VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Mindaugas Morkūnas

ELEKTRONINIO BALSAVIMO PROTOKOLO BLOKŲ GRANDINĖS TECHNOLOGIJOS PAGRINDU SUKŪRIMAS

DEVELOPMENT OF ELECTRONIC VOTING PROTOCOL ON THE BASIS OF BLOCKCHAIN TECHNOLOGY

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų saugos studijų programa,
valstybinis kodas 6211BX014
Informatikos inžinerijos studijų kryptis

Vilnius, 2021

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

FUNDAMENTINIŲ MOKSLŲ FAKULTETAS

INFORMACINIŲ SISTEMŲ KATEDRA

Mindaugas Morkūnas

ELEKTRONINIO BALSAVIMO PROTOKOLO BLOKŲ GRANDINĖS TECHNOLOGIJOS PAGRINDU SUKŪRIMAS

DEVELOPMENT OF ELECTRONIC VOTING PROTOCOL ON THE BASIS OF BLOCKCHAIN TECHNOLOGY

Baigiamasis magistro darbas

Informacijos ir informacinių technologijų saugos studijų programa,
valstybinis kodas 6211BX014
Informatikos inžinerijos studijų kryptis

Vadovas prof. habil. dr. Antanas Čenys _____ _____
(Pedag. vardas, vardas, pavardė) (Parašas) (Data)

Lietuvių kalbos konsultantė dr. Vaida Buivydienė _____ _____
(Moksl. laipsnis, vardas, pavardė) (Parašas) (Data)

Vilniaus Gedimino technikos universitetas
Fundamentinių mokslų fakultetas
Informacinių sistemų katedra

ISBN ISSN
Egz. sk. .....
Data .....-.....-.....

Antrosios pakopos studijų **Informacijos ir informacinių technologijų saugos** programos magistro baigiamasis darbas

Pavadinimas **Elektroninio balsavimo protokolo sukūrimas blokų grandinės technologijos pagrindu**

Autorius **Mindaugas Morkūnas**

Vadovas **Antanas Čenys**

Kalba: lietuvių

Anotacija

Baigiamojo magistro darbo tikslas – sukurti elektroninio balsavimo protokolą, paremtą blokų grandinės technologija. Balsavimo proceso skaidrumui užtikrinti, procesas privalo būti perkeltas į elektroninę erdvę. Dėl šios priežasties balsavimo proceso perkėlimas į elektroninę erdvę yra svarbus klausimas, tačiau kol kas dar nėra galutiniai išspręstas. Egzistuojantys elektroninio balsavimo sprendimai nėra taikomi plačiai, nes duomenų saugumas nėra užtikrinamas visu šimtu procentų. Kuriamo naujo elektroninio balsavimo protokolo tikslas yra užtikrinti balsavimo proceso vykdymo kaštų sumažinimą iki minimalaus ir užtikrinti asmens duomenų saugumą, anonimiškumą balsuojant ir paties proceso skaidrumą. Analitinėje dalyje pateikiama populiariausių blokų grandinės protokolų ir egzistuojančių elektroninio balsavimo sprendimų analizė. Pagal gautus analizės rezultatus, nuspręsta sukurti naują elektroninio balsavimo sistemą, paremtą išmaniosiomis sutartimis ir „Elektroninės valdžios vartų“ sistemos integracija.
Projektinėje dalyje aprašomas kuriamos sistemos funkcionalumas, kuris atsispindi UML diagramose bei iškeltuose reikalavimuose. Dokumente taip pat aprašytas funkcinis, saugumo ir suderinamumo testavimas.

Darbą sudaro 6 dalys: įvadas, blokų grandinės protokolų ir elektroninio balsavimo sistemų analizė, elektroninio balsavimo sistemos prototipo realizacija ir testavimas, išvados, literatūros šaltiniai ir priedai. Darbo apimtis – 79 psl. teksto, 26 paveikslėliai, 5 lentelė, 30 šaltinių ir 3 priedai.

Prasminiai žodžiai: Elektroninis balsavimas, blokų grandinė, išmaniosios sutartys

Vilnius Gediminas Technical University
Faculty of Fundamental Sciences
Department of Information Systems

ISBN ISSN
Copies No. .....
Date .....-.....-.....

Master Degree Studies **Information and Information Technologies Security** study programme Master Graduation Thesis

Title **Development of Electronic Voting Protocol on the Basis of Blockchain Technology**

Author **Mindaugas Morkūnas**

Academic supervisor **Antanas Čenys**

Thesis language: Lithuanian

Annotation

The aim of the final master's thesis is to create an electronic voting protocol based on blockchain technology. To ensure the transparency of the voting process, the process must be transferred to the electronic space. For this reason, the transfer of the voting process to the electronic space is an important issue, but to date it has not been definitively resolved. Existing electronic voting solutions are not widely used because of data security lack. The aim of the new electronic voting protocol is to ensure that the costs of the voting process are kept to a minimum, ensure the security of personal data, the anonymity of the vote and the transparency of the process itself. The analytical part presents an analysis of the most popular blockchain protocols and existing electronic voting solutions. Based on the results of the analysis, it was decided to create a new electronic voting system based on smart contracts and with the integration of the e-Government Gateway system. The project part describes the functionality of the developed system, which is reflected in UML diagrams and requirements specification. The document also describes functional, security, and compatibility testing.

Structure: introduction, blockchain protocols and electronic voting systems analysis, electronic voting system prototype implementation and testing, conclusions, references, appendixes. Thesis consists of 79 text of pages, 26 pictures, 36 tables, 30 sources and 3 appendixes.

Keywords: Electronic voting, blockchain, smart contracts

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS
FUNDAMENTINIŲ MOKSLŲ FAKULTETAS
INFORMACINŲ SISTEMŲ KATEDRA

Informatikos inžinerija studijų kryptis
Informacijos ir informacinių technologijų sauga studijų programa,
valstybinis kodas .....

TVIRTINU
Katedros vedėjas
_____
(Parašas)
_____
(Vardas, pavardė)
_____
(Data)

BAIGIAMOJO MAGISTRO DARBO UŽDUOTIS
2020-10-22 Nr.
Vilnius

Studentui (ei) Mindaugui Morkūnui
(Vardas, pavardė)

Baigiamojo darbo tema: Elektroninio balsavimo protokolo blokų grandinės technologijos pagrindu sukūrimas (angl. Development of Electronic Voting Protocol on the Basis of Blockchain Technology).

patvirtinta 2019m. lapkričio 15 d. dekano potvarkiu Nr. 406fm

Baigiamojo darbo užbaigimo terminas 2021m. birželio 1 d.

BAIGIAMOJO DARBO UŽDUOTIS:

- Atlikti blokų grandinės technologijos analizę, išanalizuoti ir palyginti esamus elektroninio balsavimo sprendimus;
- Pasiūlyti elektroninio balsavimo metodą, kuris užtikrintų balsavimo proceso skaidrumą;
- Sukurti pasiūlytu metodu paremtą elektroninio balsavimo sistemą, atlikti testavimą eksperimentinėmis sąlygomis ir įvertinti pasiūlyto metodo praktinį pritaikymą.

Vadovas ..... prof. habil. dr. Antanas Čenys
(Parašas) (Moksl. laipsnis/pedag.vardas, vardas, pavardė)

Užduotį gavau
.....
(Parašas)
Mindaugas Morkūnas
.....
(Vardas, pavardė)
2019-10-22
.....
(Data)

Turinys

Lentelių sąrašas

Paveikslų sąrašas

Įvadas

Naujos technologijos suteikia galimybes mums įprastus, sudėtingus ir monotoniškus darbus perkelti į kompiuterizuotas sistemas ir taip palengvinti gyvenimą, išvengti žmogiškų klaidų ten, kur jos neleistinos ir gali brangiai kainuoti. Tarp šių užduočių galima įtraukti ir balsavimo procesą, kurio galutinis balsavimo rezultatas gali nulemti šalies ateitį. Tačiau šis ganėtinai paprastas procesas dar nėra visiškai kompiuterizuotas ir reikalauja fizinio žmogaus dalyvavimo: balsuoti reikia atvykti į rinkimų centrą ir pažymėti nuomonę biliete. Šį procesą turi kontroliuoti nepriklausomi asmenys, kurie atsakingi už balsavimo skaidrumą ir sąžiningumą. Net balsų skaičiavimas atliekamas rankiniu būdu, o tai užtrunka nemažai laiko. Informacinės technologijos padeda kompiuterizuoti tokius procesus ir juos atlikti kelis kartus greičiau bei efektyviau.

Elektroninis balsavimas pirmą kartą valdžios rinkimuose buvo panaudotas Estijoje 2005 m. Tais metais elektroniniu būdu balsavo 1,9 % visų turinčių teisę balsuoti Estijos gyventojų. Kiekvienais metais elektroniniu balsavimu besinaudojančių skaičius auga ir 2019 m. pasiekė 43,8 % (i-Voting — e-Estonia, 2020).

Egzistuojantys elektroninio balsavimo sprendimai neišsprendžia duomenų saugumo, konfidencialumo problemų, kurios yra svarbios ir balsuojant įprastai, nes naudojamos sistemos yra centralizuotos ir rinkėjai turi pasitikėti balsavimo organizatoriais, kad duomenys bus tvarkomi atitinkamai ir nebus pakeisti. Dėl šių dabartinio elektroninio balsavimo spragų kyla klausimas, kuo dabartinis elektroninis balsavimas geriau už paprastą balsavimo būdą?

