie absolută într-un timp suficient de rapid.

1.6 Structura lucrării

Structura lucrării este următoarea:

- **Introducere** : secțiune ce prezintă contextul soluției propuse și problema rezolvată. De asemenea, sunt descrise succint obiectivele și rezultatele obținute.
- **Analiza Cerințelor / Motivație** : secțiune ce prezintă motivația care a stat la baza dezvoltării sistemului de vot, analiza cerințelor funcționale și non-funcționale, precum și cazurile de utilizare.
- **Studiu de Piață / Metode Existente** : secțiune ce prezintă soluțiile existente ce încearcă să rezolve problemele actuale ale scrutinelor de vot, precum și tehnologiile utilizate în proiect.
- **Soluția propusă** : secțiune ce prezintă panorama mai detaliată a arhitecturii sistemului, precum și asupra funcționalitătilor acestuia.
- **Detalii de implementare** : secțiune ce prezintă detaliile de implementare ale fiecărei funcționalități, alături de descrierea componentelor participante și o prezentare mai tehnică asupra tehnologiei blockchain utilizate
- **Evaluare** : secțiune ce prezintă o analiză a acurateții și a acceptanței din partea publicului.
- **Concluzii** : secțiune ce oferă un cuvânt de încheiere a prezentării sistemului de vot, o perspectivă cuprinzătoare cu privire la rezultatele obținute, cât și posibilitățile de îmbunătățire ale aplicației

2 ANALIZA CERINȚELOR / MOTIVATIE

În acest capitol, voi expune motivația care a determinat dezvoltarea sistemului de vot electronic, cât și analiza cerințelor de funcționare alături de scenariile de utilizare.

2.1 Motivație

Odată cu evoluția omenirii și a comunităților în care deciziile sunt luate la comun, s-a dezvoltat și anvergura proceselor democratice, în special votul. Acum 2000 de ani, în Grecia Antică, alegerile [6] se desfășurau la scară mult mai mică și cu reguli mult mai simple. Din cauza modului în care era formată societatea, numărul oamenilor eligibili era destul de mic comparativ cu întreaga populație. Mai mult decât atât, dintre aceștia erau aleși la întâmplare cei care își exprimau votul la un moment dat pentru a asigura faptul că cei ce aveau puterea nu erau doar cei mai înstăriți. Scrutinele de vot erau realizate într-un mod transparent și echitabil prin alegerea a diferiți jurați din cei eligibili de a vot. Astfel, orice moment al alegerilor era urmărit cu scopul de a asigura numărarea corecta a voturilor și de a combate manipularea acestora. În final, scrutinele în sine constau într-o adunare în aer liber a oamenilor unde li se prezintau deciziile ce trebuie luate. Fiecare decizie se supunea la vot, iar fiecare persoană ridică o mână dacă era "pentru" sau o ținea jos pentru "împotriva". Jurații numărau voturile și anunțau rezultatul pe loc. O variantă mai puțin răspândită erau deciziile luate prin nivelul de zgomot produs de către adunare. Votanții erau rugați să aplaude sau să țipe pentru varianta preferată a unei decizii, rezultatul fiind varianta cu cel mai mare nivel de zgomot.

Aceste moduri de a desfășură procesele electorale au funcționat până la apariția diferențelor între clasele sociale, creșterea numărului de participanți și măsluirea voturilor sub orice mod posibil de către lideri, fenomen care a început încă din Imperiul Roman. Deși au existat diferite modalități de a lua măsuri pentru a păstra echitabilitatea în luarea deciziilor de acum două mii de ani până în prezent, niciuna nu a putut să asigure atât corectitudine, cât și transparentă la costuri reduse.

Evoluția constantă în domeniul tehnologiei joacă un rol extrem de important atât în evoluția oamenilor în plan individual, cât și în evoluția omenirii, la modul cum funcționează societățile și comunitățile pentru a-și desfășură activitatea. Putem observa beneficiile aduse de digitalizare oriunde în jurul nostru, de la plata sigură online a taxelor sau depunerea documentelor în format digital până la răspândirea rapidă a informațiilor de interes public. Deși această digitalizare reprezintă un drum anevoie pe care îl parcurg atât oamenii, cât și instituțiile sau companiile, este un pas înainte spre o varianta mai bună a societății zilelor noastre.

Un domeniu interesant în care s-a încercat digitalizarea a fost chiar domeniul proceselor electorale de orice tip, de la cele naționale până la cele comunitare sau private. Deși crearea unor sisteme de vot electronice reduc semnificativ costurile și își propun să aducă echitabilitatea ideală a democrației, nu este o sarcină tocmai ușoară. Există multe țări în care s-a încercat această trecere de la votul în urnă la votul electronic, cum ar fi Brazilia, Irlanda sau Estonia, dar au existat multe dubii legate de corectitudinea și transparenta acestora, nu numai din partea cetățenilor, dar și din partea specialistilor. Elementul comun al tuturor problemelor este faptul că digitalizarea scrutinelor de vot nu înălță nevoia existenței unei entități terțe care să aibă control total asupra a ceea ce se întâmplă și, de aceea, se află în desfășurare o multitudine de cercetări pe această temă.

O soluție promițătoare o constituie tehnologia blockchain, în special contractele inteligente dezvoltate pe platforma Ethereum, care au fost gândite tocmai pentru scopul de a înălță părțile terțe din orice înțelegere între două părți. Un sistem de vot electronic care salvează voturile pe o rețea blockchain nu numai că ar reduce costurile, dar ar oferi și transparentă și auditabilitatea intrinsecă pentru a reduce orice dubiu legat de corectitudinea organizării scrutinelor de vot.

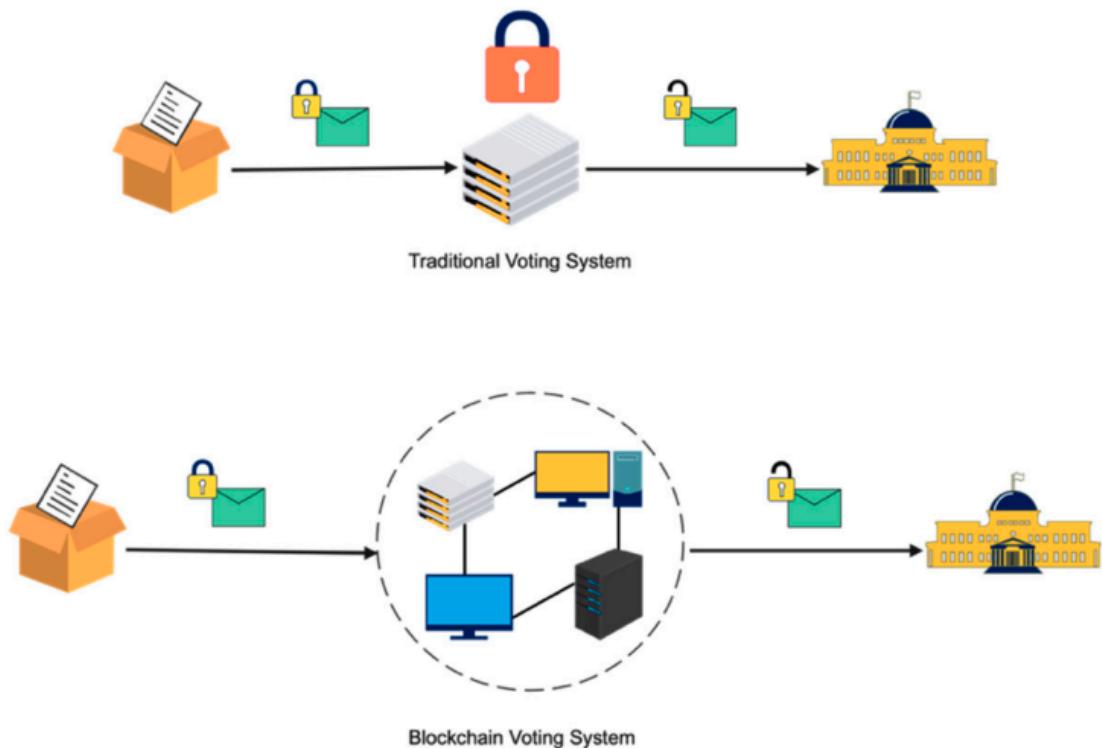


Figura 1: Sistem de vot electronic traditional vs Sistem de vot electronic peste blockchain [12]

Există câteva țări precum Japonia, Rusia sau Statele Unite ale Americii care au implementat astfel de sisteme de vot în scrutine oficiale și se arată rezultate promițătoare. Există și alte câteva țări, precum India și Coreea de Sud, care sunt la stadiul de dezvoltare și testare și

care își doresc implementarea oficială cât mai curând datorită avantajelor pe care le aduc contractele inteligente. Derularea a mai multor cercetări în acest domeniu și dorința a cât mai multor instituții să integreze un astfel de sistem în procesele acestora vor duce la crearea de sisteme de vot peste blockchain mai performante, mai sigure și mai ușor de folosit.

Avantajele unui sistem de vot electronic a cărui flux de vot să fie moderat de un contract intelligent sunt numeroase. În primul rand, sunt reduse costurile de personal și transport fizic, rămânând doar costurile de dezvoltare a sistemului, de menenanță și cel de rulare. Mai mult decât atât, înlăturarea unei autorități care să intermedieze fiecare vot duce la un nivel de echitabilitate mai ridicat, asemănător cu cel din vremea grecilor antici. În schimb, există câteva probleme ce aduc dezavantaje unui astfel de sistem. Cea mai importantă problemă este cea a autenticității voturilor. Ca un vot să fie corect, trebuie să existe o confirmare a faptului că aparține unui utilizator eligibil, fară să îi fie divulgată identitatea. De asemenea, nu trebuie permise voturi multiple sau voturi care nu sunt în nume propriu. O altă problemă caracteristică digitalizării este lipsa de încredere a oamenilor într-un astfel de sistem, soluția fiind o informare fermă și o tranziție lină. În această lucrare, vom vorbi despre cum PoliVote atacă aceste probleme.

PoliVote își propune să participe la evoluția sistemelor de vot electronice, folosindu-se de toate pârghiiile oferite de perechea formată dintre platforma Ethereum și *smart-contracts*. Mai mult decât atât, își propune că aplicația web să fie ușor de folosit, să nu creeze lipsă de încredere și să informeze utilizatorul în vederea tuturor acțiunilor ce sunt luate în fluxul votării.

2.2 Analiza cerințelor

În această secțiune, voi prezenta atât cerințele minime de funcționare pentru care PoliVote poate fi utilizată, cât și cerințele non-funcționale ce descriu capabilitățile oferite de către sistemul de vot.

Următoarele condiții sunt necesare pentru ca utilizatorul să poată folosi aplicația web a sistemului de vot:

- Un dispozitiv cu acces la internet.
- Dispozitivul să aibă instalat un browser de internet.
- Crearea unui cont de utilizator unde îi vor fi necesare informații de pe cartea de identitate.
- Instalarea extensiei MetaMask (vom detalia rolul acestei extensii în capitolele ce urmează). Această extensie este suportată de browserele Google Chrome, Microsoft Edge, Brave, Opera și Mozilla Firefox.
- Conexiunea la rețeaua blockchain a sistemului de vot prin intermediul extensiei (în cazul acestui proiect, rețeaua blockchain este locală).

De asemenea, sistemul de vot trebuie să îndeplinească următoarele cerințe non-funcționale:

- **Disponibilitate ridicată**: întreaga aplicație de vot electronic trebuie să fie funcțională la parametrii optimi pe tot parcursul unui scrutin de vot.
- **Latență mică**: aplicația trebuie să răspundă foarte rapid la interacțiunile cu utilizatorul.
- **Integritatea voturilor**: aplicația trebuie să asigure că alegerea utilizatorului nu este alterată până la criptarea votului și salvarea acestuia de către contractul intelligent.
- **Interzicerea voturilor invalide**: sistemul de vot trebuie să asigure faptul că nu există nicio posibilitate că un același utilizator să voteze mai mult de o singură dată sau un utilizator să voteze în numele altuia.
- **Confidențialitate**: sistemul trebuie să asigure că voturile sunt criptate cu credențialele corespunzătoare, fiecare vot să fie anonimizat, iar utilizatorul să aibă o confirmare că votul acestuia a fost salvat cu succes.
- **Non-repudiere**: sistemul trebuie să asigure că fiecare vot aparține unui utilizator valid și să dețină o formă de dovedire anonimizată.
- **Ascunderea rezultatelor parțiale**: sistemul trebuie să țină ascunse valorile voturilor până la finalul scrutinului pentru a nu permite calcularea rezultatelor intermediare.
- **Acuratețe în numărarea voturilor**: sistemul trebuie să numere toate voturile valide înregistrate cu o marjă de eroare nulă.
- **Auditabilitate**: sistemul trebuie să facă publice toate voturile pentru verificarea rezultatelor de către orice altă entitate.

2.3 Scenarii de utilizare

Scopul de bază a aplicației web de vot electronic este acela de a permite unui utilizator să își exprime votul, dar, mai mult decât atât, poate folosi următoarele funcționalități:

- Oferirea de instrucțiuni clare privind utilizarea aplicației, precum și modul de funcționare al sistemului de vot.
- Vizualizarea detaliilor biografice despre fiecare candidat în parte pentru a facilita luarea unei decizii a utilizatorului privind votul.
- Vizualizarea rezultatelor finale sub o formă grafică ușor de înțeles, valorile numerice aflându-se pe graficele specifice fiecărui candidat.
- Crearea unui cont de utilizator nou, precum și conectarea cu un cont existent.
- Vizualizarea în timp real a statusului scrutinului de vot. Mai precis, dacă acesta este în desfășurare sau nu, precum și datele de început și de sfârșit.

3 STUDIU DE PIATĂ / METODE EXISTENTE

În acest capitol voi prezenta stadiul curent al domeniului tehnologiei blockchain, modul cum aceasta a evoluat de-a lungul anilor, ce cazuri de utilizare sunt în prezent și cum se dorește că blockchain-ul să reprezinte o componentă esențială în sistemele de vot electronice. Mai mult decât atât, voi face o comparație între sistemul prezentat în această lucrare și alte soluții existente.

3.1 State of the Art

Blockchain-ul Ethereum a fost inventat în urma nevoii de a suporta instrucțiuni mai complexe în cadrul tranzacțiilor rețelei, de a dezvolta aplicații descentralizate și de a permite execuția unor contracte inteligente, lucru imposibil de efectuat în rețelele foarte cunoscute cum ar fi Bitcoin. Mai mult decât atât, s-a dorit ca soluțiile la aceste nevoi să beneficieze de caracteristici precum transparentă, imuabilitate și un nivel de securitate ridicat.