Blokų grandinės technologijos atsiradimas padarė didelį perversmą informacinių sistemų raidai. Greitas blokų grandinės technologijos vystymasis ir potencialūs decentralizavimo sprendimai gali pakeisti ir palengvinti daug mums tenkančių darbų. Kaip pavyzdys galėtų būti bankai ir skaitmeninė valiuta, kuri leidžia atsiskaityti už paslaugas be jokių tarpininkų, tokių kaip bankas ar „PayPal“ mokėjimo sistema.

Blokų grandinės technologijos pritaikymas elektroniniam balsavimui turėtų išspręsti kylančias duomenų saugumo, konfidencialumo ir balsavimo proceso skaidrumo problemas. Decentralizavimo būdas leidžia tai pasiekti ir balsavimo procesą visiškai perkelti į elektroninę erdvę bei užtikrinti jo skaidrumą. Kadangi centralizuotos sistemos yra lengvai pažeidžiamos, jų saugumui užtikrinti reikalingi sudėtingi saugumo sprendimai, kurie yra brangūs ir reikalauja pastovios priežiūros.

1.1 Tyrimo objektas

Tyrimo objektas – blokų grandinės technologijos taikymas elektroniniam balsavimui.

1.2 Darbo tikslas ir uždaviniai

Darbo tikslas – sukurti elektroninio balsavimo metodą, paremtą blokų grandinės technologija, kuris užtikrintų balsavimo skaidrumą.

Darbo uždaviniai:

- Atlikti blokų grandinės technologijos analizę, išanalizuoti ir palyginti esamus elektroninio balsavimo sprendimus;
- Pasiūlyti elektroninio balsavimo metodą, kuris užtikrintų balsavimo proceso skaidrumą;
- Sukurti pasiūlytu metodu paremtą elektroninio balsavimo sistemą, atlikti testavimą eksperimentinėmis sąlygomis ir įvertinti pasiūlyto metodo praktinį pritaikymą.

1.3 Temos naujumas

Blokų grandinė yra ganėtinai nauja technologija ir jos taikymas yra labai platus. Elektroninis balsavimas yra viena iš potencialių blokų grandinės technologijos panaudojimo sričių, kurios skaidrumą gali užtikrinti blokų grandinės vientisumas ir nedalumas. Elektroninis balsavimas paremtas blokų grandinės technologija šiuo metu nėra plačiai taikomas, tačiau Siera Leonėje 2018 m. pirmą kartą prezidento rinkimai buvo vykdomi būtent šia technologija paremta elektroninio balsavimo sistema.

1.4 Temos aktualumas

Atliekamas darbas yra aktualus, nes balsavimo protokolo, kuris užtikrintų elektroninio balsavimo skaidrumą ir būtų tarptautiškai pripažintas, kaip saugus ir patikimas, nėra. Toks protokolas padėtų sumažinti balsavimo vykdymo ir rezultatų skaičiavimo ne tik kaštus, bet ir visam procesui skirtą laiką. Taip pat dėl blokų grandinės technologijos galima decentralizuoti visą balsavimo procesą ir informaciją, taip suteikiant saugumo ir anonimiškumo, o tai yra svarbiausia, vykdant balsavimo procesą ir jam pasibaigus.

1.5 Tyrimo metodika

Analitinėje darbo dalyje, atliekant susijusios literatūros ir jau egzistuojančių sprendimų analizę naudojami bibliotekinio tyrimo ir lyginamosios analizės metodai. Darbo tikslui įgyvendinti ir rezultatui gauti naudojami konstravimo ir eksperimentinės analizės metodai.

1.6 Darbo struktūra

Šiame skyriuje aprašomas darbo tikslas ir uždaviniai, kurie turi būti įgyvendinti iškeltam tikslui pasiekti. Tai pat supažindinama su darbo tema ir jos naujumu bei aktualumu.

Antrame skyriuje yra atliekama blokų grandinės technologijos ir egzistuojančių protokolų analizė, pateikiant jų technologinius skirtumus. Taip pat šiame skyriuje aprašomi blokų grandinės technologijos taikomi metodai ir technologiniai sprendimai, bei jų trūkumai ir privalumai. Taip pat analizuojami egzistuojantys elektroninio balsavimo sprendimai ir kitų autorių pasiūlymai šiai problemai spręsti.

Trečiame skyriuje pateikiamas siūlomo sprendimo aprašas, kuris buvo priimtas remiantis antrame skyriuje išanalizuotos literatūros bei egzistuojančių sprendimų analizės rezultatais.

Ketvirtame skyriuje yra pateikiama elektroninio balsavimo sistemos paremtos blokų grandinės technologija sistemos prototipo reikalavimų specifikacija, sistemos aktorių panaudos atvejų, sistemos objektų būsenų diagramos ir bendra sistemos architektūra, bei jos realizacija ir testavimas.

2 Blokų grandinės ir elektroninio balsavimo sistemų analizė

2.1 Blokų grandinės technologijos apžvalga

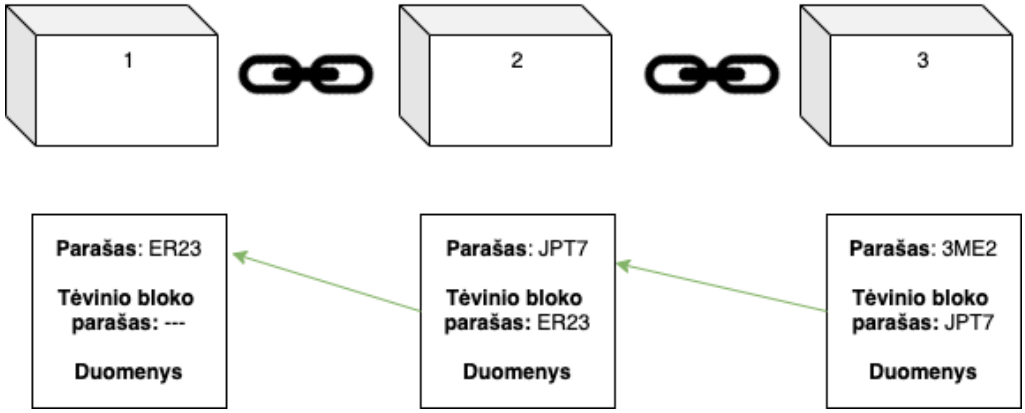
Blokų grandinė tai technologija, kuri išpopuliarėjo kartu su bitkoino kriptovaliuta. Bitkoinas pirmą kartą buvo aprašytas 2008 m. straipsnyje „Bitcoin: A Peer-to-Peer Electronic Cash System“ (Nakamoto, 2008). Straipsnyje blokų grandinės technologija nėra įvardijama ir pateikiama kaip vienas bendras sprendimas pavadinimu – bitkoinas. Sąvoka blokų grandinė atsirado kiek vėliau ir yra pateikiama, kaip decentralizuota transakcijų sistema (Yli-Huumo, Ko, Choi, Park, & Smolander, 2016). Šio straipsnio tikslas buvo pateikti naują technologiją pavadinimu bitkoinas, kuri pakeistų jau egzistuojančių bankų centralizuotas sistemas. Pasak autoriaus pagrindinė idėja yra atsisakyti pasitikėjimo kitomis įstaigomis, šiuo atveju bankais ir tarpininkais atliekant pinigines transakcijas, o turėti elektroninę mokėjimų sistemą paremtą kriptografija (Nakamoto, 2008). Trečiųjų šalių t. y. bankų, piniginių transakcijų tarpininkų tikslas yra užtikrinti, kad piniginės transakcijos būtų atliekamos tarp dviejų ar daugiau vartotojų ir visa tai būtų užregistruojama jų registre, tai yra daroma, kad būtų žinoma kas, kam ir kiek pinigų pervedė. Remiantis šia informacija yra manipuliuojama banko sąskaitose esančia pinigų suma, gaunant pinigus ji yra padidinama, o pervedus kitam sumažinama. Dėl to vartotojai yra priklausomi nuo trečiųjų šalių, nes jie saugo visą informaciją apie banko sąskaitas ir visas įvykdytas transakcijas. Iš to išplaukia centralizuotos sistemos problema, kad niekas negali patvirtinti ar transakcijos yra teisingos, tai gali garantuoti tik bankai ir vartotojams belieka jais pasitikėti. Kita problema atsiranda tuo atveju jeigu trečioji šalis šiuo atveju bankas praranda visą informaciją apie įvykdytas transakcijas ir jų atstatyti nebeįmanoma. Šias problemas ir sprendžia blokų grandinės technologija, finansų srityje.

Į blokų grandinės technologiją galima žiūrėti, kaip į decentralizuotą duomenų bazę ir tinklą, kuris suteikia saugumą, anonimiškumą ir duomenų vientisumą atsiribojant nuo trečiųjų šalių, kurios kontroliuoja visas transakcijas ir iš to užsidirba (Yli-Huumo et al., 2016). Kodėl tai yra duomenų bazė ir tinklas, kuriame vyksta visa komunikacija? Pirmiausia blokų grandinės technologijos pagrindinė idėja yra decentralizavimas, kuris yra apibrėžiamas kaip kontrolės teisių perkėlimas iš centrinės valdymo sistemos, visiems vartotojams (Bardhan, 2002). Blokų grandinės technologijos atveju, decentralizavimas reiškia transakcijų vykdymą tiesiogiai tarp dviejų vartotojų be jokių tarpininku. Kadangi kiekvienas blokas saugo atitinkamą informaciją ir blokų visuma sudaro blokų grandinę, kurios kopijos yra saugomos pas kiekvieną vartotoją, dėka šio sprendimo pavyksta decentralizuoti duomenis. Taip pat tai yra ir tinklas, nes

kiekvienas vartotojas tinkle komunikuoja tarpusavyje, o komunikacija vykdoma atliekant transakcijas.