Contractul intelligent [3], sau *smart-contract*, reprezintă conceptul principal în jurul căruia gravitează toate elementele platformei Ethereum. Un contract intelligent este un cod ce se executa la întâlnirea unor anumite condiții și la inițierea unei tranzacții, urmând o arhitectură de tipul *event-driven* [9]. Acestea au scopul de a permite realizarea a diferitor activități online fără nevoie unui intermediator. Însă, în realitate, dezvoltarea unor astfel de contracte inteligente este una anevoieasă, fiind foarte dificile împiedicarea problemelor de securitate, cum ar fi atacuri malicioase sau surgeri de informații (DAO [16]), asigurarea eficienței (modalități de creștere a puterii de procesare) sau asigurarea *privacy-ului* [3]. Pentru a combate această plajă de probleme, se efectuează constant studii și cercetări pentru dezvoltarea unor noi paradigme de scriere a contractelor, precum și *framework-uri* sau alte unelte de depanare cu ajutorul cărora se pot dezvolta contracte inteligente mai rapid și mai sigur.

Contractele inteligente reprezintă o soluție pentru o multitudine de domenii, dar cele care beneficiază cel mai mult de avantajele unei astfel de tehnologii sunt scrutinile de vot, tranzacțiile financiare, documente oficiale digitale ale căror originalitate trebuie recunoscută pretutindeni, precum și domeniul sănătății. În aceasta lucrare, focusul este centrat pe utilizarea contractelor inteligente în scrutinile de vot și care este stadiul dezvoltării acestora în privință securității și *privacy-ului*.

O problema majoră cu sistemele de vot electronice clasice constă în prezența vulnerabilităților, cum ar fi *Single Point of Failure (SPoF)* și falsificarea datelor. SPoF reprezintă o vulnerabilitate a unei componente din arhitectura sistemului care, în caz de defect, determină oprirea întregului sistem. Această situație este complet inacceptabilă în cadrul unui sistem de vot electronic, care trebuie să poată gestiona cu succes toate cererile în fiecare secundă și să fie disponibil aproape 100% din timp. În cazul unei arhitecturi clasice pe trei niveluri, este dificil să se gestioneze o astfel de problemă. Cu toate acestea, un contract intelligent poate să contracareze această problemă datorită avantajelor intrinseci oferite de Ethereum, care asigură o disponibilitate de 100% prin existența mai multor noduri care susțin rețeaua. În cazul unei defecțiuni a unei părți din noduri, rețeaua nu se oprește complet, ci, în cel mai rău caz, reduce rata de procesare. În ceea ce privește falsificarea datelor, o arhitectură clasica nu poate oferi niciodată o garanție absolută că datele înregistrate nu pot fi alterate de către nicio entitate. Aceasta se datorează faptului că, deși informațiile sunt păstrate în siguranță, există cel puțin o entitate cu acces complet. În schimb, modificarea datelor înregistrate într-o rețea blockchain devine o sarcină imposibilă, deoarece ar implica modificarea înregistrărilor fiecărui nod din rețea, ceea ce reprezintă o complexitate computațională enormă.

Deoarece tranzacțiile în rețeaua Ethereum sunt publice și transparente, un sistem de vot care comunică cu această rețea trebuie să asigure o schemă bine pusă la punct pentru criptarea datelor, astfel încât să nu existe surgeri de informații legate de votanți sau de rezultatele parțiale. Aceste scheme de criptare trebuie să fie, în același timp, sigure și scalabile, astfel încât rata de procesare să nu scadă semnificativ datorită calculelor algoritmilor. De aceea, există o multitudine de studii de cercetare în desfășurare pentru a dezvolta o soluție de criptare fiabila la costuri computationale reduse, precum dezvoltarea de algoritmi de criptare homomorfica, ce permit folosirea datelor fără decriptarea acestora, sau dezvoltarea de rețele mixte (Mixnets [18]) care ascund expeditorul fiecărei tranzacții, înlăturând legătura dintre acesta și datele transmise.

Un alt aspect care se are în vedere în construirea unui sistem de vot electronic peste o rețea Ethereum este caracterul de nonrepudiere al voturilor. Mai precis, este necesar că votanții să demonstreze validitatea voturilor printr-o modalitate ce nu este intensiv computațională și, mai mult decât atât, să nu divulge nicio informație sensibilă. Si în direcția aceasta există nenumerate studii de cercetare ce încearcă să dezvolte modalități de semnare a datelor unei tranzacții de către expeditor. Până în momentul actual, cea mai populară paradigmă de verificarea autenticității datelor este *Zero-Knowledge Proof*. ZKP [11] este o modalitate prin care o entitate numită "doveditor" dovedește că informațiile transmise de către acesta sunt adevărate fără să divulge nicio informație către o altă entitate numită "verificator". Cercetătorii în domeniul concurează în dezvoltarea unor implementări ale acestei paradigmă, iar rezultatele sunt promițătoare. Un exemplu de algoritm ZKP de succes este *zk-SNARKs* (*Zero-Knowledge Succinct Non-Interactive Argument of Knowledge*) ce a fost folosit cu succes în proiecte precum *Zcash* [11].

Deși, pentru dezvoltarea unui sistem de vot electronic, este posibilă utilizarea unei rețele Ethereum publice, este recomandată folosirea unei rețele private din considerente de securitate și *privacy*. Un prim avantaj al dezvoltării și utilizării unei rețele private este flexibilitatea în restrictionarea accesului numai la conturi de utilizator autorizate și aplicarea anumitor criterii de eligibilitate pentru a putea realiza tranzacții. De asemenea, rețelele blockchain private pot utiliza mecanisme de consens [15] croite pe nevoie sistemului de vot, printre care *Proof of Authority (PoA)* sau *Practical Byzantine Fault Tolerance (PBFT)*. Mai mult decât atât, o rețea privată poate folosi tehnici diferite de procesare a tranzacțiilor pentru a îmbunătăți performanțele, precum *sharding* [21] sau *side-chains* [17].

PoliVote implementează un sistem de vot electronic experimental ce testează potențialul platformei Ethereum pentru a susține un scrutin de vot sigur și transparent. Acest sistem utilizează o rețea blockchain privată și tehnici de criptare RSA pentru a permite alegătorilor să-și exprime voturile într-un mod complet anonim și fără divulgarea acestora până la finalul procesului electoral.

3.2 Tendințe viitoare

În prezent, deși utilizarea rețelelor blockchain în cadrul sistemelor de vot electronice se arată a fiind o soluție promițătoare, există în continuare o multitudine de probleme ce încă sunt puse sub lupa cercetătorilor. Se lucrează într-un ritm alert la dezvoltarea de noi paradigmă sau de îmbunătățire a celor existente. Majoritatea lucrărilor de cercetare viitoare se vor concentra pe îmbunătățirea performanțelor rețelelor în cazul sistemelor de vot cu număr mare de utilizatori, modalități de scalabilitate și eficientizarea consumului de energie a blockchain-ului. De asemenea, se va lucra la modalități noi de a asigura *privacy*-ul participanților și de a tine în siguranță datele prin excluderea totală a unei entități terțe, lucru deocamdată impracticabil. Un alt aspect ce va fi acoperit în viitor este lucrul la partea diplomatică a unui sistem de vot electronic peste blockchain. Mai precis, se va lucra la sporirea încrederii oamenilor în aceasta tehnologie revoluționară și educarea lor în privință utilizării unui astfel de sistem de vot. Mai mult decât atât, se va lucra și în privința taberei opuse, adică cea a majorității liderilor politici care se opun unei astfel de inițiative din pricina posibilității pierderii puterii curente.

3.3 Abordări existente / Studiu de piață

În această secțiune, voi prezenta soluțiile existente în literatura specifică acestui domeniu, având în vedere modalitatea de utilizare a rețelei blockchain, modalitățile de criptare a datelor sensibile și cum se realizează autentificarea și validitatea votanților.

Follow My Vote

Follow My Vote este un sistem de vot electronic dezvoltat pentru a oferi transparentă, securitate și integritate a voturilor. Acesta se folosește de contracte inteligente aflate pe rețea blockchain privată, derivată din Ethereum. Protocolul de mecanism de consens este cel *Proof-of-Work* și este menit să ofere integritate datelor, viteza de procesare ridicată și protecție împotriva accesării neautorizate și modificării datelor.

Înscrierea utilizatorilor este anonimă, datorită componentei *Registrar* care verifică întări identitatea reală a votantului, alături de cheia lui publică, apoi generează un număr de identificare unic semnat de către autoritate. Ultimul pas este executat de către votant care își creează o nouă identitate cu ajutorul numărului de identificare, astfel anonimizând identitatea reală și păstrând, în același timp, autenticitatea persoanei.

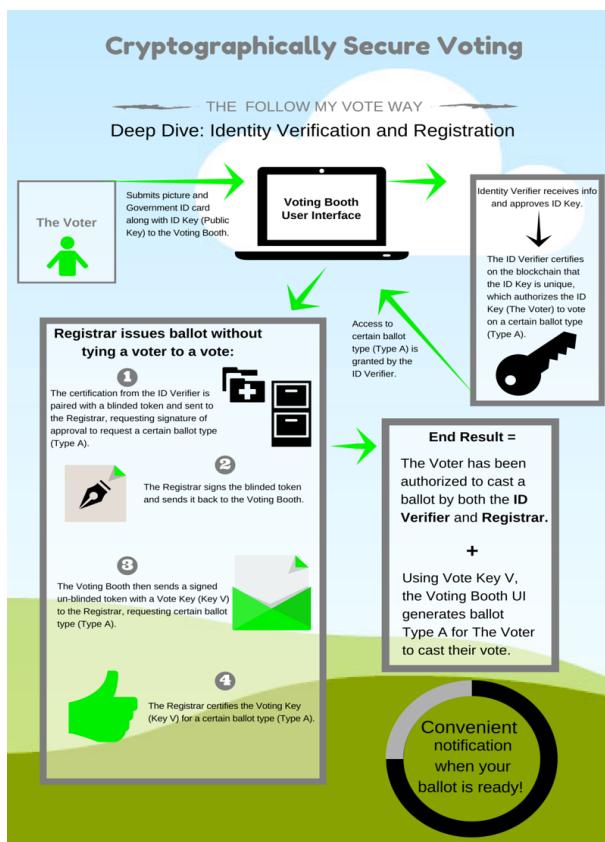


Figura 2: Follow My Vote - validarea utilizatorilor¹

Pentru validarea tranzacțiilor, mai precis a voturilor, pe rețea blockchain a aplicației *Follow My Vote*, se folosește un algoritm criptografic de tip "curbă eliptică" [8], comparativ cu PoliVote ce folosește un set de chei RSA ale votantului pentru validarea propriului vot. Astfel, sistemul de vot electronic *Follow My Vote* oferă siguranță și protecție a datelor utilizatorilor, precum și stocarea rezistență la modificări a voturilor, toate acestea la o viteză și eficiență ridicate.

¹© <https://followmyvote.com/cryptographically-secure-voting-2/>

Polys

Polys [13] este un sistem de vot electronic dezvoltat pentru a ajuta guvernele și companiile private să organizeze într-un mod sigur și eficient scrutinile de vot. Acesta combina atât modul electronic de vot, cât și pe cel tradițional, prin care îți poți exprima fizic votul, iar acesta din urmă putând fi salvat pe blockchain și invers, voturile din rețeaua blockchain putând fi printate și numărate fizic. Echipa ce a lucrat la dezvoltarea acestui sistem pune la dispoziție, pe lângă aplicația web și mobilă, terminale de vot dedicate cu ajutorul cărora participanții pot vota în interiorul unor sedii dedicate.

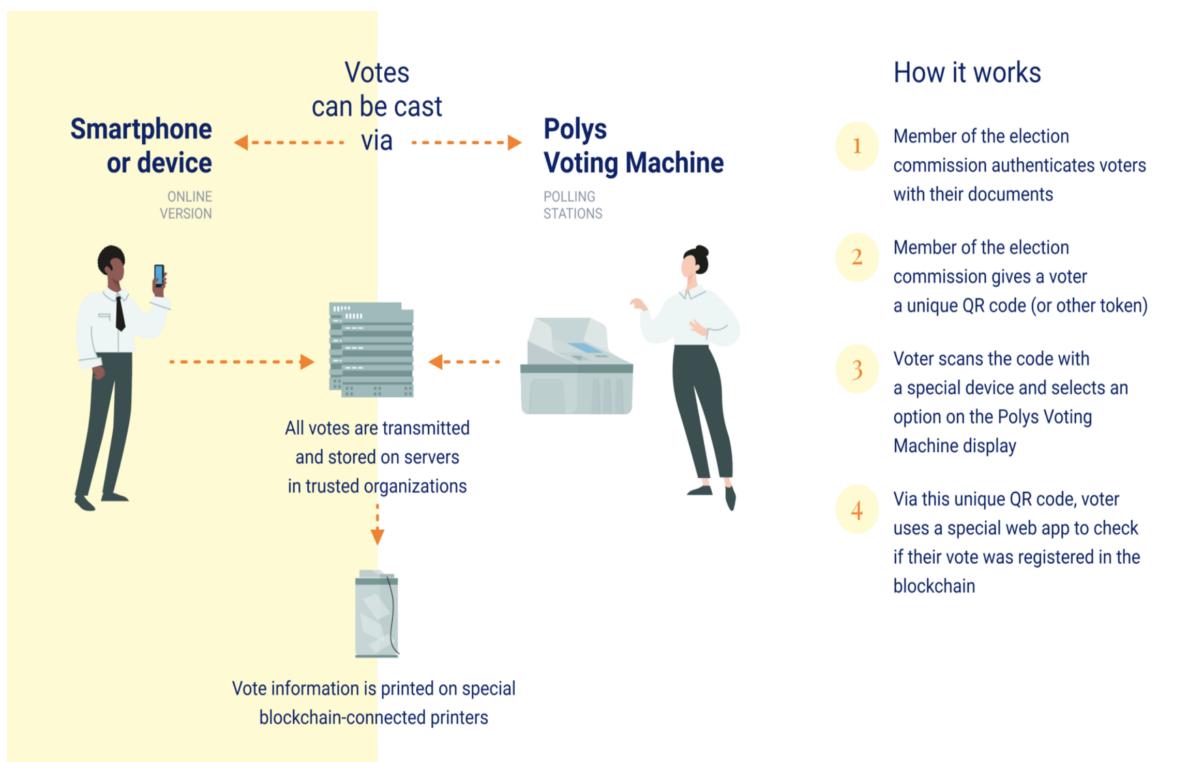


Figura 3: Polys - modalități de exprimare a votului ²

Polys utilizează o rețea blockchain privată bazată pe platforma Ethereum, mecanismul de consens utilizat fiind cel de *Proof-of-Work*. În ceea ce privește criptarea datelor și verificarea utilizatorilor, Polys folosește algoritmi de criptare de tip *Shamir's secret sharing* [10]. Acest tip de algoritm presupune împărțirea unei chei de decriptare în mai multe fragmente ce sunt împărțite fiecărui membru dintr-un grup. Pentru a folosi cheia de decriptare, este necesară o reasamblare a fragmentelor. Avantajul principal al acestui algoritm este rezistența la furturile de informații, deoarece atacatorul ar avea nevoie de toate fragmentele pentru a recompozi cheia, lucru greu de realizat computațional.