2.1.1 Blokų grandinės architektūra

Blokų grandinė susideda iš dviejų sudedamųjų dalių, kurios ir sudaro pavadinimą tai: blokas (*angl. Block*) ir grandinė (*angl. Chain*). Pagrindinis architektūrinis blokų grandinės sprendimas yra tas, kad grandinė yra sudaryta iš atskirų blokų, iš kurių kiekvienas saugo atitinkamą informaciją apie atliktą transakciją ir tėvinio bloko parašą (Zheng, Xie, Dai, Chen, & Wang, 2017). Toks blokų sujungimas į grandinę leidžia užtikrinti jos nedalumą ir vientisumą.



1 pav. Blokų grandinės struktūra

Kaip galime matyti iš aukščiau pateiktos blokų gardinės struktūros schemos, toks blokų grandinės sprendimas padaro grandinę nedalomą ir nepakeičiamą, pakeitus nors vieno bloko informaciją arba pašalinus atsitiktinį bloką, grandinė tampa negaliojanti, nes kažkurio bloko tėvinio bloko parašas nesutaps su vaiko bloke saugoma informacija. Sekantis svarbus aspektas apie blokų grandinę yra tas, kad atsiradus naujam blokui grandinėje jis yra pasidalinimas tarp visų vartotojų, kurie yra tame pačiame blokų grandinės tinkle, taip yra įgyvendinimas decentralizavimas. Dėl to blokų grandinės tinklą yra lengviausia įsivaizduoti, kaip tinklą, kuriame visi vartotojai yra sujungti tarpusavyje be jokių tarpininkų. Toks tinklas blokų grandinės terminologijoje yra vadinimas lygiarangių tinklas (*angl. Peer-to-Peer network*) (Swan, n.d.). Kiekvienas įrenginys prijungtas prie blokų grandinės tinklo yra mazgas (*angl. Node*), kuris yra sujungtas su kitais mazgais. Kiekvienas naujas mazgas tokiame tinkle gauna kopiją blokų grandinės, kurios kiekvienas blokas saugo informaciją apie jau atliktas transakcijas. Tai reiškia, kad kiekviena transakcija buvo atlikta būtent tarp tų mazgų ir ta informacija yra išsaugoma visuose mazguose ir visam laikui. Toks sprendimas suteikia blokų grandinės technologijai papildomą saugumo lygį, kadangi kiekvienas mazgas turi visą blokų

grandinę ir norint ją pakeisti, tai reikėtu padaryti visuose kituose mazguose, tai reiškia, kad kuo daugiau mazgų yra tinkle tuo saugesnis jis yra.

2.1.2 Egzistuojantys protokolai

Blokų grandinė tai bendra technologija, kuri aprašo principus ir naudojamus metodus. O protokolas tai technologijos įgyvendinimas ir pritaikymas praktikoje. Šiai dienai egzistuoja daugybė blokų grandinės protokolų, kurie siūlo įvairius sprendimus, sprendžiančius saugumo, greitaveikos ar panaudojimo problemas savaip. Tačiau visų protokolų tėvu galima laikyti bitkoino protokolą, kuris buvo sugalvotas ir sukurtas kartu su blokų grandinės technologija. Šioje darbo dalyje pateiksiu ir apžvelgsiu trys pagrindinius ir populiariausius blokų grandinės protokolus: bitkoinas, „Ethereum“ ir „Ripple“.

2.1.2.1 Bitkoino protokolas

Bitkoinas – tai virtuali valiuta, kurios atsiradimas davė pradžią blokų grandinės technologijai. Pagrindinės bitkoino protokolo charakteristikos: galimybė atlikti transakcijas tiesiogiai be trečiųjų šalių, nėra galimybės atšaukti arba pašalinti jau įvykusios transakcijos, sumažina transakcijų kainą ir kaštus lyginant su centralizuotomis bankų sistemomis, išsprendžia dvigubo pinigų panaudojimo (*angl. Double-Spending*) problemą (Nakamoto, 2008). Visiems šitiems sprendimams įgyvendinti yra naudojami šie metodai:

- Maiša (*angl. Hash*);
- Skaitmeninis parašas;
- Viešo rakto kriptografija (*angl. Public-key Cryptography*);
- Lygiarangių tinklas;
- Darbo įrodymo principas.

Šių metodų ir sprendimų rinkinys neleidžia dubliuoti mokėjimų ir keisti duomenų, tai užtikrina decentralizuotos valiutos skaidrumą, saugumą.

2 pav. Bitkoino kriptovaliutos logotipas (Bitcoin - Open source P2P money, 2019)

2.1.2.2 „Ethereum“ protokolas

„Ethereum“ – tai blokų grandinės technologija pagrįstas, atvirojo kodo protokolas. „Ethereum“ protokolo idėja buvo pristatyta 2012 m. programuotojo Vitalik Buterin publikuotame straipsnyje „Ethereum White Paper“, jis aprašė naujo protokolo veikimo principą ir pagrindinius skirtumus tarp bitkoino ir naujo siūlomo sprendimo, dabar žinomo kaip išmaniosios sutartys (*angl. Smart Contracts*) (Buterin, 2012). Išmaniosios sutartys – automatiškai įgyvendinami susitarimai veikiantys blokų grandinės tinkle. Pagrindinė sutarčių idėja įvykdyti susitarimą įgyvendinus aprašytas sąlygas, ką ir padaro išmaniosios sutartys automatiškai. Visos sutartys yra programuojamos specialiai sukurta „Solidity“ programavimo kalba, kuri yra labai panaši į vieną populiariausių programavimo kalbų „Javascript“. Sutartyje yra aprašomos sąlygos kurios turi būti įvykdytos sėkmingai, kad sutartis būtų laikoma sėkmingai įvykdyta. Jeigu nors viena sutarties sąlyga yra neįvykdomos, automatiškai sutartis laikoma neišpildyta ir toliau ji nėra vykdoma. „Solidity“ programavimo kalbos programinis išeities kodas yra kompiliuojamas į „Ethereum Virtual Machine“ baitų kodą ir joje vykdomas. Visos išmaniosios sutartys yra vykdomos „Ethereum“ blokų grandinės tinkle. Pagrindinės išmaniųjų sutarčių savybės (Tso, Liu, & Hsiao, 2019):

- Nekintamumas (*angl. Immutable*);
- Paskirstymas (*angl. Distributed*).

Blokų grandinės tinkle veikiančios išmaniosios sutartys yra nekintančios, tai reiškia, kad niekas negali pakeisti jų sąlygų, šis principas užtikrina, kad sutartys nebus pakeistos norint pasipelnyti arba blogiems tikslams pasiekti. Sekanti savybė – paskirstymas. Ši savybė leidžia užtikrinti, kad visi priklausantys sutarčiai arba dalyvaujantys joje žino visas sąlygas ir sukčiavimo atveju bus lengva nustatyti ir atsekti, kad yra bandoma sukčiauti. Nes vienam vartotojui nurodžius jog sutarties sąlygos yra įgyvendintos, tai yra patikrinama ir pas kitus tame pačiame tinkle esančius vartotojus, toks pasiskirstymas leidžia užtikrinti išmanių sutarčių skaidrumą. Kiekvienas iš „Ethereum“ naudotojų gali sukurti savo išmaniąją sutartį ir ją vykdyti blokų grandinės tinkle, tačiau kiekviena operacija šiame tinkle yra apmokestinama.

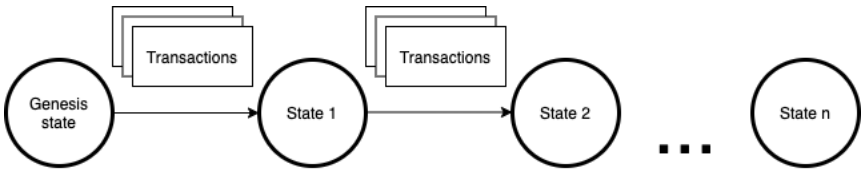
Protokolo tikslas decentralizuoti visą internetą ir tokius procesus kaip elektroninės parduotuvės, bankai ar net balsavimas. Protokolas veikia globalioje decentralizuotoje virtualioje mašinoje (*angl. Virtual Machine*) pavadinimu – „Ethereum Virtual Machine“. Šios virtualios mašinos tikslas vykdyti programinio kodo scenarijus (*angl. Script*) naudojant išorinį interneto tinklą ir „Ethereum“ protokolui priklausančią „Ether“ kriptovaliutą (Luu, Chu, Olickel, Saxena, & Hobor, 2016).

ethereum

3 pav. „Ethereum“ logotipas (Home | Ethereum, 2019)

Dėl išmaniųjų sutarčių lengvo sudarymo ir vykdymo, „Ethereum“ protokolas greitai tapo itin populiarus, nes panaudojimas labai platus, pavyzdžiu: decentralizuotų programų kūrimas, išmaniųjų sutarčių sudarymas ir decentralizuotų autonominių organizacijų protokolas.