²https://polys.vote/images/tild6333-3736-4136-b565-323639386635_eng.png

Agora

Agora [1] este un sistem de vot electronic dezvoltat peste rețea blockchain Bitcoin, având o arhitectură formată din mai multe niveluri ce comunică între ele pentru a oferi un scrutin de vot sigur, transparent și verificabil.

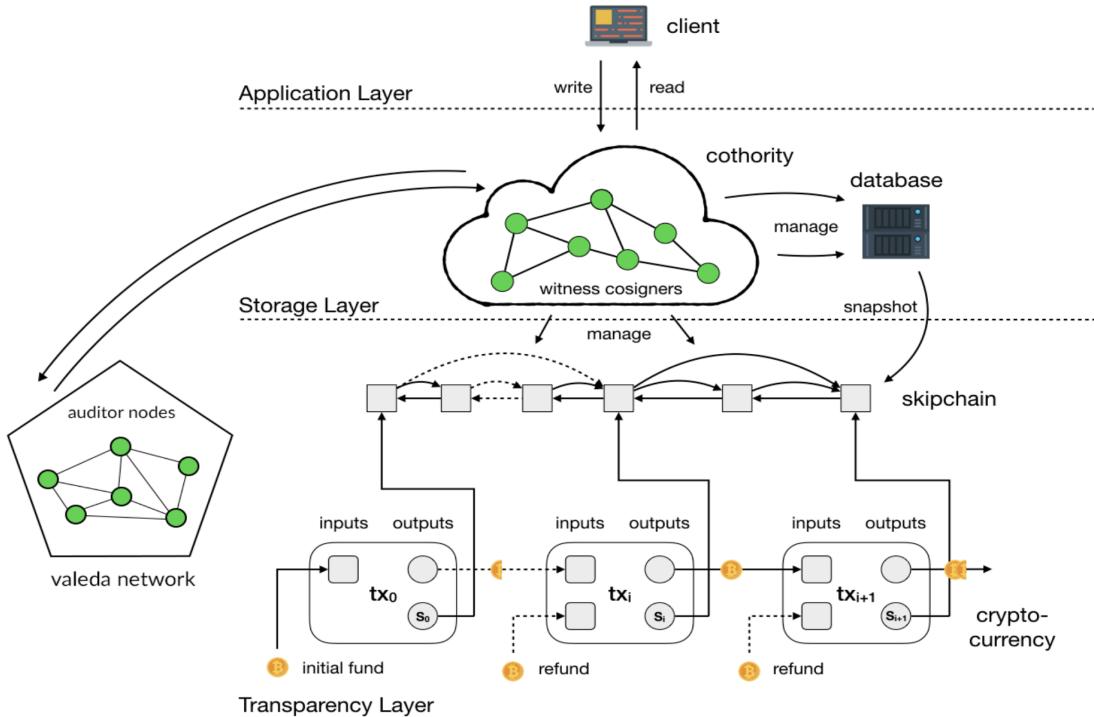


Figura 4: Agora - arhitectura generală [1]

Primul nivel este *Bulletin Board-ul*, o rețea blockchain privată și personalizată ce implementează o arhitectură de tip *Skipchain* [4]. Aceasta reprezintă coloana vertebrală a sistemului, componenta centrală de comunicare și stocare a datelor. Nodurile rețelei sunt compuse atât din calculatoare aparținând unor organizații neutre din punct de vedere politic, cât și din voluntari ce primesc un procent din suma de bani plătită de către organizatorii scrutinului de vot sub forma unor criptomonede³ ce pot fi vândute. Al doilea nivel este reprezentat de a doua rețea blockchain, *Cotena*, construită peste Bitcoin, ce este folosită pentru a ține evidența tranzacțiilor și dovezilor de validitate și autenticitate. Cel de-al treilea nivel este reprezentat de rețeaua *Valeda* compusă dintr-un grup de noduri ce are responsabilitatea de a valida și decripta voturile în timpul calculării rezultatelor. Ultimul nivel este cel al aplicației care interacționează cu toate celelalte componente și care permite unui utilizator fie să voteze, fie să verifice rezultatele scrutinului de vot, fie să se înscrive ca și nod în *Bulletin Board*.

³<https://www.agora.vote/vote-token>

3.4 Tehnologii utilizate

Sistemul de vot electronic PoliVote este implementat sub forma unei arhitecturi pe patru nivele. Interfața cu utilizatorul și componența de criptare și semnare a voturilor a fost dezvoltată cu ajutorul framework-ului *ReactJS* și limbajului JavaScript, din următoarele motive:

- **Arhitectură bazată pe componente:** ReactJS permite dezvoltatorilor să construiască componente reutilizabile UI, ceea ce îmbunătățește organizarea codului. Această abordare modulară facilitează întreținerea aplicațiilor la scară largă.
- **Ecosistem bogat în biblioteci și unelte:** ReactJS oferă o gamă largă de biblioteci și unelte precum *React Router* sau *Redux* ce îmbunătățesc dezvoltarea și utilizarea aplicațiilor.
- **Sintaxa JSX:** JSX este o extensie a limbajului JavaScript ce simplifică procesul de creare a componentelor.
- **Suport constant și comunitate activă.**

Pentru server, s-a folosit framework-ul *Spring Boot* și limbajul Java din următoarele considerente:

- **Ecosistem bogat în biblioteci și unelte;**
- **Dezvoltare modulară și scalabilitățile:** Spring Boot permite dezvoltarea modulară prin utilizarea *design pattern-urilor IoC* (Inversion of Control) și **dependency injection**.
- **Cuplarea slabă a componentelor:** Spring Boot permite concentrarea dezvoltatorului pe logica de business, înălțând nevoia întreținerii infrastructurii.

Pentru stocarea datelor anonimizate ale utilizatorilor și a datelor sesiunilor de vot, s-a folosit o bază de date relațională *PostgreSQL*, datorită scalabilității la creșterea volumului de date și accesarea rapidă a rândurilor din tabele prin folosirea indecșilor.

Pentru stocarea voturilor și dirijarea atât a tranzacțiilor, cât și a momentelor scrutinului electoral, s-a folosit o rețea privată Ethereum prin intermediul emulatorului *Ganache*, pe care s-a dezvoltat un contract inteligent construit cu ajutorul limbajului *Solidity*. Alegerea a fost făcută datorită următoarelor motive:

- Ethereum este cea mai populară soluție pentru dezvoltarea de aplicații descentralizate, Solidity fiind limbajul de bază pentru construirea contractelor inteligente. De asemenea, ecosistemul de biblioteci este unul bogat, iar comunitatea este foarte activă și oferă sprijin constant.
- Standardele Ethereum de dezvoltare a contractelor inteligente au devenit standardele întregii industrii, demonstrând încredere și fiabilitate.
- Solidity este un limbaj *strongly typed*, ce permite identificarea erorilor și îmbunătățirea robustetii.

4 SOLUȚIA PROPUȘĂ

În acest capitol, voi expune o panoramă detaliată asupra sistemului de vot electronic, având că obiectiv îmbunătățirea celui mai important pilon al democrației, și anume votul. Scrutinele clasice de vot sunt costisitoare, sunt dificil de organizat, iar asigurarea transparentei și corectitudinii nu este întotdeauna perfectă, lăsând loc de eroare umană la numărătoare, fraudă sau discriminare.

Soluția propusă în această lucrare descrie o opțiune mult mai sigură, mai transparentă și mai puțin costisitoare pentru desfășurarea unui scrutin de vot la orice scară, bazându-se pe beneficiile pe care le oferă o rețea blockchain, acestea fiind integritatea, transparenta și disponibilitatea datelor salvate pe rețea. Această opțiune este reprezentată de o aplicație web prin care utilizatorii eligibili își pot alege candidatul preferat în câțiva pași simpli. Votul lor este complet confidențial și înregistrat pe o rețea blockchain, iar rezultatele devin disponibile publicului doar după încheierea procesului electoral, asigurându-se astfel că nu există alegeri părtinitoare bazate pe rezultate parțiale.

Acet sistem este format din trei componente:

- **componenta client**, prin care utilizatorul se înregistrează, își poate alege candidatul preferat și care se ocupa de criptarea și semnarea voturilor;
- **componenta server**, prin care entitatea organizatorică generează și modereză scrutinul de vot, verifică eligibilitatea utilizatorilor ce doresc să se înscrive și se ocupă de numărarea voturilor;
- **componenta blockchain**, prin care voturile sunt validate și salvate, asigurându-se astfel că nu pot fi modificate ulterior.

Utilizatorul este întâmpinat de pagina de înregistrare, unde își solicită datele de conectare și informațiile de pe cardul de identitate. După ce identitatea este confirmată, își generează o pereche de chei asimetrice, care vor fi utilizate pentru a semna viitorul sau vot. Această pereche de chei este salvată într-o bază de date securizată a autorității competente. În momentul în care autoritatea de organizare deschide procesul de votare, utilizatorul are posibilitatea să-și exprime votul o singură dată, fără posibilitatea de a-l modifica sau retrage ulterior. Votul este criptat cu cheia publică a autorității, semnat de către utilizator și stocat în rețeaua blockchain. După încheierea scrutinului, utilizatorul poate vizualiza rezultatele finale.

Din perspectiva confidențialității, informațiile de pe cardul de identitate sunt protejate prin criptare cu o funcție hash (SHA256) după ce au fost verificate în prealabil. Aceste informații sunt stocate pentru a preveni duplicarea utilizatorilor sau înregistrarea a celor care nu corespund criteriilor (e.g. vîrstă, naționalitate). În plus, perechea de chei asimetrice (RSA cu un

key size de 2048 de biți) generată la înregistrare este păstrată într-o bază de date securizată a autorității și este utilizată pentru a semna votul utilizatorului, astfel împiedicându-se orice tip de fraudă. Valoarea fiecărui vot este criptată cu cheia publică (RSA cu un key size de 2048 de biți) a scrutinului electoral. Pentru un nivel de securitate mai ridicat, cheia de decriptare este stocată într-un fișier binar și este salvată în baza de date numai după încheierea procesului electoral, moment din care valoarea voturilor poate fi cunoscută public.

După ce utilizatorul își alege candidatul favorit, votul este criptat cu cheia publică a autorității, semnat cu perechea de chei a utilizatorului și salvat direct în rețeaua blockchain, astfel autoritatea nu are nicio cunoștință legată de ce utilizatori au votat, cu cine au votat și care sunt rezultatele.

Scopul principal al *smart-contract*-ului este acela de a stoca voturile manifestate fară posibilitatea modificării, alterării sau eliminării acestora. În același timp, efectuează validări pentru momentele în care votul poate avea loc, înregistrarea utilizatorilor și, de asemenea, validează cine are dreptul de a vota și cine poate modifica starea scrutinului.

Sistemul de vot electronic din soluția propusă urmează o arhitectură de tip client-server, conform figurii de mai jos:

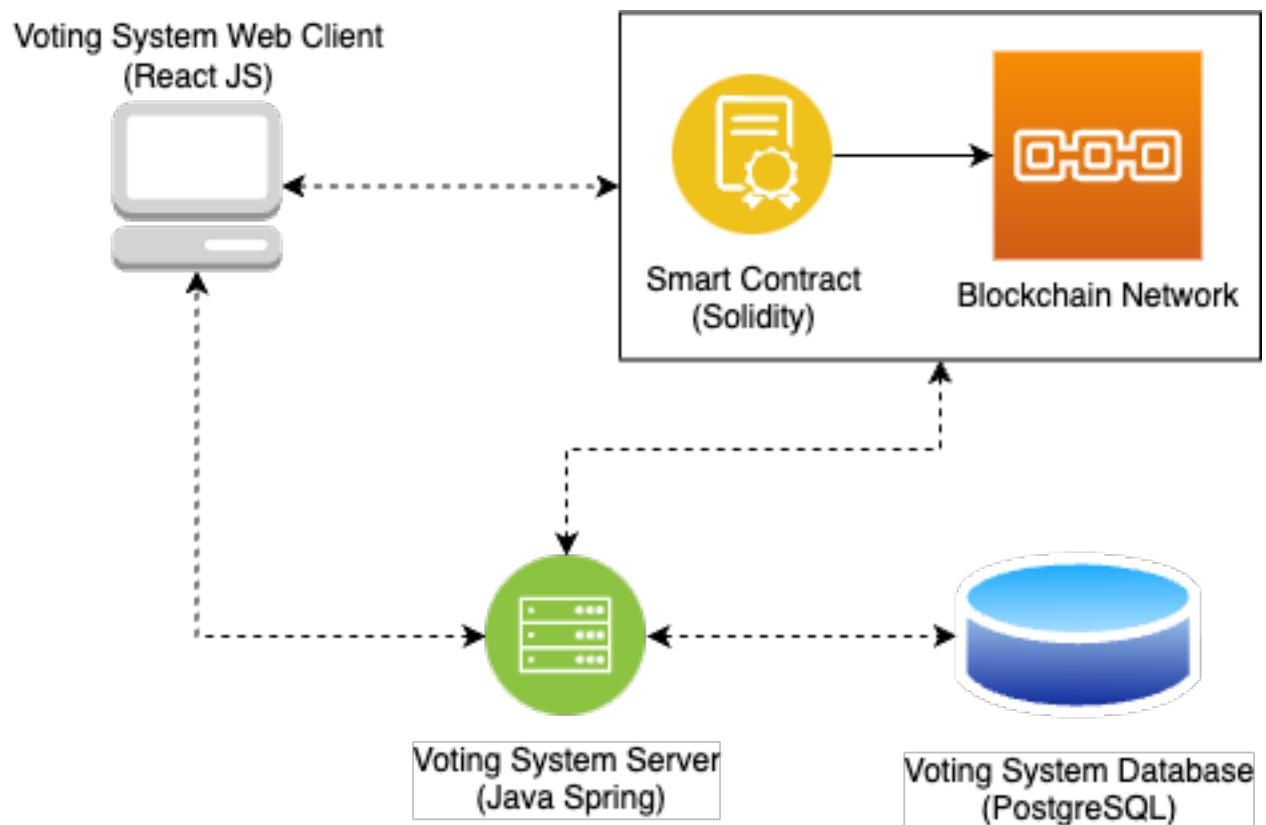


Figura 5: Arhitectura sistemului de vot

5 DETALII DE IMPLEMENTARE

În acest capitol, voi prezenta modul cum funcționează sistemul de vot electronic, detalii despre implementarea acestuia, cum procesează datele și cum interacționează utilizatorul cu aplicația web.

5.1 Aplicația Web - Componenta Front-end

În aceasta secțiune, voi explica caracteristicile componentei front-end, care reprezintă interfața cu care utilizatorul interacționează, și tehnologiile utilizate pentru implementarea acesteia din două perspective: cea a utilizatorului care dorește să voteze și cea a administratorului de sistem, reprezentant al autorității.

5.1.1 Perspectiva utilizatorului normal

Main Page

Utilizatorul este întâmpinat de pagina principală a aplicației web, descrisă de Figura 6. În partea stangă se află bara de navigare cu opțiuni ce duc la următoarele pagini:

- **What are we voting for?**, în care se oferă detalii despre scopul scrutinului curent de vot (vezi Figura 7);
- **Who are the candidates?**, în care se prezintă fiecare candidat în parte (vezi Figura 8);
- **How to vote?**, în care se explică detaliat pașii pe care trebuie să îi urmeze utilizatorul pentru a vota cu succes candidatul ales și cum se vizualizează rezultatele finale ;
- **Cast a vote**, în care utilizatorul poate vota candidatul ales numai după ce a început scrutinul de vot (vezi Figura 9);
- **Results**, în care utilizatorul poate vizualiza grafic rezultatele finale numai după ce scrutinul de vot s-a încheiat (vezi Figura 11).

De asemenea, în partea stangă, sub bara de navigare se află trei câmpuri care oferă informații despre starea curentă a scrutinului de vot: **Când începe? (When does it start?)**, **Când se termină? (When does it end?)** și **Statusul scrutinului (Session status)**. Detaliile despre starea procesului electoral, dar și paginile de **Cast a vote** și **Results** sunt accesibile doar după autentificarea utilizatorului în sistem.

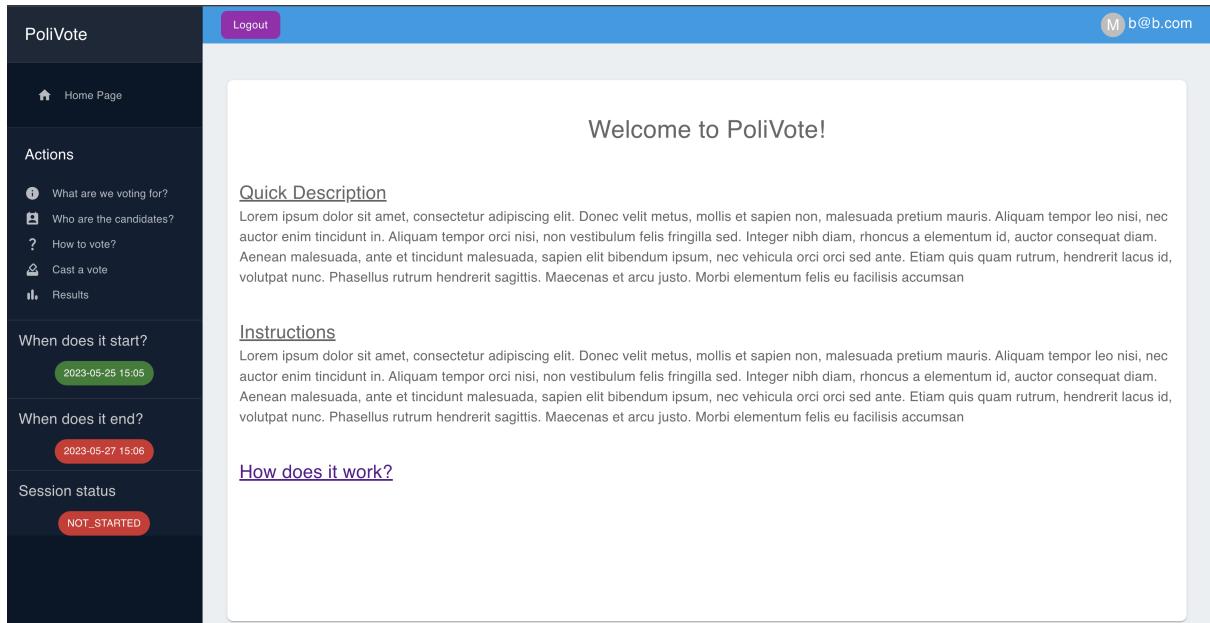


Figura 6: Pagina principală - perspectiva utilizator normal

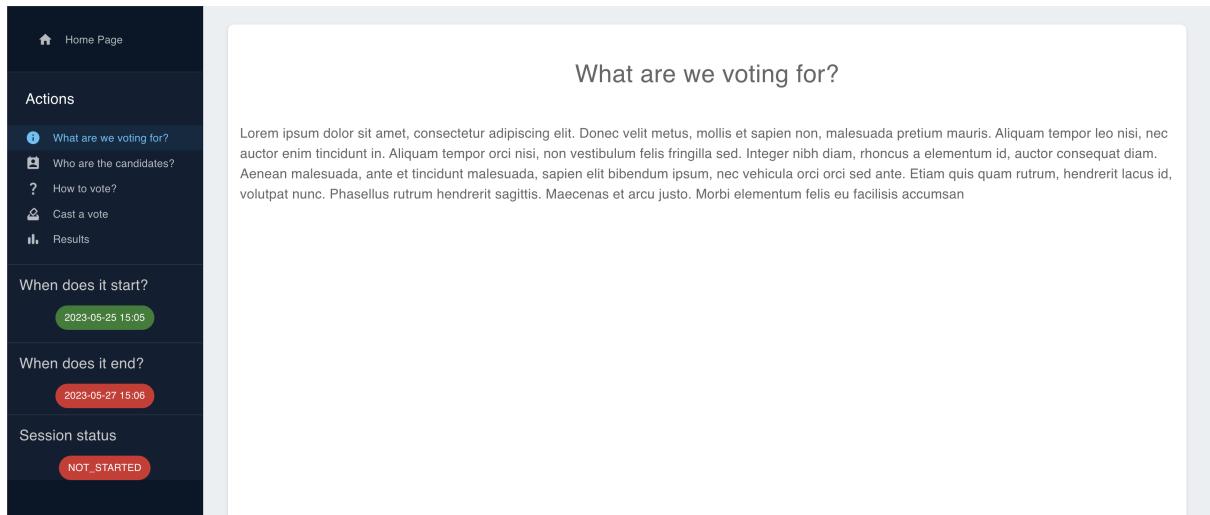


Figura 7: Pagina "Pentru ce votăm?" - perspectiva utilizator normal

Figura 8: Pagina "Cine sunt candidații?" - perspectiva utilizator normal

"Cast a vote" page

În cadrul acestei secțiuni a interfeței utilizatorului, este prezentată o listă a candidaților din procesul electoral curent. După ce a luat decizia, utilizatorul apasă pe butonul "Vote" situat lângă candidatul ales. De reținut este faptul că lista candidaților este disponibilă numai după ce autoritatea a dat startul desfășurării scrutinului de vot.

Figura 9: Pagina "Voteaza" - perspectiva utilizator normal

Fluxul procesului de vot

Primul pas esențial de realizat este conectarea browser-ului de internet al utilizatorului la rețeaua blockchain prin intermediul unui software de portofel digital, cum ar fi MetaMask¹. Această extensie permite comunicarea între browser și aplicațiile descentralizate care rulează pe rețeaua Ethereum. Pentru conectare, este necesar un cont de portofel digital de unde vor fi realizate tranzacțiile smart-contract-ului de vot. În cadrul acestui sistem de vot, care servește drept *proof-of-concept*, se utilizează o rețea blockchain locală furnizată de Ganache², o aplicație care permite emularea unei astfel de rețele pe baza platformei Ethereum. Astfel, contul de portofel digital necesar conectării la MetaMask poate fi oricare din cele oferite de Ganache și nu este obligatoriu ca aceste conturi să difere pentru fiecare vot. Fiecare vot este semnat cu credentialele de pe server ale utilizatorului. La o scară mai mare, autoritatea ar detine o rețea blockchain privată bazată pe Ethereum, cu un număr limitat de conturi de pe care să se poată realiza tranzacțiile voturilor. Valoarea în Ether din fiecare wallet ar fi utilizată exclusiv pentru realizarea tranzacțiilor, fiind imposibilă transferarea fondurilor către wallet-uri externe.

Odată ce conexiunea cu rețeaua blockchain este stabilită, votul utilizatorului este criptat cu cheia publică a autorității electorale. Perechea de chei asimetrice (RSA - 2048 biți) este generată odată cu scrutinul de vot, cheia publică fiind salvată în baza de date, iar cheia privată fiind salvată într-un fișier binar care poate fi stocat fie în mediul de stocare al server-ului, fie într-un mediu separat de tipul sau *USB Flash Drive* pentru un nivel de securitate ridicat.

Criptarea votului reprezintă prima treime din tranzacția ce va fi salvată pe rețeaua blockchain, a doua treime fiind reprezentată de semnătură votului. Semnătura unui vot reprezintă criptarea cu cheia privată a utilizatorului a sirului de caractere format din **cheia publică a utilizatorului în format hexazecimal și valoarea criptată a votului**, și ce este criptat cu o funcție hash (SHA256) în prealabil. Această semnătură are ca scop validarea faptului că votul aparține utilizatorului, evidențiind totodată caracterul de nonrepudiere. În plus, semnătura este verificată în procesul de numărare a voturilor pentru a exclude voturile invalide efectuate de către persoane neautorizate.

Ultimul pas implică împachetarea datelor relevante într-o tranzacție care va fi validată de *smart contract*. Aceste date sunt reprezentate de tuplul format din **cheia publică a utilizatorului, valoarea criptată a votului și semnătura votului**. Odată împachetate, se creează tranzacția prin apelarea funcției din smart-contract responsabilă cu validarea și salvarea votului. Folosirea funcțiilor smart-contract-ului se face cu ușurință în JavaScript prin importarea smart-contract-ului în format JSON și utilizarea bibliotecii *Web3j*. Pentru ca tranzacția să fie completă, această trebuie confirmată din extensia *MetaMask* după cum se poate vedea în Figura 10, care specifică costurile aferente tranzacției, pe ce rețea se efectuează și care este destinatarul (în cazul acesta, este adresa smart-contract-ului).

¹<https://support.metamask.io/hc/en-us/articles/360015489531-Getting-started-with-MetaMask>

²<https://trufflesuite.com/docs/ganache/>

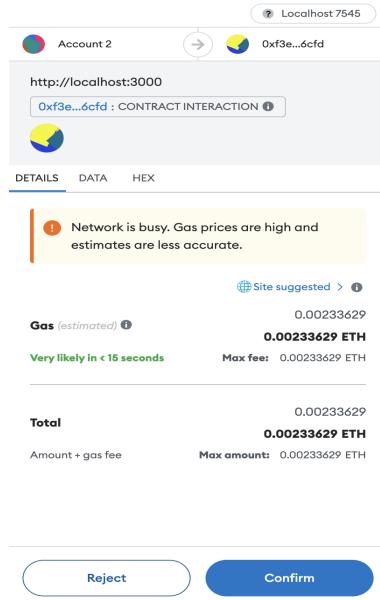


Figura 10: MetaMask - confirmare tranzacție

"Results" page

În această pagină a aplicației web, utilizatorul poate vizualiza grafic rezultatele finale ale scrutinului de vot, precum și numărul de voturi pentru fiecare candidat prin menținerea cursorului deasupra portiunii corespunzătoare candidatului din diagramă. Diagrama poate fi vizualizată numai după terminarea completă a procesului electoral. Diagrama este una de tip *Doughnut* cu portiunile fiecărui candidat de culori diferite, alături de o legendă sugestivă. Numărătoarea voturilor este realizată de componenta server și o vom discuta în secțiunile ce urmează.

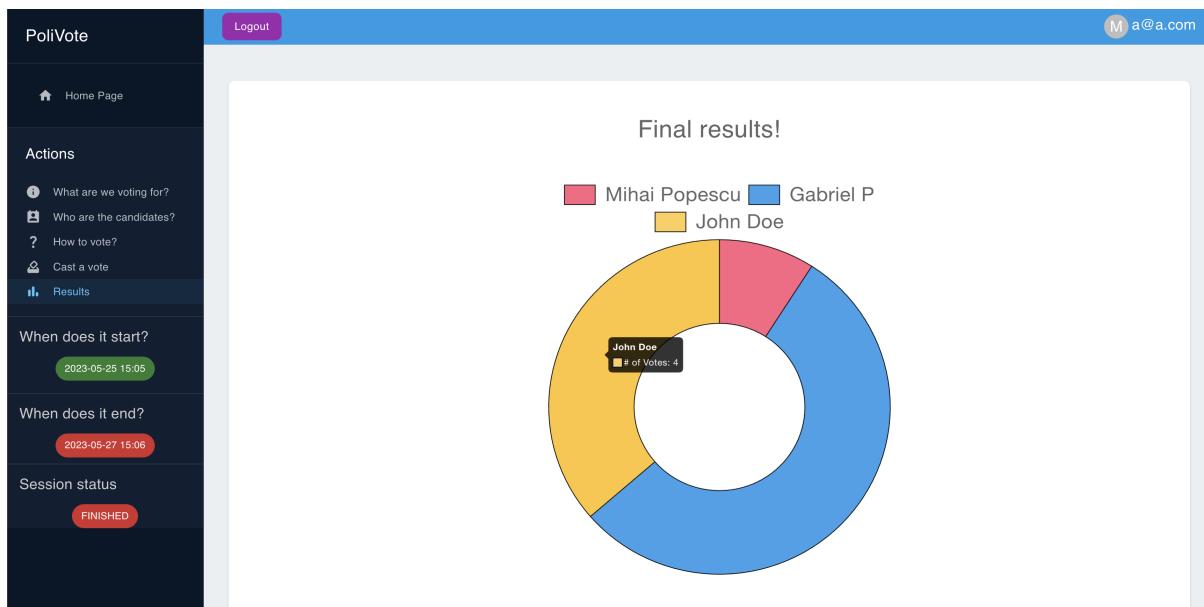


Figura 11: Rezultatele alegerilor - perspectiva utilizator normal

5.1.2 Perspectiva administratorului

Din punctul de vedere al designului, portalul administratorului se aseamănă cu interfața normală destinată utilizatorilor. Ce diferă, în schimb, sunt funcționalitățile puse la dispoziție administratorului, descrise grafic în Figura 12:

- **Add Candidate**, în care se deschide un formular pentru adăugarea unui candidat nou cu câmpurile de nume și descriere;
- **Edit candidates**, în care se afișează un tabel cu toți candidații, fiecare cu acțiunile de editare și stergere;
- **Create Voting Session**, în care se deschide un formular unde se poate crea o sesiune nouă de vot cu datele de început, respectiv de sfârșit. De reținut este faptul că poate exista o singură sesiune de vot la un moment dat, deci încercarea creării uneia noi vă fi întâmpinată cu un mesaj de eroare;
- **Edit voting session**, în care se pot modifica datele de început, respectiv de sfârșit a sesiunii de vot curente;
- **Add all voters to smart contract**, în care se adaugă manual toți utilizatorii înregistrați în lista de votanți eligibili ai smart-contract-ului. Motivul acestei acțiuni vă fi detaliat în secțiunile ce urmează (poate numele direct).
- **Start/End Voting Session**, în care se poate începe sau opri procesul electoral.