„Ethereum“ iš esmės yra transakcijomis pagrįsta būsenų (*angl. Transaction-Based)* mašina. Tai reiškia, kad naudojant „Ethereum“ būsenų mašiną pirmoji būsena nuo kurios pradedama vadinama, kilmės būsena (*angl. Genesis State*). Tokia pradinė būsena nurodo, kad iki šiol blokų grandinės tinkle nebuvo vykdoma jokių transakcijų. Vykdant operacijas blokų grandinės tinkle, kilmės būsena pereina į tam tikrą galutinę būseną (*angl. Final State*), kuri šiuo metu reiškia ir dabartinę ir galutinę „Ethereum“ mašinos būseną.



4 pav. „Ethereum“ blokų grandinės mašinos būsenų diagrama (Buterin, n.d.)

Viena būsena gali saugoti milijonus transakcijų savyje. Transakcijos yra grupuojamos į blokus, kurių kiekvienas blokas turi ryši su prieš tai esančiu bloku ir taip yra sudaroma blokų grandinė. Kiekvienas blokas yra tikrinimas „kalnakasio“ (*angl. Miner*) blokų grandinės tinkle ir jeigu atlikus matematinius skaičiavimus blokas yra teisingas jis yra pridedamas į pagrindinę blokų grandinę, šis procesas yra vadinamas – kasyba (*angl. Mining*).

Kasyba yra naudojama blokų grandinės tinkle, kaip įrankis leidžiantis užtikrinanti blokų grandinės vientisumą ir duomenų saugumą. Darbo įrodymo (*angl. Proof-of-Work*) tikslas yra kriptografiniu būtų įrodyti, kad grandinė yra teisinga. Dėl kiekvieno bloko ryšio vienas su kitu, t. y. kiekviename bloke yra ankstesnio bloko maišos vertė, todėl sugeneravus sekantį bloką, ankstesnio bloko turinio negalima lengvai pakeisti, tačiau jeigu blokas buvo

sugadintas, tada yra paveikiami ir kiti ankstesni blokai, tai reiškia, kad norint sugadinti viena bloką, reikia perskaičiuoti visų blokų maišos reikšmes, tačiau norint pasiekti šį tikslą reikia labai daug skaičiavimo galios, nes didėjant blokų grandinei sunkėja jos patikrinimas.

„Ethereum“ protokolas susideda iš šių pagrindinių komponentų (Buterin, n.d.):

- Paskyros;
- Dujos (*angl. Gas*);
- Mokesčiai (*angl. Fees*);
- Transakcijos;
- Blokai;
- Transakcijų vykdymas;
- Kasyba;
- Darbo įrodymas.

Paskyros – tai „Ethereum“ bendros būsenos (*angl. Shared-State*) objektai, galintys komunikuoti tarpusavyje naudojant pranešimų perdavimo sistemą. Kiekviena tokia paskyra turi jai priskirtą 20 baitų adresą, kuris reikalingas paskyros identifikavimui. Paskyros gali būti dviejų tipų:

1. Sutarčių paskyros (*angl. Contract Accounts)* – šios paskyros yra kontroliuojamos pagal jų sutarties kodą ir turi su jomis susietą unikalų numerį;
2. Išoriškai valdomos paskyros (*angl. Externally owned accounts*) – šios paskyros yra valdomos naudojant privačius raktus ir priešingai nei sutarčių paskyros neturi su jomis susieto numerio.

Pagrindinis skirtumas tarp šių dviejų paskyrų tipų yra tas, kad iš išoriškai valdomos paskyros galima siųsti pranešimus tokio pat tipo paskyroms arba sutarčių tipo paskyroms, tas transakcijas pasirašant savo privačiu raktu. Priešingu atveju sutarčių paskyros negali pačios inicijuoti transakcijos, šios paskyros gali tik atsakyti į kitas gautas transakcijas vykdant paskyroje aprašytus kodus.

„Dujos“ ir mokesčiai – viena iš svarbiausių „Ethereum“ koncepcijų yra mokesčių politika. Mokestis yra imamas už kiekvieną „Ethereum“ tinkle atliktą skaičiavimo operaciją. Toks mokestis yra mokamas „dujomis“, tai mokesčio nominalas „Ethereum“ tinklo ekosistemoje. „Dujos“ yra vienetas, naudojamas apskaičiuoti mokesčius, reikalingus tam tikriems skaičiavimo veiksmams atlikti. „Dujų“ kaina yra maža dalis „Ether“ kriptovaliutos, kurią norima išleisti kiekvienam dujų vienetui. Kiekvienoje transakcijoje siuntėjas privalo

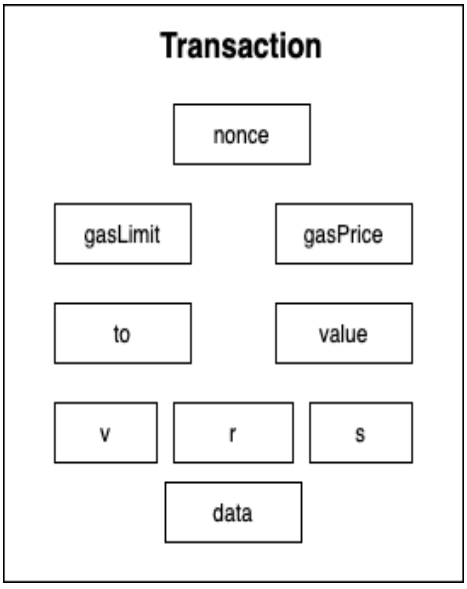
nurodyti kiek daugiausiai nori sumokėti „dujų“ ir siūlomą kainą už jas. „Dujų“ kainos produktas ir „dujų“ limitas rodo maksimalią „dujų“ kainą, kurią siuntėjas nori sumokėti už transakcijos įvykdymą.

Transakcijos – tai šifruota instrukcija, kurią sukuria išoriškai valdoma paskyra, kuri vėliau yra verifikuojama ir perduodama į blokų grandinės tinklą. Taip pat transakcijos yra skirstomos į du tipus:

1. Žinučių skambučiai (*angl. Message Calls*);
2. Sutarčių sudarymas (*angl.* Contract Creations).

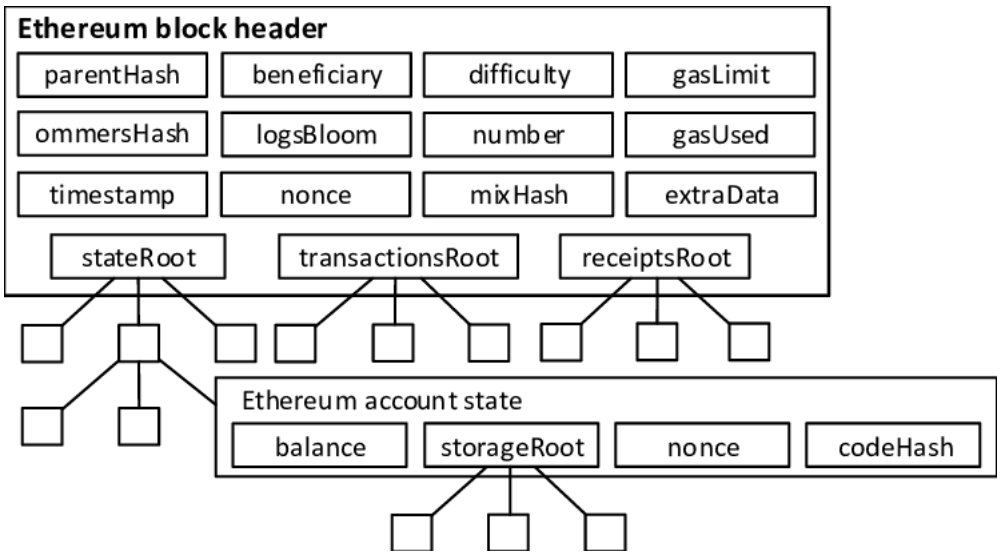
Sutarčių sudarymo transakcijos reikalingos kuriant naujus „Ethereum“ išmaniąsias sutartis. Kiekvieną transakciją sudaro šie komponentai (Buterin, n.d.):

- **„nonce“** – ši reikšmė nurodo, kiek transakcijų atliko siuntėjas;
- **„gasPrice“** – suma kurią siuntėjas nori sumokėti už transakcijos įvykdymą;
- **„gasLimit“** – maksimali suma, kurią siuntėjas nori sumokėti už transakcijos įvykdymą;
- **„to“** – gavėjo unikalus adresas;
- **„value“** – suma, kurią siuntėjas nori persiųsti gavėjui;
- **„v“, „r“, „s“** – šios reikšmes naudojamos generuojant transakcijos siuntėjo skaitmeninį parašą;
- **„data“** – neprivalomas laukelis, naudojamas nurodyti papildomai informacijai arba pateikti reikalingus parametrus;



5 pav. „Ethereum“ transakcijos struktūra

Blokai – tai pagrindinė blokų grandinės dalis iš kurios ir susidaro visa neskaidoma grandinė. Blokai yra sudaryti iš šių sudedamųjų dalių: bloko antraštė (*angl. Block Header*), informacija apie transakcijas esančias bloke ir rinkinys dviejų ankstesnių blokų antraščių. Pagrindinė bloko dalis, kuri saugo visą informaciją yra bloko antraštė. Bloko antraštėje yra saugoma tokia informacija:



6 pav. „Ethereum“ bloko antraštė (Weber et al., 2019)

2.1.2.3 „Ripple“ protokolas

„Ripple“ – tai transakcijų protokolas, sukurtas 2012 m., kaip atviro kodo paskirstytas sutarčių, interneto protokolas ir jam priklausanti valiuta pavadinimu „Ripples“ (Baliga, 2017). „Ripple“ protokolas suteikia galimybę greitai, beveik nemokamai ir saugiai atlikti bet kokias finansines operacijas. Protokolo privalumas prieš kitus protokolus tas, kad jis gali palaikyti ne tik savo kriptovaliutą „Ripples“ tačiau ir kitus žetonus (*angl. Tokens*), kurie reprezentuoja kitas kriptovaliutas.