The screenshot displays the administrator's interface with the following sections:

- Register Candidate**: A form with fields for "Official Name*" and "Description", and a "Register candidate" button.
- Create Voting Session**: A form with fields for "Starting At*" and "Ending At*", and a "Create Vote Session" button.
- Add Voters to Contract**: A button labeled "Add Voters to Contract".
- Start or End the vote session**: Buttons labeled "Start Vote Session" (green) and "End Vote Session" (red).
- Candidate List**: A table with columns "ID", "Official Name", "Description", and "Actions". It lists three candidates:
 - Mihai Popescu (ID 9): Description - "Mihai Popescu este un om cum nu mai este nicuin alt om la varsta sa." Actions: Edit, Delete.
 - Gabriel P (ID 7): Description - "Gabriel P is a visionary leader who combines strategic thinking with a hands-on approach. With a keen eye for detail and a passion for innovation, he consistently drives organizational growth and success. Gabriel's strong analytical skills, coupled with his ability to inspire and motivate teams, make him a catalyst for positive change." Actions: Edit, Delete.
 - John Doe (ID 8): Description - "John Doe is a highly skilled professional with a proven track record in delivering exceptional results. With his extensive experience and expertise in various domains, he consistently exceeds expectations. John is a dedicated team player, known for his exceptional problem-solving skills and ability to adapt to changing environments." Actions: Edit, Delete.

Figura 12: Funcțiile administratorului

5.2 Componenta Server

În această secțiune, voi prezenta caracteristicile componentei server, care reprezintă mediatorul principal al întregului sistem de vot electronic. De asemenea, voi descrie în detaliu procesul de validare a eligibilității utilizatorilor, modul în care gestionează momentele cheie ale scrutinului de vot, ce date sunt salvate, locul unde acestea sunt stocate și modul în care interacționează cu *smart-contract-ul*.

5.2.1 Înscrierea și validarea utilizatorilor

Pentru înscrierea unui utilizator, este nevoie de următoarele date: adresa de e-mail, numărul de telefon, codul numeric personal sau CNP, seria cărții de identitate, data de expirare a cărții de identitate, parola și confirmarea parolei.

Procesul de înregistrare începe prin verificarea datelor de pe cartea de identitate pentru validare. Serverul verifică dacă CNP-ul este valid, dacă utilizatorul are vârstă necesară pentru vot, data de expirare a cărții de identitate și dacă seria este una validă. Vârstă utilizatorului este extrasă din CNP a cărui validitate este verificată prin calcularea cifrei de control³. În secvența de mai jos este descris algoritmul de calculare a cifrei de control. Acesta constă în înmulțirea fiecărei cifre din CNP cu cifra de pe același index al numărului 279146358279. Aceste produse sunt însumate și apoi împărțite la 11, restul împărțirii reprezentând cifra de control. Dacă aceasta este egală cu ultima cifră a CNP-ului, atunci este valid.

Algorithm 1 Algoritmul de validare al CNP-ului

```
1: function VALIDATECNP(CNP)
2:   if length(CNP) ≠ 13 then
3:     return false
4:   end if
5:   control ← [2, 7, 9, 1, 4, 6, 3, 5, 8, 2, 7, 9]
6:   sum ← 0
7:   for i ← 0 to 11 do
8:     sum ← sum + CNP[i] × control[i]
9:   end for
10:  rest ← sum%11
11:  if rest = 10 then
12:    rest ← 1
13:  end if
14:  return rest == cnp[12]
15: end function
```

³<https://cnpgenerator.ro/verificare-cnp>

După verificarea informațiilor cărții de identitate, se verifică dacă email-ul, numărul de telefon sau cartea de identitate se află în baza de date a server-ului. Dacă totul este în regulă, se salvează noul utilizator, iar informațiile legate de cartea de identitate sunt *hash*-uite cu ajutorul algoritmului SHA256 pentru un nivel de securitate ridicat.

Deoarece acest sistem de vot este un *proof-of-concept*, validarea informațiilor cărții de identitate se face la un nivel minimal. La o scară mai mare, autoritatea ar verifica cărțile de identitate într-o bază de date guvernamentală.

După ce informațiile de identificare ale utilizatorului sunt salvate, se generează o pereche de chei asimetrice folosind algoritmul RSA cu o dimensiune a cheii de 2048 de biți. Această pereche de chei este apoi stocată în baza de date a autorității. Astfel, utilizatorul este înregistrat complet în sistemul de vot.

5.2.2 Începerea și sfârșitul procesului electoral

Server-ul acționează, în cazul acesta, ca un dirijor care transmite atât *smart-contract*-ului, cât și interfeței cu utilizatorul în ce moment al scrutinului ne aflăm. Înaintea începerii alegerilor, *smart-contract*-ul nu permite adăugarea voturilor în rețeaua blockchain sau a listei de votanți eligibili (vom detalia acest aspect la secțiunea destinată componentei *smart-contract*), iar interfața web nu permite vizualizarea opțiunilor de vot. Odată pornită sesiunea de vot de către administrator, reprezentant al autorității, *smart-contract*-ul permite adăugarea voturilor și interzicerea înscrierii de utilizatori noi, iar interfața web afisează utilizatorului lista de candidați din pagina *Cast a vote*.

La închiderea sesiunii de vot de către administrator, server-ul transmite *smart-contract*-ului să nu mai permită nicio tranzacție de salvare pe rețeaua blockchain, iar opțiunile de vot să nu mai fie disponibile în pagina web. Totodată, server-ul pune la dispoziție tuturor cheia de decriptare a voturilor, care a fost prealabil ținută în siguranță pe un mediu de stocare extern sau chiar în mediul de stocare al server-ului, prin salvarea acesteia în baza de date a autorității.

5.2.3 Comunicarea cu smart-contract-ul

Datorită faptului că serverul este dezvoltat în limbajul de programare Java și folosește framework-ul *Spring Boot*, este nevoie de crearea unui wrapper al smart-contract-ului pentru a permite *Java Virtual Machine* să execute codul acestuia. Acest wrapper este creat cu ajutorul *Web3j CLI*⁴ și compilatorului de *Solidity*⁵.

⁴https://docs.web3j.io/4.10.0/command_line_tools/

⁵<https://docs.soliditylang.org/en/v0.8.20/>

Pașii pentru crearea wrapper-ului sunt următorii:

- compilarea codului scris în *Solidity* al *smart-contract-ului*. Comanda prin care s-a compilat codul sursă este următorul:

```
solc Election.sol --bin --abi --optimize -o ./
```

Rezultatul este generarea unui fișier binar și a unui *Application Binary Interface* sau ABI, care descrie întreaga interfață a *smart-contract-ului* în format JSON. Aceste două fișiere vor fi folosite mai departe în generarea codului Java, wrapper al funcțiilor *smart-contract-ului*.

- generarea clasei Java reprezentând wrapper peste interfața *smart-contract-ului* folosind *Web3j CLI*:

```
web3j generate solidity
-a Election.json
-b Election.bin
-o ./src/main/java
-p com.gpoalelungi.licenta.contract
```

Rezultatul este generarea clasei **Election** ce reprezintă întreaga interfață a *smart-contract-ului*. Opțiunea **-a** primește ca parametru fișierul ABI generat anterior, opțiunea **-b** primește fișierul binar generat anterior, opțiunea **-o** specifică destinația clasei Java generate, iar opțiunea **-p** specifică pachetul Java în care aceasta să fie inclusă.

Interfața generată conține doar definiția funcțiilor, evenimentelor și membrilor *smart-contract-ului*, neștiind să comunice cu rețeaua blockchain. Pentru a permite server-ului să comunice cu rețeaua, acesta trebuie să instanțieze contractul-wrapper cu următorii parametri:

- **Manager de tranzacții**: clasă ce permite folosirea unui nod al rețelei blockchain pentru realizarea tranzacțiilor. Acesteia îi sunt necesare o adresă a unui portofel digital de pe care să se realizeze tranzacțiile și adresa IP:PORT unde se află rețeaua. Astfel, serverul poate efectua apeluri JSON-RPC (*JavaScript Object Notation Remote Procedure Call*)⁶ [14] către rețea pentru a utiliza funcțiile smart contractului;
- **Adresa contractului**: adresă în format hexazecimal pe 20 de octeți ce identifică *smart-contract-ul*;
- **Contract Gas Provider**: clasă ce specifică costului unei tranzacții și limita acestui cost, numite *gas price* și *gas limit*

⁶<https://ethereum.org/en/developers/docs/apis/json-rpc/>

Din acest punct, pentru a realiza un apel de funcție al *smart-contract*-ului, sunt necesari următorii pași:

- **Definirea tranzacției.** Mai precis, **specificarea adresei expeditorului** (care este aceeași cu adresa portofelului digital setat managerului de tranzacții), **costul tranzacției** (*gas price* și *gas limit*), **adresa contractului sistemului de vot** și **referința encodată a funcției** ce va fi apelată. Mai jos este prezentat un exemplu de creare de tranzacție pentru apelul funcției *startVote()* al *smart-contract*-ului

```
Transaction.createFunctionCallTransaction(  
    ADMIN_ADDRESS, null, GAS_PRICE, GAS_LIMIT, CONTRACT_ADDRESS,  
    election.startVote().encodeFunctionCall());
```

- **Trimiterea tranzacției.** După ce tranzacția este definită, trebuie să fie trimisă către rețeaua blockchain pentru a fi procesată de nodurile acesteia. Biblioteca *Web3j* utilizează protocolul *JSON-RPC* pentru a realiza acest lucru. Se efectuează un apel sincron către rețea și se așteaptă un răspuns care conține rezultatul întors de funcție în cazul în care tranzacția a fost realizată cu succes, sau motivele pentru care tranzacția a fost respinsă (nodurile din rețea nu permit tranzacții care conțin erori și revin la starea anterioară a blocului în care s-a încercat realizarea tranzacției).

```
EthCall response = web3j  
    .ethCall(transaction, DefaultBlockParameterName.LATEST)  
    .send();
```

- **Primirea "chitanței" tranzacției.** După ce tranzacția a fost confirmată și înregistrată într-un bloc al blockchain-ului, se generează o "chitanță" a tranzacției care conține detalii despre blocul în care a fost înregistrată tranzacția, costurile asociate, numărul de tokeni transferați, expeditorul, destinatarul, identificatorul unic al tranzacției, logurile execuției funcției, adresa *smart-contract*-ului și, în special, valoarea returnată de funcție.

```
String transactionHash = web3j  
    .ethSendTransaction(transaction)  
    .send()  
    .getTransactionHash();
```

```
TransactionReceipt receipt = web3j  
    .ethGetTransactionReceipt(transactionHash)  
    .send()  
    .getTransactionReceipt();
```

5.2.4 Decriptarea, validarea și numărarea voturilor

Odată încheiat scrutinul de vot, se interzice exprimarea voturilor noi și orice alta tranzacție asupra rețelei blockchain, se eliberează cheia privată de decriptare a voturilor ce devine disponibilă tuturor și începe manual procesul de numărare.

Procesul începe prin obținerea întregii liste de voturi a contractului intelligent folosind API-urile expuse de clasa wrapper. După ce primește lista, serverul decriptează fiecare vot și îl atribuie candidatului corespunzător, aşa cum este descris în secvență următoare de cod Java. Metoda `getAllVotes()` apelează, de fapt, metoda `getAllVotes()` a contractului intelligent și returnează o listă de voturi de tipul `Election.Vote`, asociată 1:1 cu structura `Vote` a contractului.

```
List<Candidate> candidates = getAllCandidates();
List<Election.Vote> votes = votingSessionService.getAllVotes();

for (Election.Vote vote : votes) {
    String choice = votingSessionService.decryptVote(...);
    for (Candidate candidate : candidates) {
        if (candidate.getOfficialName().equals(choice)) {
            incrementVoteCount(candidate.getId());
        }
    }
}
```

Pentru decriptarea și validarea votului, sunt necesare cheia publică a utilizatorului, valoarea votului criptat și semnătura acestuia. La pasul de validare se decriptează semnătura cu cheia publică a utilizatorului și se verifică dacă valoarea decriptată este aceeași cu sirul de caractere format prin concatenarea cheii publice a utilizatorului și valoarea criptată a votului. În caz contrar, înseamnă că votul este malformat, nu aparține utilizatorului și, deci, nu este luat în considerare la numărătoare. În caz afirmativ, metoda `decrypt()` își continua execuția și decriptează votul cu cheia privată a sesiunii de vot. Valoarea votului este chiar numele candidatului din baza de date. Mai jos sunt prezentate secvențele de cod pentru pasul de validare și unde este acesta folosit:

```
public String decryptVote(String publicKey, String voteToDecrypt,
String signature) throws Exception {
    ...
    if(!isValidSignature(signature, publicKey, voteToDecrypt)) {
        throw new InvalidVoteException("Invalid vote found");
    }
    ...
}
```

```

private Boolean isVoteValid(String voteSignature, String voterPublicKey, String encryptedVote) throws Exception{
    byte[] voterPublicKeyBytes = Base64.getDecoder().decode(voterPublicKey);
    KeyFactory publicKeyFactory = KeyFactory.getInstance( algorithm: "RSA");
    EncodedKeySpec publicKeySpec = new X509EncodedKeySpec(voterPublicKeyBytes);
    PublicKey publicKey = publicKeyFactory.generatePublic(publicKeySpec);

    Cipher cipher = Cipher.getInstance( transformation: "RSA");
    cipher.init(Cipher.DECRYPT_MODE, publicKey);

    byte[] decryptedSignature = cipher.doFinal(Base64.getDecoder().decode(voteSignature));

    String decryptedSignatureToString = bytesToHex(decryptedSignature);
    String encryptedVoteHash = getSha256Hash( plainText: voterPublicKey + encryptedVote);

    return encryptedVoteHash.equals(decryptedSignatureToString);
}

```

Figura 13: Metoda de validare a unui vot

5.2.5 Stocarea datelor

În ceea ce privește stocarea datelor, sistemul de vot folosește atât rețeaua blockchain pentru persistența voturilor și a listei de utilizatori eligibil (vom detalia acest aspect în secțiunea destinată componentei blockchain), cât și o baza de date relațională pentru persistența datelor utilizatorilor și ale sesiunii de vot.