7 pav. „Ripple“ logotipas (Instantly Move Money to All Corners of the World | Ripple, 2020)

2.1.2.4 Išmaniosios sutartys

Išmaniosios sutartys tai pagrindinė „Ethereum“ blokų grandinės technologijos koncepcija, kuri leidžia vykdyti nedideles užduotis, programas decentralizuotame blokų grandinės tinkle. Dėl išmaniųjų sutarčių paprasto programavimo ir nedidelės vykdymo kainos, jos greitai tapo itin populiarios, iškeliant „Ethereum“ protokolo populiarumą tarp kitų kriptovaliutų.

Išmaniųjų sutarčių pagrindinės savybės yra išskiriamos šios (Antonopoulos & Wood, n.d.):

- **Paprastumas** – išmaniosios sutartys yra tiesiog paprastos kompiuterinės programos, skirtumas tik tas, kad joms programuoti naudojama specialiai tam sukurta „Solidity“ programavimo kalba;
- **Nekintamumas** – kai išmanioji sutartis pradedama vykdyti, jos pakeisti neįmanoma, toks sprendimas leidžia užtikrinti, kad niekas eigoje negalės pakeisti sutarties taisyklių ir tuo pasinaudoti. Lyginant su tradicinėmis programomis, išmaniąsias sutartis galima modifikuoti tik vienu būdų, sukuriant naują sutartį;
- **Vientisumas** – kiekvienas vartotojas, kuris vykdo tapačią sutartį, visada gauna tą patį rezultatą;
- **„Ethereum“ virtuali mašina** – išmaniosios sutartys veikia labai ribotame kontekste, kurį sudaro „Ethereum“ virtuali mašina. Išmaniosios sutartys gali sužinoti tik savo būseną, peržiūrėti kontekstą tos transakcijos, kurios metu ji buvo iškviesta ir gali gauti informacijos apie naujausiai sukurtus blokus;
- **Decentralizavimas** – „Ethrereum“ virtuali mašina veikia kiekviename iš „Ethereum“ mazgų kaip lokali virtuali mašina, tačiau kadangi visi „Ethrereum“ virtuali mašina pradeda toje pačioje būsenoje „Genesis state“ ir sukuria tą pačią galutinę būseną „Final state“, tai tokia sistema vadinasi veikia, kaip vienas, bendras kompiuteris, nes operuoja tomis pačiomis būsenomis tuo pačiu metu.

Išmaniosios sutartys yra programuojamos pasitelkiant aukšto lygio programavimo kalbas, tokias kaip anksčiau minėta „Solidity“ kalba, tačiau vėliau prieš pradedant jas vykdyti „Ethrereum“ virtualios mašinos kodas yra kompiliuojamas į žemo lygio baitų kodą ir tik tada vykdomas. Kai sutartis sukompiliuojama yra įvykdoma pradinė transakcija „Contract creation transaction“. Kiekviena nauja sutartis yra identifikuojama pagal „Ethereum“ šešioliktainį adresą, kuris yra gaunamas iš pirmosios transakcijos. Taip pat išmaniosios sutartys yra

vykdomos tik tada, kai jos yra iškviečiamos per transakcijas, kitu atveju jos negali būti vykdomos.

2.1.3 Protokolų palyginimas

Šiame poskyryje protokolų palyginimo lentelėje pateikiami svarbūs kriterijai, kurie parodo esminius skirtumus tarp analizuotų protokolų.

1 lentelė. Egzistuojančių protokolų palyginimas

NR.	Protokolas / Kriterijus	Bitkoinas	„Ethereum“	„Ripple“
1.	**Sukūrimo data**	2009 m.	2012 m.	2012 m.
2.	**Tiekėjas**	Satoshi Nakamoto	„Ethereum“ programuotojai	„Ripple Labs“
3.	**Pritaikymo sritis**	Finansai	Platus panaudojimas	Finansai
4.	**Kriptovaliuta**	Bitkoinas	„Ether“	„Ripple“
5.	**Naudojami mechanizmai**	„Proof of Work“	„Proof of Work“	Tikimybinis balsavimo protokolas
6.	**Išmaniosios sutartys**	-	+	-
7.	**Išmaniųjų sutarčių programavimo kalba**	-	Solidity	-
8.	**Kuriamų sistemų paskirtis**	Finansinės sistemos	Platus pritaikymas	Finansinės sistemos

Remiantis lentelės duomenimis, puikiai matoma, kad „Ethereum“ protokolas yra tinkamiausias elektroninio balsavimo sistemai kurti, nes jo pritaikymo sritis yra labai plati, dėka naudojamų išmaniųjų sutarčių, kurios ir suteikia papildomą lankstumą technologijos pritaikymui.

2.1.4 Blokų grandinės technologijos privalumai

Šiame poskyryje pateikiami blokų grandinės technologijos privalumai lyginant su centralizuotomis sistemomis. Blokų grandinės privalumai (Gatteschi, Lamberti, Demartini, Pranteda, & Santamaria, 2018; Nakamoto, 2008):

- Naudojama bendra saugykla, kuri yra palaikoma mazgų esančių blokų grandinės tinkle. Tai reiškia, kad kiekviename mazge saugoma kopija blokų grandinės ir tai neleidžia prarasti duomenų netikėtu atveju;
- Skaitmeninis parašas užtikrina, kad kiekvienas mazgas ir vartotojas elgiasi teisingai ir tam užtikrinti nereikia tarpininkų;
- Lankstumas. Ateityje blokų grandinės tinklas gali tapti decentralizuota duomenų saugykla, kurią gali skaityti ir pildyti kiekvienas vartotojas;
- Garantuotas skaidrumas;
- Nepakeičiamumas. Duomenų niekas negali pakeisti ar pašalinti;
- Decentralizacija. Technologija gali veikti be centrinės valdžios ir negali būti kontroliuojama jos;
- Naudojant išmanius kontraktus visi blokų grandinės tinkle vykstantys procesai gali būti vykdomi automatiškai pagal aprašytas taisykles.

2.1.5 Blokų grandinės technologijos kylančios grėsmės

Blokų grandinės technologijos kylančios grėsmės (Gatteschi et al., 2018; Lindman, Tuunainen, & Rossi, 2017):

- Darbo įrodymo, „Proof-of-Stake“ ir kiti blokų grandinėje naudojami mechanizmai reikalauja dideliu energijos resursų;
- Ankščiau minėtas blokų grandinės privalumas, duomenų vientisumas ir nepakeičiamumas gali būti traktuojamas kaip ir trūkumas. Keisti blokų grandinės duomenų ar programinį kodą yra labai sudėtinga, tokiam atvejui galimas tik vienas sprendimas, atsisakymas senos grandinės ir inicijuoti naują;
- Blokų grandinės tinkle naudoja asimetrinę (*angl. Asymmetric*) kriptografiją, kuri privataus rakto pagalba vartotojams suteikia nuosavybės teises turimai kriptovaliutai. Todėl atsiranda rizika, kad vartotojui praradus savo privatų raktą kriptovaliuta bus irgi prarasta, o atgauti prarastą kriptovaliutą yra nebeįmanoma;

- Greitai didėjantys blokų grandinių dydžiai. Šiuo metu bitkoino grandinei saugoti reikia apie 200 GB laisvos kietojo disko atminties. Dėl didelių grandinių tinklas rizikuoja prarasti mazgus, nes vartotojai nepajėgs tokių didelių grandinių atsisiųsti ir saugoti.

2.2 Egzistuojančių elektroninio balsavimo sprendimų analizė

Tradicinis mums visiems priimtinas balsavimo būdas, kai ateiname į balsavimo dieną paskirtą vietą, pasirašant ir užpildant balsavimo bilietą buvo sugalvotas vadovaujantis demokratijos principais. Kiekvienas šalies pilietis turi teisę ir prievolę balsuoti sprendžiant svarbius šalies klausimus ir renkant šalies valdžią. Balsavimas naudojant bilietus tenkina visus keliamus saugumo reikalavimus, tačiau yra labai brangus, nepatogus žmonėms negalintiems atvykti dėl asmeninių priežasčių arba dėl sveikatos problemų, taip pat reikalaujantis pasiruošimo ir savanorių pagalbos organizuojant balsavimo centrus ir registruojant bei administruojant balsuojančius piliečius. Vykstant balsavimui yra užtikrinamas balsuojančiųjų anonimiškumas. Atvykęs pilietis turintis balsavimo teisę yra identifikuojamas pagal jo asmens tapatybės pasą arba kortelę ir užregistruojamas, kad buvo atvykęs. Užregistravus pilietį, jam išduodamas anoniminis balsavimo bilietas, kuriame pateikiami galimi balsavimo variantai. Savo pasirinkimą pilietis privalo pažymėti varnele ir tai atlikti turi balsavimo būdelėje, kurioje niekas negalėtu matyti už ką balsavo, pažymėjus savo pasirinkimą bilietas yra įdėdamas į balsavimo bilietų urną ir procesas baigiamas. Pasibaigus visam balsavimo visi bilietai yra skaičiuojami ir tai užtrunką priklausomai nuo balsavusių žmonių skaičius. Šio etapo metu iškyla balsų sąžiningo skaičiavimo problema, nes yra pasitikima tais, kurie skaičiuoja balsus ir juos užregistruoja, tai reiškia, kad balsai gali būti praleisti arba papirkti, tai priklauso nuo balsavimo vykdymo komisijos. Tačiau šiai dienai šis balsavimo būdas yra laikomas saugiausiais ir lengviausiai kontroliuojamas valstybės.