În momentul înregistrării, fiecărui utilizator i se solicită să furnizeze adresă de email, numărul de telefon, parola, confirmarea parolei și informațiile de pe cartea de identitate, în scopul verificării dreptului utilizatorului de a vota. Înainte ca aceste date să fie stocate într-o tabelă a unei baze de date *PostgreSQL*, ele sunt prelucrate pentru a fi anonimizate. Mai exact, parola este criptată folosind o funcție de hash numită *BCrypt*[2], care este specializată în criptarea parolelor și este utilizată în industrie datorită folosirii unui salt și a unui algoritm de hash care consumă resursele procesorului. De asemenea, informațiile de pe cartea de identitate sunt criptate utilizând același algoritm de hash *BCrypt*, astfel încât să nu fie posibilă identificarea persoanelor. Valoarea criptată rezultată este ulterior utilizată pentru a verifica dacă un utilizator încearcă să se înregistreze cu aceleași informații.

După înregistrarea cu succes a utilizatorului, se generează o pereche de chei asimetrice utilizând algoritmul *RSA* cu un *key size* de 2048 biți. Această pereche de chei este stocată într-o altă tabelă a bazei de date relaționale și este disponibilă doar proprietarului prin intermediul unei cereri *HTTP*. Serverul verifică tokenul *JWT*⁷ din antetul cererii *HTTP*, extrage numele de utilizator (care este de fapt adresa de email) și caută în tabela corespunzătoare perechea asociată acestui utilizator.

⁷<https://jwt.io/introduction>

5.3 Componenta blockchain

În această secțiune, voi prezenta caracteristicile componentei blockchain, care reprezintă cea mai vitală piesă din întreg sistemul de vot electronic, având responsabilitatea de a salva fără modificări ulterioare voturile pe rețeaua blockchain, de a asigura transparenta, disponibilitatea și integritatea acestora. De asemenea, voi explica în detaliu modul în care *smart-contract*-ul implementează un proces electoral corect, echitabil și de încredere.

5.3.1 Blockchain, Ethereum, Solidity și Smart-Contracts

Înainte de a discuta despre implementarea logicii la nivelul smart-contract-ului, este necesar să menționam câteva aspecte de bază legate de blockchain, ethereum, limbajul de programare Solidity și Smart-Contracts pentru a facilita înțelegerea întregului ansamblu.

Blockchain

Prin definiție, blockchain [3] este un registru digital, descentralizat, reprezentat de o lanțuire imutabilă de blocuri ("block chain") care este în continuă creștere. Fiecare bloc conține un set finit de tranzacții, un identificator unic, un marcap temporal și un hash criptografic al conținutului său.

Principala caracteristică a tehnologiei blockchain constă în descentralizare. Blockchain-ul operează pe o rețea de noduri, fiecare nod fiind un calculator care detine o copie a registratorului. Acest lucru asigură transparentă și redundantă, deoarece fiecare participant are acces la aceleși informații. Nodurile lucrează conform unui set de reguli sau mecanisme de consens, astfel încât ele să ajungă la un acord cu privire la ordinea și validitatea tranzacțiilor. Astfel, această arhitectură elimină nevoie unei entități centrale care să țină evidența datelor, făcând sistemul transparent și sigur.

Tehnologia blockchain se bazează în mare măsură pe metode criptografice, atât pentru securizarea și anonimizarea datelor, cât și pentru a detecta orice tip de modificare adusă blocurilor. Fiecare tranzacție este semnată digital cu ajutorul cheilor criptografice, garantând astfel autenticitatea și integritatea datelor. De asemenea, blocurile sunt conectate între ele prin intermediul hash-urilor criptografice. Orice modificare adusă unui bloc implică recalcularea hash-ului aceluia bloc și a tuturor blocurilor următoare, ceea ce face imposibilă modificarea istoricului blockchain-ului.

Astfel, tehnologia blockchain este potrivită ca soluție la probleme ce implică confirmarea autenticității datelor și acțiunilor, transparenta, siguranța și disponibilitatea acestora. Un caz potrivit este problema abordată în aceasta lucrare: un sistem de vot electronic ce stochează voturile într-un mediu transparent, sigur și disponibil, anonimizat pentru a nu asocia votul cu votantul, iar voturile să poată fi verificate de orice entitate, fară că acestea să fie cenzurate.

Ethereum

Ethereum [3] este o platformă open-source ce utilizează tehnologia Blockchain și este folosită în mare măsură pentru a dezvola aplicații descentralizate, numite și *dApp-uri* (*decentralized Apps*), utilizând *smart-contract-uri*. Această platformă este guvernată de existența criptomonedei native, *Ether*, ce este folosită în principal pentru a plăti taxele tranzacțiilor (*gas price*), dar și că monedă ce poate fi transferată în schimbul unor servicii sau produse. Aceste criptomonede *Ether* sunt oferite anumitor noduri din rețea ce validează și securizează blocurile și tranzacțiile noi apărute printr-un mecanism de consens.

Smart-Contracts și Solidity

Smart-contracts sau *contracte-inteligente* sunt programe ce rulează pe *EVM*⁸, *Ethereum Virtual Machine*, ce au capacitatea de a se auto-executa în momentul în care anumite termene sunt îndeplinite, fapt ce le aseamănă cu contractele clasice. Ele au fost introduse pentru prima oară pe platforma Ethereum și au schimbat paradigma modului în care se pot utiliza anumite servicii fără nevoie unei entități terță pentru a valida și autoriza acțiunile. Pentru a explica în termeni simpli ce face un smart-contract, ne putem gândi la o analogie cu un automat de răcoritoare. Automatul oferă o gamă de băuturi în schimbul unei sume de bani. Acesta acceptă anumite tipuri de bancnote și poate elibera una dintre băuturile dorite numai dacă suma de bani acumulată depășește prețul articoului. Întreg automatul se ocupă de comercializarea bunurilor, gestionarea banilor și siguranța împotriva furturilor, nefiind necesară existența unei persoane care să supravegheze și să intermedieze tot procesul de vânzare-cumpărare. Asemănător funcționează și un smart-contract, care permite realizarea tranzacțiilor specifice contractului doar dacă sunt îndeplinite anumite condiții, fără nevoie supravegherii și validării acțiunilor de către o entitate terță.

Există mai multe limbi de programare folosite pentru a scrie codul unui contract intelligent, iar cel mai popular dintre ele este *Solidity*, limbajul nativ al platformei open-source *Ethereum*. Prin intermediul *Solidity*, sunt implementate funcții care pot fi apelate și executate de către participanții rețelei, sunt definite condițiile pentru executarea acestora, se stabilesc variabilele în care se stochează date specifice și se generează evenimente odată cu executarea anumitor acțiuni. Odată încheiată elaborarea codului sursă, contractul este încărcat în rețeaua blockchain, de unde participanții îl pot apela pentru a încheia tranzacții.

Smart-contract-urile, datorită faptului că funcționează pe o rețea blockchain, sunt descentralizate și imuabile, însemnând că, după ce sunt create și implementate, ele nu pot fi modificate. În plus, aceste contracte inteligente aduc transparentă și auditabilitate, deoarece toate tranzacțiile și acțiunile sunt înregistrate permanent în blockchain.

⁸<https://ethereum.org/en/developers/docs/evm/>

5.3.2 Implementarea smart-contract-ului electoral

Odată familiarizați cu tehnologia blockchain, platforma *Ethereum*, limbajul de programare Solidity și contractele inteligente, putem trece la următorul aspect al componentei blockchain al sistemului de vot electronic. Mai precis, la implementarea propriu-zisă a contractului procesului electoral.

Am discutat despre rolul serverului în sistem și despre modul în care acesta acționează ca un moderator și efectuează validări pentru a se asigura că utilizatorii sunt eligibili. De fapt, serverul adaugă doar un nivel suplimentar de verificări pentru a spori securitatea. Contractul intelligent efectuează ultimele verificări privind exprimarea votului înainte de a fi înregistrat în blockchain. Aceste lucruri le vom detalia în paragrafele ce urmează.

Moderarea momentelor procesului electoral

Deși această moderare se realizează la nivelul serverului, este esențial ca aceasta să existe și la nivelul contractului intelligent pentru a preveni eventualele atacuri malicioase sau efectuarea de tranzacții invalide care să ducă la înregistrarea unor voturi ilegale. Astfel, în *smart-contract* există doi membri de contract publici:

```
bool public isVoteStarted;  
bool public isVoteFinished;
```

Această pereche de variabile cu nume sugestive expun care este stadiul curent al procesului electoral: înainte de începere (amândouă cu valoarea *false*), început (*true*, respectiv *false*) și sfârșit (amândouă cu valoarea *true*). Perechea este folosită mai departe în cadrul a trei *modificatoare de funcții* (*function modifiers*⁹) ce sunt folosite pentru a impune în mod declarativ anumite condiții înaintea execuției funcțiilor pe care acestea s-au aplicat:

```
modifier onlyAfterVoteFinished() {  
    require(isVoteFinished == true);  
    _;  
}  
  
modifier onlyAfterVoteStartedAndNotFinished() {  
    require(isVoteFinished == false && isVoteStarted == true);  
    _;  
}  
  
modifier onlyBeforeVoteStarted() {  
    require(isVoteFinished == false && isVoteStarted == false);  
    _;  
}
```

⁹<https://docs.soliditylang.org/en/v0.8.19/contracts.html#function-modifiers>

Aceste modificatoare sunt specificate în antetul funcției și comandă EVM-ului ca înainte de rularea funcției să evalueze predicatul din funcția `require()`. Dacă este adevărat, continuă cu execuția funcției apelate (se cedează înapoi fluxul execuției prin `_`). În caz contrar, nu se execută funcția apelată și se aruncă o excepție prin apelarea funcției `revert()`, care resetează starea blocului la ultima variantă stabilă. Mai jos este redat un exemplu de utilizare a acestor modificatoare de funcții. Astfel, dacă starea scrutinului de vot nu este în desfășurare (adică `isVoteStarted = false` și `isVoteFinished = true`), se va apela direct funcția `revert()`, fără ca fluxul execuției să intre în funcția `addVote()`. În caz contrar, condițiile sunt îndeplinite și fluxul execuției poate fi redat funcției `addVote()`:

```
function addVote(...) public onlyAfterVoteStartedAndNotFinished
{ \\\code }
```

Momentele procesului electoral pot fi modificate prin intermediul a două funcții din *smart contract* ce pot fi apelate doar de către administratorul scrutinului de vot. Astfel, în implementarea *smart contract*-ului este specificat contul de portofel digital al administratorului din rețea blockchain în variabila de tip `address` numită `authorizedAddress`. Dat fiind faptul că cele două funcții `startVote()` și `endVote()` sunt funcții ce modifică starea blocului (adică execută operații de `write`), ele conțin un câmp ce specifică adresa expeditorului (`msg.sender`). Acest câmp este comparat cu `authorizedAddress` și în caz de inegalitate, nu se modifică starea scrutinului de vot. Astfel, numai autoritatea are controlul deplin asupra procesului electoral, împiedicând orice tip de atac malicios din exterior.

```
address private authorizedAddress;

modifier onlyAuthorized() {
    require(msg.sender == authorizedAddress, "Unauthorized caller");
    _;
}

function startVote() public onlyAuthorized {
    isVoteStarted = true;
    emit VoteStartedOrFinished("Vote has started");
}

function endVote() public onlyAuthorized {
    isVoteFinished = true;
    emit VoteStartedOrFinished("Vote has finished");
}
```

Înscrierea utilizatorilor pe smart-contract

Am vorbit la secțiunea dedicată componentei server despre modul în care se realizează înscrierea și validarea utilizatorilor. Odată finalizată lista de votanți (mai precis, odată ce se decide începerea procesului electoral), este transmisă contractului intelligent lista cu cheile publice ale tuturor votanților. Astfel, fiecare cheie publică din orice tranzacție de vot este verificată cu cheile din lista finală. Dacă aceasta există, atunci votul aparține unui utilizator valid și este înregistrat mai departe într-un bloc al rețelei. În caz contrar, se apelează funcția revert(), iar blocul revine la o stare anterioară stabilă.

Însă, o problemă ce trebuie atacată este posibilitatea ca un utilizator eligibil să poată vota mai mult de o singură dată. Solutia la această problemă este dată de existența unei structuri în smart-contract care reține pentru fiecare utilizator dacă este înregistrat (pentru a nu permite duplicarea votanților) și dacă și-a exprimat votul. Definirea structurii și a listei de votanți este exemplificată în secvența de cod următoare:

```
struct Voter {  
    bool isRegistered;  
    bool hasVoted;  
}  
  
mapping(bytes32 => Voter) private voters;
```

Mapping reprezintă o structură de date ce se aseamănă cu un dicționar al altor limbaje de programare, dar diferă prin faptul că valorile sunt stocate în memoria contractului (numită și *storage*, memoria contractului acționează ca o bază de date ce persistă anumite date chiar și după terminarea execuției programului), iar valorile nu sunt accesibile în mod direct din exterior, fiind necesară definirea unor funcții de tipul *getter* sau *setter*. În cazul de fată, *voters* reprezintă un dicționar în care fiecare cheie este reprezentată de hash-ul cheii publice a unui utilizator obținut printr-un algoritm de hash numit *Keccak256* [7], iar fiecare valoare este reprezentată de structura *Voter*. *Keccak256* este o funcție de hash folosită în Solidity și face parte din familia *SHA3* [5]. Aici, este folosită pentru a optimiza căutarea după cheie în dicționar prin transformarea cheii publice a utilizatorului din sir de caractere în *bytes32*, tip de date ce reprezintă un vector de 32 de octeți. De reținut este faptul că dicționarul este privat, adică nu poate fi accesat din exterior decât cu ajutorul unei funcții, evitând astfel o eventuală scurgere de informații ale utilizatorilor.