Laikui bėgant buvo daugybė būdų ir sprendimų teikiama, kaip galima būtų kompiuterizuoti balsavimo procesą ir padaryti jį paprastesniu, patogesniu ir svarbiausia skaidresniu. Dėl to atsirado daugybė siūlymų ir potencialių sprendimų, kurie galėtu pakeisti tradicinį balsavimo procesą. Lyginant tradicinį balsavimą naudojant bilietus ir atvykstant į balsavimo centrą, elektroninis balsavimas atrodo daug patogesnis, greitesnis ir visiškai išsprendžianti žmogiškos klaidos faktorių skaičiuojant balsus (Liu & Wang, n.d.). Elektroninio balsavimo idėja yra leisti šalies piliečiams balsuoti internetu ir tai atlikti saugiai ir anonimiškai.

2.2.1 „I-voting“

Estija 2005 m. tapo pirmoji šalis pasaulyje kuri pristatė elektroninio balsavimo sistemą (“i-Voting — e-Estonia,” 2020). Estija pasaulyje gerai žinoma, kaip valstybė turinti elektroninę valdžią (*angl. E-goverment*), kurioje plačiai stengiamasi įdiegti technologijas į demokratiją ir į žmonių gyvenimą, taip norint užtikrinti piliečių duomenų saugumą ir valstybės organizacijų skaidrumą. Kiekvienais metais vis daugiau Estijos piliečių renkasi elektroninį balsavimą. Pirmame balsavime vykusiame 2005 m., net 1,9 % šalies gyventojų pasirinko balsavimą naudojant naują elektroninio balsavimo sistemą (Voting results in detail, 2019). Ir kiekvienais metais elektroninį balsavimą pasirenkančių ir sistema pasitikinčių piliečių skaičius auga. Nes 2019 m. balsavimo bilietus internetu panaudoju net 43,8 % šalies gyventojų (Voting results in detail, 2019).

„I-voting“ internetinio balsavimo sistema piliečių identifikavimui naudoja Estijos piliečių identifikavimo elektroninę kortelę, kuri yra išduodama kiekvienam Estijos gyventojui. Ji skirta gyventojų internetiniam identifikavimui ir prisijungimui prie valdžios sistemų, kurios prieinamos šalies gyventojams (Gibson, Krimmer, Teague, & Pomares, 2016).

„I-voting“ balsavimo sistema naudoja viešo rakto kriptografiją, kuri užtikrina balsavimo bilietų informacijos saugumą. Kiekvienas bilietas turi yra pasirašytas naudojant skaitmeninį parašą, kuris užtikrina bilieto unikalumą ir nurodo, kad bilietas priklauso būtent tam piliečiui. Kai balsavimo dalyvis pažymi savo balsą elektroniniame biliete, bilietas padalinamas į dvi dalis, šiame etape vykdomas duomenų nuasmeninimas, ir bilietas perduodamas į du fiziškai atskirus kompiuterius, kurių vienas išsaugo informaciją kas balsavo, o kitas skaičiuoja balsus. Toks sprendimas leidžia užtikrinti balsų anonimiškumą (Springall et al., 2014). Tačiau kaip ir kiekviena sistema, „I-voting“ yra pažeidžiama ir turi technologinių spragų, kurias kiekvienais metais „I-voting“ komanda bando spręsti ir ieško naujų sprendimų, kaip užtikrinti balsavimo saugumą. Estijos valdžia atsižvelgiant į visas potencialias atakas ir grėsmes nežada atsisakyti „I-voting“ sistemos ir siekia ją tobulinti (Springall et al., 2014).

2.2.2 Blokų grandinės technologijos taikymas elektroniniame balsavime

Atlikus blokų grandinės technologijos analizę galima matyti, kad blokų grandinės technologija yra puikiai tinkama decentralizuoti balsavimo procesą ir jį padaryti elektroniniu. Blokų grandinės technologija yra potencialus būdas išspręsti kylančias balsavimo problemas dėl saugumo, duomenų centralizavimo, balsų padirbinėjimo (Mentor, 2016). Šiai dienai yra daug straipsnių ir mokslinių darbų, kuriuose ekspertai ieško geriausio technologinio blokų grandinės sprendimų rinkinio, kuris užtikrintų visus keliamus saugumo reikalavimus

balsavimui. Tačiau šioje vietoje dėl blokų grandinės taikymo valstybės klausimų sprendimui balsavimo būdų, ekspertų nuomonės išsiskiria, vieni teigia, kad tai yra saugus būdas pateikiant kokius kriptografijos būdus galima būdų naudoti, o kiti pateikia argumentus kodėl dar šiandien nėra galutinio sprendimo, kuris būtų plačiai naudojamas spręsti šiam klausimui.

Kiekvienas blokų grandinės elektroninio balsavimo sprendimas turi atitikti šiuos kriterijus, kad būtų tinkamas viešam naudojimui (Curran, 2018; Liu & Wang, n.d.):

- Galimybė viešai patikrinti. Kiekvienas elektroninio balsavimo dalyvis gali patikrinti visą rinkimų procesą ir jo rezultatus;
- Kiekvienas balsavimo dalyvis gali patikrinti savo balsavimo procesą ir įsitikinti, kad jo balsas buvo užregistruotas ir pridėtas į galutinį balsų skaičių;
- Patikimumas. Kriptografiniai algoritmai ir blokų grandinės mechanizmai turi apsaugoti balsavimo nuo nesąžiningo elgesio ir galimų atakų;
- Nuoseklumas. Visi rinkimuose dalyvaujančių dalyvių balsų įrašai gaunami vykdant tapačia balsavimo tvarką ir tokiu pat būdų pridedami į bendrą rinkimų rezultatą;
- Visas balsavimo procesas, vykdomas naudojant blokų grandinės technologiją, turi būti audituojamas po rinkimų;
- Anonimiškumas. Balsavimo įrašai turi būti nuasmeninti ir niekas negali žinoti už ką balsavo rinkimų dalyvis, kaip tik jis pats;
- Skaidrumas. Visas balsavimo procesas yra atviras ir viešas, tai lemia didesnį teisingumą ir pagrįstumą.

Laikantis šių elektroninio balsavimo saugumo kriterijų dalinai galima laikyti balsavimo protokolą, pagrįstą blokų grandinės technologija, saugiu.

2.2.2.1 „Follow my vote“

Tai atviro kodo elektroninio balsavimo sistema, veikianti blokų grandinės technologijos pagrindu (“The Online Voting Platform of The Future - Follow My Vote,” 2020). Programinis sistemos kodas yra patalpintas „Github“ kodo talpykloje ir viešai prieinamas visiems. Toks sprendimas leidžia į vystymo procesą įtraukti daugiau žmonių ir bendrai dalintis idėjomis vystant „Follow My Vote“ elektroninio balsavimo sistemą.

Pagrindinis sistemos principas yra – vienas žmogus, vienas balsas (*angl. One Person, One Vote Voting*) (“The Online Voting Platform of The Future - Follow My Vote,” 2020), šiuo principu yra vykdomi politiniai rinkimai. Kitas principas, kuriuo yra paremta sistema tai –

svertinis balsavimas (*angl. Stake Weighted Voting*). Šio principo idėja yra ta, kad ne visų balsuojančių balso vertė yra vienodą. Tinkamas pavyzdys šiam principui galėtu būti balsavimas įmonėje, kurioje balsuoja įvairių pareigų darbuotojai, tai pagal šį principą balsų vertė yra nustatoma pagal jų pareigas. Toks balsavimo principas yra naudojamas Europos Taryboje ("The Online Voting Platform of The Future - Follow My Vote," 2020).

Norint pradėti naudotis sistema, ją reikia įsidiegti savo kompiuteryje, telefone arba planšetėje. Vartotojų identifikavimas vykdomas nurodant savo asmens dokumentą, tai gali būti asmens tapatybės kortelė, pasas arba vairuotojo teisės. Sėkmingai identifikuotas vartotojas gali gauti balsavimo bilietą ir jame pažymėti savo balsą. Užpildytas ir patvirtintas bilietas pridedamas, kaip naujas blokas blokų grandinėje. „Follow My Vote" taip pat leidžia pakeisti savo balsą dar nepasibaigus balsavimo laikui ir pasibaigus balsavimui užskaitomas tik paskutinis balsavusio bilieto rezultatas. Pasibaigus balsavimui kiekvienas vartotojas gali įsitikinti, kad jų balsas buvo priimtas ir pridėtas prie galutinio rinkimų rezultato (Mentor, 2016).

2.2.2.2 „Agora Vote"

„Agora Vote" tai elektroninio balsavimo sistema paremta blokų grandinės technologija. „Agora" yra 2015 m. Šveicarijoje įsikūrusi balsavimo technologijų įmonė kurianti patikimus balsavimo sprendimus vyriausybinėms institucijoms ("Agora Vote," 2017). „Agora Vote" elektroninio balsavimo sistemos pagrindiniai principai:

- Skaidrumas – kiekvienas rinkimų proceso žingsnis turi būti lengvai suprantamas balsuojančiam ir balsavimo organizatoriui;
- Privatumas – kiekvieno balsuojančio pasirinkimas turi būti niekam nežinomas;
- Sąžiningumas – tik balsavimo teisę turintys dalyviai gali balsuoti;
- Kaina – rinkimų procesas turi būti įperkamas vyriausybėms ir jos piliečiams;
- Prieinamumas – visi rinkėjai neatsižvelgiant į vietą, socialinį statusą, negalią, turi turėti galimybę atiduoti savo balsą.

„Agora Vote" susideda iš penkių technologinių sluoksnių ("Agora Vote," 2017):

1. Skelbimų lentos (*angl. Bulletin Board*) blokų grandinė;
2. „Catena";
3. Bitkoino blokų grandinė;
4. „Valeda" tinklas;
5. „Votapp".

Visi šie sluoksniai reikalingi tam, kad naudojant kriptografiją užtikrinti saugų balsavimo procesą. Aukščiau išvardinti sluoksniai bendrauja tarpusavyje įvairiose rinkimų proceso vietose. Pirmasis sluoksnis skelbimų lentos blokų grandinės tinklas yra sukurtas „Agora“ komandos, šiame tinkle valdomi ir vykdomi „Agora“ ir trečiųjų šalių procesai, kurie yra įtraukti į baltą sąrašą (*angl. White-List*). Šis sluoksnis atsakingas už vartotojų identifikavimą, naudojant trečiųjų šalių sistemas, gavus teigiamą atsakymą iš kitų sistemų apie vartotoją duomenys yra išsaugoti.

Antrasis sluoksnis sistemoje yra „Catena“, tai klastojimui atsparus registravimo mechanizmas pagrįstas bitkoino protokolu. Sluoksnis sujungia skelbimų lentą ir palaikančius kriptografinius sprendimus blokų grandinės sprendimu, kuris suteikia decentralizuotą duomenų nekintamumą.

Trečiasis bitkoino sluoksnis yra skaitmeninė decentralizuota sistema, kurioje saugomi visi sandoriai, kurie ir vyksta per bitkoino tarpusavio tinklą. Pagrindinė idėja šios technologijos yra ta, kad ji leidžia balsavimo dalyviams saugoti ir perduoti duomenis internetu neturint centralizuotos sistemos.

Ketvirtasis sluoksnis – „Valeda“ tinklas. Tai „Agora“ komandos sukurtas decentralizuotas tinklas, sudarytas iš nepatikimų mazgų, kurių tikslas yra patvirtinti balsavimo rezultatus skelbimų lentoje. Sistemos sluoksnio užduotis įrodyti, kad pirmajame sluoksnyje surinkti duomenys apie vartotoją ir apie jo balsą yra galiojantys.

Paskutinis sluoksnis šioje sistemoje yra „Voteapp“. Šis sistemos sluoksnis leidžia visiems sukurti savo programas, kurios komunikuotu „Agora“ tinkle. Tai yra padaryta tam, kad vartotojo sąsaja būtų kuriami atskirai nuo pačios sistemos, todėl tai buvo iškelta į aukščiausią sluoksnį. Pagrindinės programos esančios „Voteapp“ sluoksnyje yra „Voting Booth“, „Audit“ ir „Node“.

Visi šie išvardinti sluoksniai padeda užtikrinti sistemos saugumą, o sistemos padalijimas į sluoksnius leidžia atitinkamas operacijas su duomenis vykdyti atitinkamuose sluoksniuose, kurie duomenis atskiria. Kiti sluoksniai reikalingi patvirtinti duomenų teisingumą ir vientisumą.

2.2.2.3 „Polys“

„Polys“ tai „Kaspersky Innovation Hub“ komandos 2020 m. pristatytas elektroninio balsavimo sprendimas paremtas blokų grandinės technologija. Pagrindinis skirtumas šio sprendimo lyginant su kitais, kad jis yra padalintas į dvi balsavimo galimybes, nes „Polys“ yra

ne tik balsavimo sistema, bet ir techninė įranga skirta fiziniams balsavimams atlikti. Todėl vykdant balsavimus naudojant „Polys“ yra du būdai, kaip galima dalyvauti balsavime, naudojantis:

1. „Polys“ balsavimo mašina;
2. Išmaniuoju įrenginiu.

Šio sprendimo privalumas yra tas, kad abu balsavimo būdai yra susieti vienas su kitu ir visi rezultatai yra talpinami viename ir tame pačiame blokų grandinės tinkle.

Kitas unikalus architektūrinis sprendimas yra atskirti balsavimo ir balsų skaičiavimo etapai, kurie tarpusavyje yra niekuo nesusiję, tačiau iš balsavimo proceso į balsų skaičiavimą yra perduodami rezultatai (*Polys-the digital voting system*, n.d.). Toks sprendimas padeda užtikrinti balsavimo etapų tarpusavio nepriklausomybę ir duomenų saugumą.

Tačiau šiai dienai nėra pristatytas galutinis produktas ir detali informacija apie naudojamas technologijas ir pritaikytus sprendimus, produktas dar yra kūrimo stadijoje ir patys, kūrėjai nėra paviešinę jokių kodo pavyzdžių.

2.3 Egzistuojančių elektroninio balsavimo sprendimų palyginimas

Egzistuojančių elektroninio balsavimo sprendimų paremtų blokų grandinės technologija palyginimas pateikiamas lentelėje, kurioje atsispindi esminiai sprendimų skirtumai ir trūkumai.

2 lentelė. Egzistuojančių elektroninio balsavimo sistemų palyginimas

NR.	Sistema / Kriterijus	„I-voting“	„Follow my vote“	„Agora Vote“	„Polys“
1.	**Sukūrimo data**	2005 m.	2016 m.	2015 m.	2020 m. (pristatyta)
2.	**Sukūrimo šalis**	Estija	JAV	Šveicarija	Rusija
3.	**Veikimo šalis**	Estija	JAV	Visos	-
4.	**Naudojama blokų grandinės technologija**	-	+	+	+
5.	**Išmaniosios sutartys**	-	-	-	-
6.	**Duomenų saugojimo tipas**	Centralizuotas	Decentralizuotas	Decentralizuotas	Decentralizuotas
7.	**Asmenų identifikavimo būdas**	Estijos piliečio asmens ID	Nuosavas vartotojų identifikavimas	Nuosavas vartotojų identifikavimas	Nuosavas vartotojų identifikavimas

Remiantis atliktos, egzistuojančių elektroninių balsavimo sprendimų, analizės rezultatais, nei vienas iš egzistuojančių sprendimų nepalaiko Lietuvos piliečių identifikavimo būdo, kadangi naudojami nuosavi vartotojų identifikavimo sprendimai neužtikrina tinkamo asmens duomenų saugumo ir jų aktualumo ir tokie duomenys gali būti falsifikuojami. Taip pat visi ankščiau aprašyti sprendimai nenaudoja išmaniųjų sutarčių technologijos, kuri leidžia

lengvai ir automatiškai vykdyti susitarimus remiantis juose aprašytomis vykdymo sąlygomis, kas leidžia užtikrinti balsavimo proceso saugumą.

2.4 Skyriaus išvados

1. Atlikta blokų grandinės technologijos analizė parodo, kad blokų grandinės technologija leidžia užtikrinti duomenų vientisumą, balsavimo proceso skaidrumą. Yra ir trūkumai į kuriuos reikia atsižvelgti tai yra duomenų saugumas nuasmeninant, juos talpinant į blokų grandinę ir galimybė įsitikinti, kad balsas buvo įtrauktas į galutinį balsavimo rezultatą.
2. Atlikus egzistuojančių elektroninio balsavimo sprendimų analizę, egzistuojantys sprendimai nėra tinkami dėl tam tikrų technologinių ir šalių veikimo apribojimų, tokių kaip nėra galimybės identifikuoti Lietuvos piliečių tapatybės prieš balsavimą, nėra naudojamos išmaniosios sutartys, kurios padeda užtikrinti automatinį balsavimo proceso vykdymą ir jo saugumą. Egzistuojantys elektroninio balsavimo sprendimai tokie, kaip skyriuje apžvelgti „Follow My Vote“ ir „Agora Vote“ nėra galutinai tinkami plačiajam naudojimui, o „Polys“ yra dar tik pradinėje kūrimo stadijoje.
3. Apsižvelgus į egzistuojančius Lietuvos reikalavimus elektroninio balsavimo procesui kompiuterizuoti buvo nuspręsta sukurti naują sistemą paremtą „Ethereum“ blokų grandinės protokolu naudojant išmaniąsias sutartis balsavimo procesui vykdyti ir su „Elektroniniai valdžios vartai“ sistemos integracija Lietuvos piliečių turinčių balsavimo teisę identifikavimui.

2.5 Siūlomas elektroninio balsavimo sprendimas

Šiame skyriuje pateikiamas elektroninio balsavimo kompiuterizavimo problemos sprendimas, kurio pagrindas yra egzistuojančio blokų grandinės protokolo – „Ethereum“ technologijos panaudojimas, kuriant decentralizuotą elektrinio balsavimo sistemą. „Ethereum“ išmaniųjų sutarčių pagrindu bus kuriama elektroninio balsavimo sistema, kuris bus vykdoma „Ethereum“ blokų grandinės tinkle.