Mai jos este funcția care adaugă un utilizator în lista de votanți eligibili ai contractului intelligent. Aceasta primește că parametru un sir de caractere ce reprezintă cheia publică a unui utilizator. Sirul trece prin funcția hash specificată mai devreme, obținându-se un vector de 32 de octeți. Se verifică dacă cheia a mai fost înregistrată prin verificarea câmpului *isRegistered*. Dacă nu a mai fost înregistrat, se adaugă în lista de votanți și se consemnează acțiunea

prin emiterea unui *eveniment*¹⁰. Un aspect important de menționat este faptul că Solidity initializează automat mapping-ul cu valorile implicite atunci când se accesează pentru prima dată o cheie nouă. De aceea, pentru un utilizator neînregistrat, *currentVoter.isRegistered* nu este NULL și are valoarea implicită *false*.

```
function addVoter(string memory _publickey) public onlyAuthorized
onlyBeforeVoteStarted {
    bytes32 hashedPublicKey = keccak256(abi.encodePacked(_publickey));
    Voter memory currentVoter = getVoter(hashedPublicKey);
    require(currentVoter.isRegistered == false, "Voter already registered");

    voters[hashedPublicKey] = Voter(true, false);
    emit VoterAdded(hashedPublicKey, true, false);
}

function getVoter(bytes32 hashedPublicKey)
private view returns (Voter memory){
    return voters[hashedPublicKey];
}
```

Adăugarea voturilor

Am discutat în secțiunea dedicată componentei front-end despre rolul acesteia în crearea tranzacției care conține votul criptat al utilizatorului, precum și cheia publică și semnătura acestuia. Acum, voi prezenta modul în care componenta blockchain, mai precis contractul intelligent, validează votul și îl stochează în rețeaua blockchain.

Smart contract-ul implementează, asemănător listei de votanți, o listă de voturi exprimate, conform secvenței de cod de mai jos. Fiecare vot este reprezentat de structura de date *Vote* ce conține cheia publică a utilizatorului, votul criptat cu cheia publică a autorității și semnătura utilizatorului, toate sub forma de sir de caractere. Lista de voturi este reprezentată de un *mapping* unde fiecare cheie este un număr întreg ce reprezintă indexul votului, iar fiecare valoare este votul propriu-zis.

```
struct Vote {
    string publicKey;
    string encryptedVote;
    string signature;
}
```

¹⁰<https://docs.soliditylang.org/en/v0.8.19/contracts.html#events>

```

uint private voteCount = 0;
mapping(uint => Vote) private votes;

```

Odată cunoscută structura unui vot și modul în care este stocat, putem discuta despre funcția contractului inteligent care validează și salvează voturile utilizatorilor, prezentată în secvența de cod de mai jos. Prin urmare, funcția primește ca parametri cele trei valori importante ale unui vot: cheia publică a utilizatorului, votul criptat și semnătura. În primul rând, se verifică dacă utilizatorul și-a exprimat deja votul anterior. În caz contrar, se apelează funcția revert() și execuția funcției se încheie. Altfel, votul este salvat în listă, iar utilizatorul este marcat ca având un vot exprimat. Dacă analizăm funcția, observăm că votul nu este salvat în spațiul de stocare al întregii rețele, ci în *storage-ul* contractului inteligent. Confirmarea salvării votului este reprezentată prin emiterea unui eveniment care este atașat "chitanței" tranzacției. Această dovedă este necesară pentru a oferi utilizatorului asigurarea că votul său a fost salvat în siguranță și nu poate fi modificat. O observație importantă este faptul că voturile pot fi adăugate doar dacă scrutinul de vot este în desfășurare cu ajutorul modificadorului de funcție *onlyAfterVoteStartedAndNotFinished*.

```

function addVote(
    string memory _publicKey,
    string memory _encryptedVote,
    string memory _signature) public onlyAfterVoteStartedAndNotFinished
{
    bytes32 hashedPublicKey = keccak256(abi.encodePacked(_publicKey));
    Voter memory currentVoter = getVoter(hashedPublicKey);
    require(currentVoter.hasVoted == false && currentVoter.isRegistered == true,
    "Unauthorized voter");

    voteCount++;
    votes[voteCount] = Vote(_publicKey, _encryptedVote, _signature);
    voters[hashedPublicKey].hasVoted = true;
    emit VoteCasted(_publicKey, "Vote has been registered for the voter");
}

```

Publicarea și numărarea voturilor

Publicarea voturilor este o acțiune care necesită o atenție deosebită pentru a evita dezvăluirea rezultatelor partiale. Aceasta trebuie făcută doar la sfârșitul procesului electoral și din acest moment nimeni să nu mai poată vota. Din acest motiv au fost luate măsuri de precauție, printre care reducerea vizibilității listei de voturi prin specificatorul *private*. De asemenea, funcția responsabilă de publicarea întoarce lista de voturi doar după ce procesul electoral s-a

încheiat. Ne reamintim că momentele scrutinului de vot pot fi modificate numai de către autoritatea sistemului de vot. Astfel, este o misiune imposibilă pentru un atacator să obțină orice fel de rezultate intermediare. Mai jos se află secvența de cod a funcției ce întoarce lista de voturi la momentul încheierii procesului electoral.

```
function getAllVotes() public view onlyAfterVoteFinished
returns (Vote[] memory) {
    Vote[] memory returnedVotes = new Vote[](voteCount);
    for (uint i = 0; i < voteCount; i++) {
        returnedVotes[i] = votes[i+1];
    }
    return returnedVotes;
}
```

5.3.3 Rețeaua blockchain - Ethereum

Doar scrierea codului sursă a contractului intelligent nu este suficientă ca acesta să funcționeze. După implementare, contractul trebuie compilat într-un format ce este recunoscut de către EVM. Mai precis, codul sursă este transformat într-un *Application Binary Interface*, sau ABI, ce descrie toate intrările, ieșirile și funcțiile contractului. Compilarea este urmată de încărcarea ABI-ului în rețea. Această încărcare se face în trei pași:

- **crearea tranzacției de deployment:** practic, se creează o cerere de încărcare a ABI-ului împachetată într-o tranzacție care este trimisă nodurilor pentru procesare;
- **validarea de către nodurile rețelei:** tranzacția este trimisă la noduri care validează cererea de încărcare și includ contractul într-un bloc. Odată ce blocul este adăugat în rețea, contractul poate fi utilizat;
- **generarea noii adrese a contractului:** după ce încărcarea s-a terminat cu succes, rețeaua blockchain (în cazul nostru, Ethereum) atribuie contractului o adresă cu ajutorul căruia poți efectua tranzacții și apela funcțiile contractului.

Pentru acest sistem, a fost folosit *Ganache*, un program ce permite găzduirea unei rețele Ethereum pe mașina locală. Ganache creează o rețea blockchain locală ce oferă zece portofele digitale, fiecare cu un număr finit, dar suficient de *ETH* ce poate fi consumat în tranzacții. Aplicația expune un *RPC Server* [14] prin care se poate interacționa cu blockchain-ul și oferă detalii despre contractele încărcate și tranzacțiile ce au loc.

De asemenea, pentru dezvoltarea și încărcarea contractului intelligent, s-a folosit *Truffle*, un framework pentru dezvoltarea de *smart-contract-uri* pe Ethereum. Acesta conține o sută de unelte ce permit dezvoltarea, testarea și încărcarea contractelor.

The screenshot shows the Ganache interface with the following details:

- MNEMONIC:** funny prize rude holiday develop come lab aspect warrior custom silver puppy
- HD PATH:** m/44'/60'0'@account_index
- CURRENT BLOCK:** 961
- GAS PRICE:** 20000000000
- GAS LIMIT:** 6721975
- HARDFORK:** MERGE
- NETWORK ID:** 5777
- RPC SERVER:** HTTP://127.0.0.1:7545
- MINING STATUS:** AUTOMINING
- WORKSPACE ELECTION:**
- SWITCH:**
- SETTINGS:**

ADDRESS	BALANCE	TX COUNT	INDEX	KEY
0x17987d50dD4439Ed4Cf3975Ef6752D1C7a076cA3	98.41 ETH	961	0	🔑
0x82CE336804ea6d3E25a78c1DB776E435A5B298b9	100.00 ETH	0	1	🔑
0x463A4E16E259bCdA77B46B5DE5d9430289E3f05C	100.00 ETH	0	2	🔑
0x13691165812F4863B67bAc4623977c99F107da3e	100.00 ETH	0	3	🔑
0x04780e76353f2Ce4630B38b2DA70d10dC84Ac682	100.00 ETH	0	4	🔑
0x6d6b73b4FDA4615445254dFd497E309469a2239d	100.00 ETH	0	5	🔑
0x6491d4b1ce293fFe883404b6870fAfC409082CbF	100.00 ETH	0	6	🔑

Figura 14: Ganache - panou principal

Costurile încărcării și ale tranzacțiilor

Potrivit modului în care arhitectura *Ethereum* este concepută, tranzacțiile care implică citirea din blockchain sunt gratuite, deoarece nodurile nu necesită validări și, prin urmare, nu consumă resurse. În schimb, orice operatie de scriere, inclusiv încărcarea unui contract în rețea, implică o taxă denumită *gas price*. Această taxă este plătită de nodurile care efectuează validările și scrierile în blocuri. *Gas price*-ul este proporțional atât cu dimensiunea și tipul operațiilor de scriere, cât și cu cererea și oferta în rețea. Cu cât există mai multe tranzacții în așteptare, cu atât valoarea taxei crește.

Pentru contractul nostru, costurile totale vor fi suma costurilor pentru adăugarea votanților în lista de utilizatori eligibili, costurile pentru adăugarea fiecărui vot în lista totală de voturi și costul încărcării contractului în rețea. De reținut este faptul că în cazul exceptiilor apărute în cadrul tranzacțiilor de scriere ce au sfârșit cu *revert()*, "gazul" consumat până în acel moment nu poate fi rambursat. De asemenea, nefiind cererea mare de tranzacții pe rețea blockchain locală, nu se adaugă costurile generate de trafic.

6 EVALUAREA SISTEMULUI DE VOT

Din punct de vedere al fiabilității sistemului, s-a realizat testarea temeinică a componentei blockchain pentru a asigura rezistență absolută la orice tip de atac malitios sau utilizare neautorizată prin intermediul testelor unitare folosite cu ajutorul utilitarului *Truffle*. De asemenea, întregul sistem a fost testat manual cu ajutorul unor scenarii de testare:

- Validarea și înscrierea cu succes a utilizatorului doar înainte de începerea scrutinului;
- Imposibilitatea votării înainte de începerea scrutinului;
- Adăugarea cu succes a listei de votanți în lista contractului intelligent doar înainte de începerea scrutinului;
- Criptarea corectă și semnarea anonimă a votului;
- Transmiterea cu succes a votului către rețeaua blockchain;
- Aruncarea exceptiilor în cazurile de vot neautorizat sau duplicat de către contractul intelligent;
- Eliberarea cheii de decriptare a voturilor doar după încheierea scrutinului;
- Imposibilitatea înscrierii și votării după terminarea scrutinului;
- Calcularea rezultatelor numai după terminarea scrutinului;
- Calcularea corectă a rezultatelor la fiecare interogare;
- Afisarea rezultatelor finale doar după terminarea scrutinului de vot;

De asemenea, sistemul de vot electronic PoliVote a fost testat de către un număr de voluntari care au fost rugați să răspundă la un set de întrebări referitoare atât la funcționarea aplicației, cât și la cunoștințele generale legate de tehnologia blockchain.

Care este nivelul dumneavoastra de familiaritate cu tehnologia blockchain?

 Copiază

26 de răspunsuri

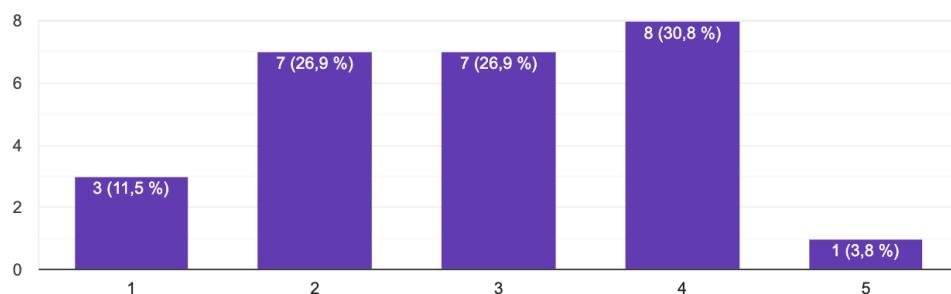


Figura 15: Nivelul de familiaritate cu tehnologia blockchain

Ati votat pana acum prin intermediul unei aplicatii online?

 Copiază

26 de răspunsuri

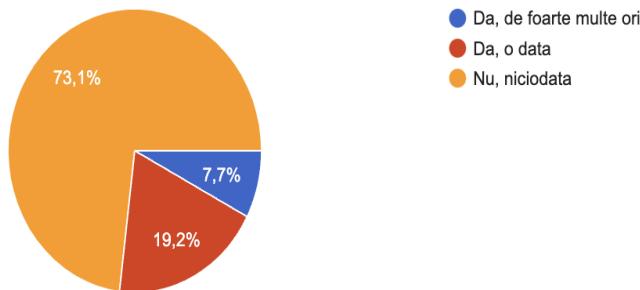


Figura 16: De câte ori au votat online participanți

Conform figurii 15, un total de 34.6% dintre participanți dețin cunoștinte medii sau avansate despre tehnologia blockchain. Restul de 65.4% susțin faptul că dețin cunoștinte minime, doar au auzit despre blockchain sau nu sunt deloc familiari cu acest domeniu. Din aceste date putem trage concluzia că un sistem de vot electronic inovator, bazat pe o rețea blockchain poate fi folosit cu încredere de către populație doar cu o campanie puternică de informare și educare în privința funcționării acestei tehnologii. De asemenea, conform figurii 16, majoritatea utilizatorilor participanți la chestionar au recunoscut că nu au nicio experiență cu aplicațiile existente de vot electronic.

Cat de intuitiva vi se pare aplicatia PoliVote?

 Copiază

26 de răspunsuri

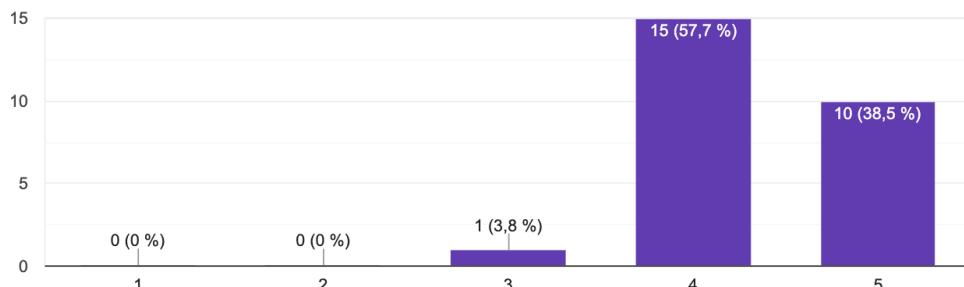


Figura 17: Intuitivitatea aplicației

Pentru ca sistemul de vot electronic să fie util, acesta trebuie nu numai să funcționeze corect, ci să fie și ușor de folosit. Având în vedere că publicul țintă al acestei aplicații este populația generală, fară cunoștinte avansate în domeniu, este absolut necesar ca aplicația să fie intuitivă. Conform figurii 17, 96.2% dintre participanți au considerat aplicația PoliVote că fiind foarte ușor de folosit.

Care este nivelul de incredere pe care il aveti in ceea ce priveste confidentialitatea datelor gestionate de PoliVote?

 Copiază

26 de răspunsuri

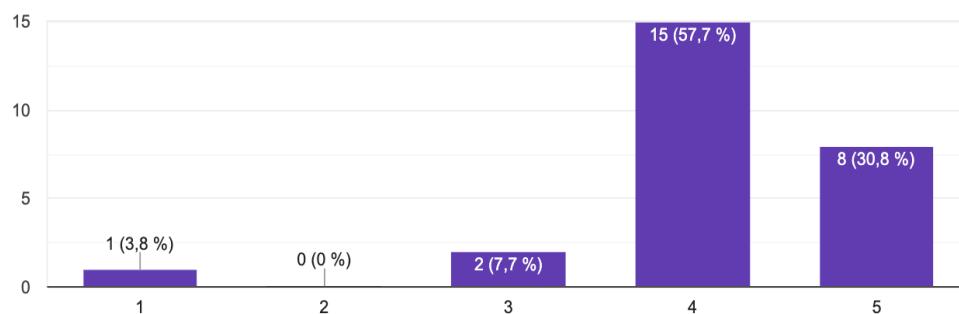


Figura 18: Încrederea în confidențialitatea datelor

Conform figurii 18, persoanele participante la chestionar au privit cu încredere aplicația de vot electronic referitor la stocarea datelor personale și la asigurarea anonimitatii votului. Cu toate acestea, încă există dubii ale utilizatorilor cu privire la acest aspect, datorită noutății și a schimbării de paradigmă a modului de exprimare a votului.

Considerati ca votul dumneavoastra a ramas nemodificat pana la sfarsitul procesului electoral?

 Copiază

26 de răspunsuri

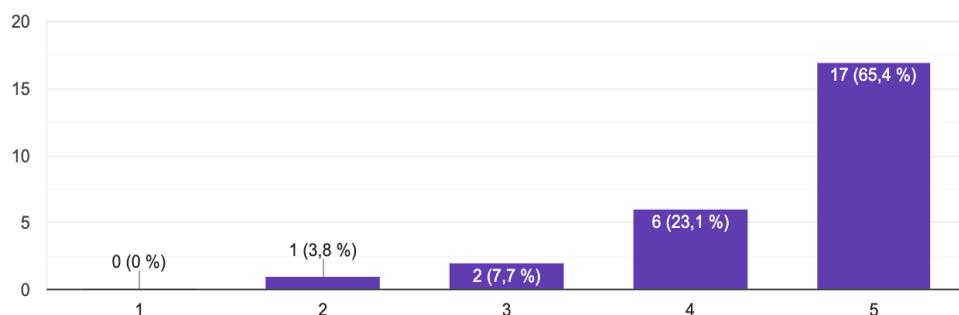


Figura 19: Încrederea în nemodificarea votului

Majoritatea participanților nu au exprimat îndoieri legate de corectitudinea voturilor acestora, așa cum se poate observa în figura 19. Acest lucru se datorează implementării simple a procesului de vot și a transparenței acesteia.

Care este nivelul de incredere pe care il aveti in ceea ce priveste calcularea rezultatelor finale ale voturilor de catre PoliVote?

 Copiază

26 de răspunsuri

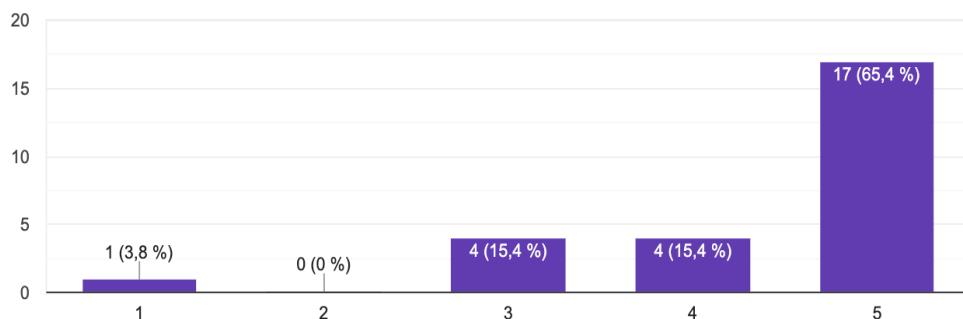


Figura 20: Încrederea în calcularea rezultatelor

De asemenea, majoritatea participantilor nu au exprimat îndoieri legate de calcularea corectă a rezultatelor, conform figurii 20. Încrederea se datorează punerii la dispoziție a întregii liste de voturi ce poate fi folosită pentru calcularea rezultatelor finale de către orice entitate doritoare.

Cat de confortabil(a) sunteți cu utilizarea unui sistem de vot electronic ce folosește tehnologia blockchain?

 Copiază

26 de răspunsuri

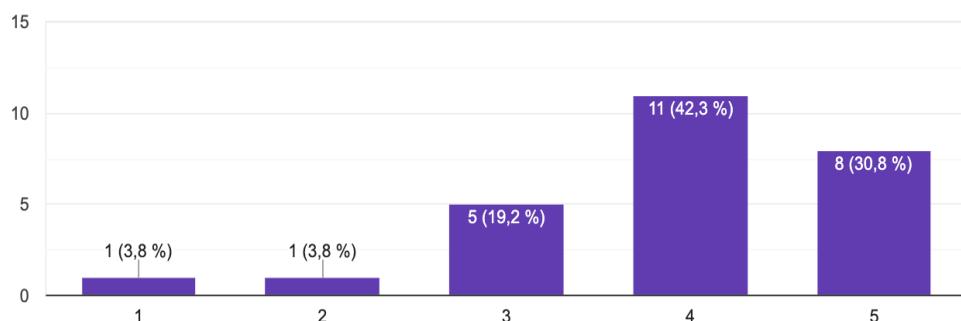


Figura 21: Încrederea în utilizarea unui astfel de sistem

Deși rezultatele anterioare au arătat că participanții au încredere în fluxurile de informații ale sistemului de vot, există îndoieri legate de potențialul tehnologiei blockchain cu privire la împiedicarea fraudării și conservarea *privacy-ului*. Conform figurii 21, aproape 73% dintre participanți prezintă o încredere mare sau totală într-un astfel de sistem de vot electronic, în timp ce restul de 27% consideră că nu își pot exprima votul în totală siguranță și anonimitate.

Sumarizând cele prezentate, aplicația de vot electronic PoliVote prezintă susținere și încredere din partea publicului, confirmând nevoia implementării la scară largă a unui astfel de sistem, alături de o companie de informare a publicului cu privire atât la tehnologia utilizată, cât și la modul de funcționare.

7 CONCLUZII

În această lucrare, am prezentat o soluție alternativă la metodele tradiționale, costisitoare și netransparente pentru organizarea scrutinelor electorale, implementând un sistem de vot electronic ce folosește beneficiile tehnologiei blockchain pentru a oferi siguranță și încredere în exprimarea votului.

PoliVote este un sistem de vot electronic ce oferă o soluție simplă pentru organizarea unui proces electoral. Acesta asigură atât votarea anonimă, cât și calcularea rapidă, eficientă și corectă a rezultatelor. Prin intermediul său, participanții la scrutinul de vot pot alege ușor și informații candidatul preferat, fără a se preocupa de lipsa anonimatului sau de posibilitatea fraudării voturilor. Toate aceste avantaje sunt oferite de utilizarea rețelei blockchain Ethereum, care, prin contractul intelligent implementat, dirijează fluxurile de informații și momentele procesului electoral.

Evaluând rezultatele obținute atât în urma chestionarului, cât și a scenariilor de testare, PoliVote se dovedește a fi o soluție promițătoare de vot electronic care, prin modificări ulterioare, ar putea fi utilizată la scară largă.

7.1 Dezvoltări ulterioare

În ceea ce privește evoluția acestui sistem de vot, există în plan următoarele idei:

- Extinderea sistemului prin dezvoltarea unei aplicații mobile.
- Îmbunătățirea și eficientizarea criptării datelor prin utilizarea unor algoritmi mai eficienți și mai siguri.
- Folosirea *side-chain-urilor* pentru a crește rata de procesare a tranzacțiilor.
- Implementarea autentificării multi-factor.
- Implementarea unui algoritm de tipul *Zero Knowledge Proof* pentru semnătura votanților.
- Extinderea rețelei blockchain curente astfel încât să poată fi accesată din internet.

BIBLIOGRAFIE

- [1] Agora. Bringing our voting systems into the 21st century. https://static1.squarespace.com/static/5b0be2f4e2cccd12e7e8a9be9/t/5f37eed8cedac41642edb534/1597501378925/Agora_Whitepaper.pdf. Ultima accesare: 17 iunie 2023.
- [2] Toras Batubara, Syahril Efendi, and Erna Nababan. Analysis performance bcrypt algorithm to improve password security from brute force. *Journal of Physics: Conference Series*, 1811:012129, 03 2021. Ultima accesare: 18 iunie 2023.
- [3] Bin Hu, Zongyang Zhang, Jianwei Liu, Yizhong Liu, Jiayuan Yin, Rongxing Lu, and Xiaodong Lin. A comprehensive survey on smart contract construction and execution: paradigms, tools, and systems. 2020. Ultima accesare: 1 iunie 2023.
- [4] Bryan Ford. How Do You Know It's On the Blockchain? With a SkipChain. <https://bford.info/2017/08/01/skipchain/>, 2017. Ultima accesare: 17 iunie 2023.
- [5] Nithin Chandran and Ebin Manuel. Performance analysis of modified sha-3. *Procedia Technology*, 24:904–910, 12 2016.
- [6] Dave Roos. How People Voted in Ancient Elections. <https://www.history.com/news/ancient-elections-voting>, 2022. Ultima accesare: 10 iunie 2023.
- [7] Itai Dinur, Orr Dunkelman, and Adi Shamir. New attacks on keccak-224 and keccak-256. In Anne Canteaut, editor, *Fast Software Encryption*, pages 442–461. Springer Berlin Heidelberg, 2012. Ultima accesare: 18 iunie 2023.
- [8] followmyvote.com. Elliptic Curve Cryptography. <https://followmyvote.com/elliptic-curve-cryptography/>, 2021. Ultima accesare: 15 iunie 2023.
- [9] Grace Jansen. Advantages of the event-driven architecture pattern. <https://developer.ibm.com/articles/advantages-of-an-event-driven-architecture/>, 2020. Ultima accesare: 15 iunie 2023.
- [10] Kishor Datta Gupta, Md Lutfar Rahman, Dipankar Dasgupta, and Subash Poudyal. Shamir's secret sharing for authentication without reconstructing password. In *2020 10th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0958–0963, 2020. Ultima accesare: 18 iunie 2023.
- [11] Jahid Hasan. Overview and applications of zero knowledge proof (zkp). 8:5, 10 2019. Ultima accesare: 18 iunie 2023.

- [12] Uzma Jafar, Mohd Aziz, and Zarina Shukur. Blockchain for electronic voting system—review and open research challenges. *Sensors*, 21:5874, 08 2021. Ultima accesare: 12 iunie 2023.
- [13] Kaspersky Lab. Polys online voting system. https://polysdocs.website.yandexcloud.net/Whitepaper/7262_WP_Polys_En_WEB_7.pdf, 2021. Ultima accesare: 18 iunie 2023.
- [14] Sándor Király, Szilveszter Székely, Roland Király, and Tamás Balla. Some aspects of using rpc. pages 145–156, 01 2018. Ultima accesare: 18 iunie 2023.
- [15] Bahareh Lashkari and Petr Musilek. A comprehensive review of blockchain consensus mechanisms. *IEEE Access*, 9:43620–43652, 2021. Ultima accesare: 15 iunie 2023.
- [16] Izhar Mehar, Charles Shier, Alana Giambattista, Elgar Gong, Gabrielle Fletcher, Ryan Sanayhie, Henry Kim, and Marek Laskowski. Understanding a revolutionary and flawed grand experiment in blockchain: The dao attack. *Journal of Cases on Information Technology*, 21:19–32, 01 2019. Ultima accesare: 19 iunie 2023.
- [17] Blessing Ngonidzashe Musungate, Büşra Candan, Umut Can Çabuk, and Gökhan Dalkılıç. Sidechains: Highlights and challenges. In *2019 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–5, 2019. Ultima accesare: 18 iunie 2023.
- [18] Krishna Sampigethaya and Radha Poovendran. A survey on mix networks and their secure applications. *Proceedings of the IEEE*, 94(12):2142–2181, 2006. Ultima accesare: 18 iunie 2023.
- [19] Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. <https://bitcoin.org/bitcoin.pdf>, 2008. Ultima accesare: 18 iunie 2023.
- [20] Vitalik Buterin. Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform. https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf, 2014. Ultima accesare: 18 iunie 2023.
- [21] Gang Wang, Zhijie Shi, Mark Nixon, and Song Han. Sok: Sharding on blockchain. pages 41–61, 10 2019.