Siūlomo sprendimo tikslas užtikrinti vartotojų anonimiškumą, duomenų saugumą, bei tuo pačiu įgyvendinti duomenų patikrinamumą, kad užtikrinti balsavimo proceso skaidrumą. Toliau aprašomas sprendimas pasirinktas, nes tai padeda decentralizuoti sistemą, todėl tai leidžia atsiriboti nuo centralizuotų saugyklų, serverių ir išmaniuosius kontraktus vykdyti kiekviename iš blokų grandinės tinkle esančių gijų. Išmaniosios sutartys užtikrins, kad tik įgyvendinus jame aprašytas balsavimo sąlygas balsas bus įtrauktas į galutinį rezultatą ir darbo patikrinimo koncepcija leis įsitikinti, kad balsavimo bilietai (šiuo atveju vienas balsavimo bilietas yra vienas grandinės blokas), nebus padirbami arba sugadinami vykstant balsavimo procesui ir jam pasibaigus.

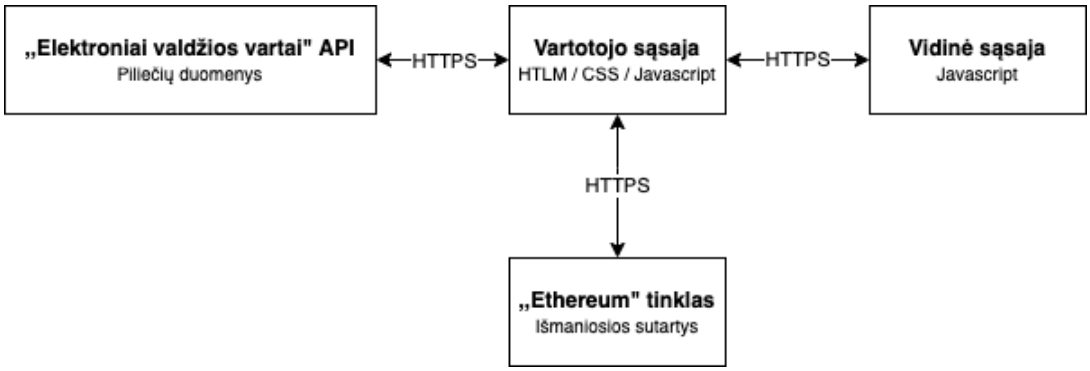
2.5.1 Elektroninio balsavimo sistemos architektūra

Toliau yra aprašoma siūlomos elektroninio balsavimo sistemos architektūra, kurioje atsispindi, su kokiomis išorinėmis sistemomis bus vykdoma komunikacija ir kokiais tikslais. Taip pat sistemoje reikalingas vartotojų autentifikavimas, kurio metu vartotojai naudojant „Elektroninės valdžios vartų“ sistemą yra identifikuojami ir nustatoma jų balsavimo teisė, sėkmės atveju vartotojai gauna unikalių sesijos raktą, kuris bus naudojamas jungiantis prie privataus blokų grandinės tinklo.

Elektroninio balsavimo sistemos paremtos „Ethereum“ blokų grandinės technologija architektūra susideda iš keturių pagrindinių sistemos dalių:

1. Vartotojo sąsaja (*angl. Front-end*) – tai vartotojui matoma dalis, kurioje bus atvaizduojami vykstantys balsavimai, jų rezultatai bei galimybė prisijungus balsuoti;
2. Vidinė sąsaja (*angl. Back-end*) – tai posistemė atsakinga už balsavimo administracijos autorizavimą ir jų funkcijų vykdymą;
3. „Ethereum“ tinklas – tai „Ethereum“ blokų grandinės vidinis tinklas, kuriame bus vykdomos išmaniosios sutartys skirtos balsavimo procesui įgyvendinti;

4. „Elektroniniai valdžios vartai“ aplikacijos programavimo sąsaja (*angl. Application Programming Interface – API*) – tai papildomos sistemos integracija, skirta Lietuvos piliečių tapatybės identifikavimui.



8 pav. Abstrakti elektroninio balsavimo sistemos architektūra

Komunikacija tarp šių dviejų sistemos dalių bus vykdomai įprastomis hipertekstų siuntimo protokolas saityno duomenims siųsti (*angl. Hypertext Transfer Protocol – HTTPS*) užklausomis naudojant transporto lygmens (*angl. Transfer layer security protocol – TLS*) protokolą, informaciją perduodant „Javascript“ objektų notacijos (*angl. „Javascript Object Notation – JSON*) formatu.

Toliau pateikiami pagrindiniai siūlomos elektroninio balsavimo sistemos komponentai:

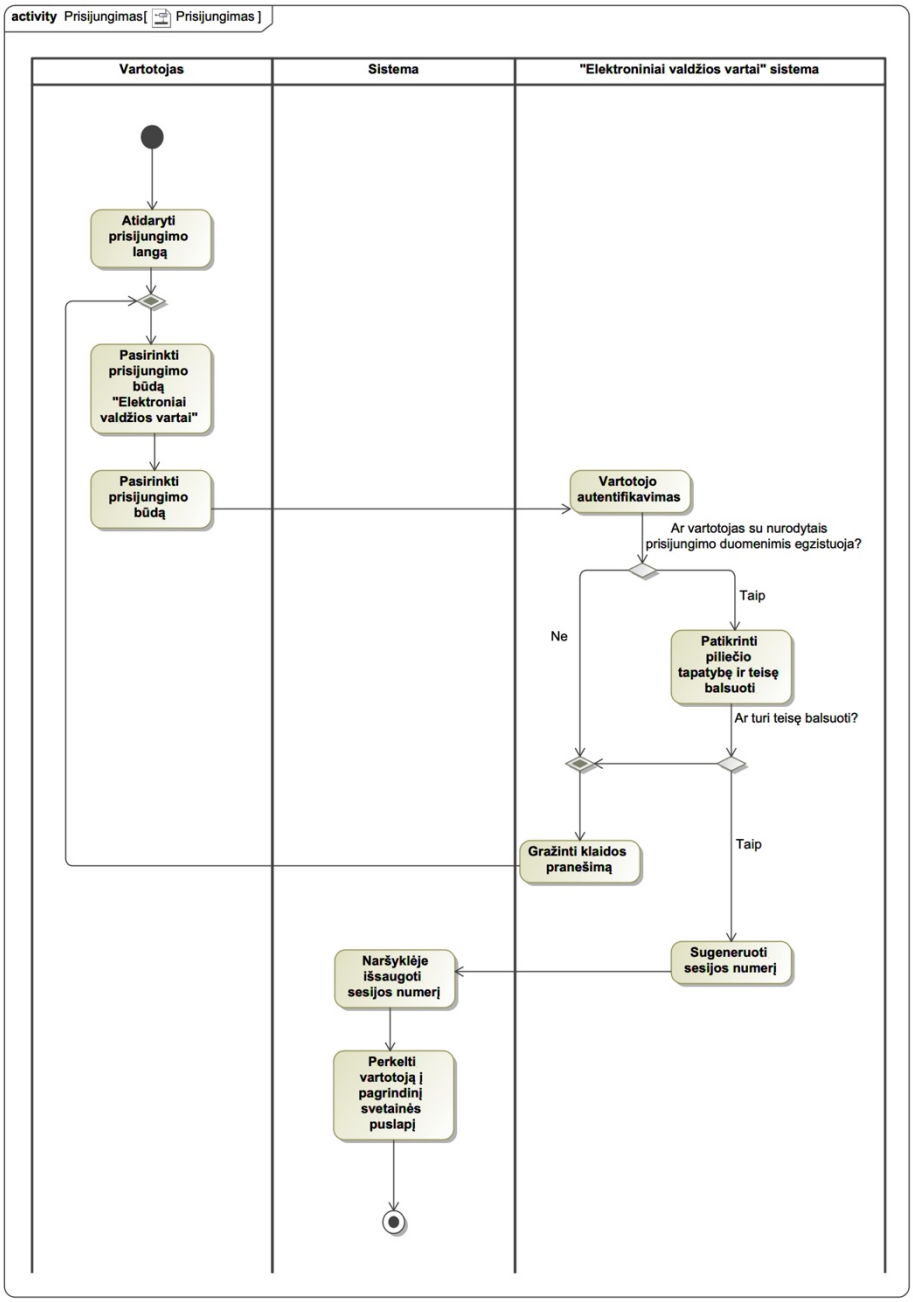
1. Rinkėjas – tai vartotojas, kuris yra identifikavęs save naudojant „Elektroniniai valdžios vartai“ sistemą ir iš jos gavęs unikalų sesijos numerį, taip pat Lietuvos Respublikos pilietis turintis teisę dalyvauti rinkimuose;
2. Asmens tapatybės nustatymo serveris („Elektroniniai valdžios vartai“ programos sąsaja) – ši sistemos dalis atsakinga už vartotojų identifikavimą ir unikalių sesijos raktų generavimą, bei šių raktų perdavimą rinkėjui atgal sėkmingo identifikavimo atveju ir perduodant jį į privatų elektroninio balsavimo tinklą;
3. Balsavimo sistema – tai žiniatinklio svetainė, kurioje vartotojai sėkmingai prisijungę, gali balsuoti, kaip galima matyti ankščiau pateiktoje diagramoje informacija yra perduodama į „Ethereum“ tinklą, kurioje yra atliekami sekantys verifikavimo ir naujo bloko bei transakcijos kūrimo etapai;
4. Išmanioji sutartis – yra dinamiška ir suteikia rinkėjams galimybę peržiūrėti savo balsavimo bilietus ir įsitikinti, kad balsas buvo įskaičiuotas. Tokia galimybė padidina vartotojų pasitikėjimą rinkimais.

Šie komponentai bendroje visumoje sudaro bendrą elektroninio balsavimo sistemą paremta „Ethereum“ blokų grandinės technologija naudojant išmaniąsias sutartis.

2.5.2 Sistemos dinaminis vaizdas

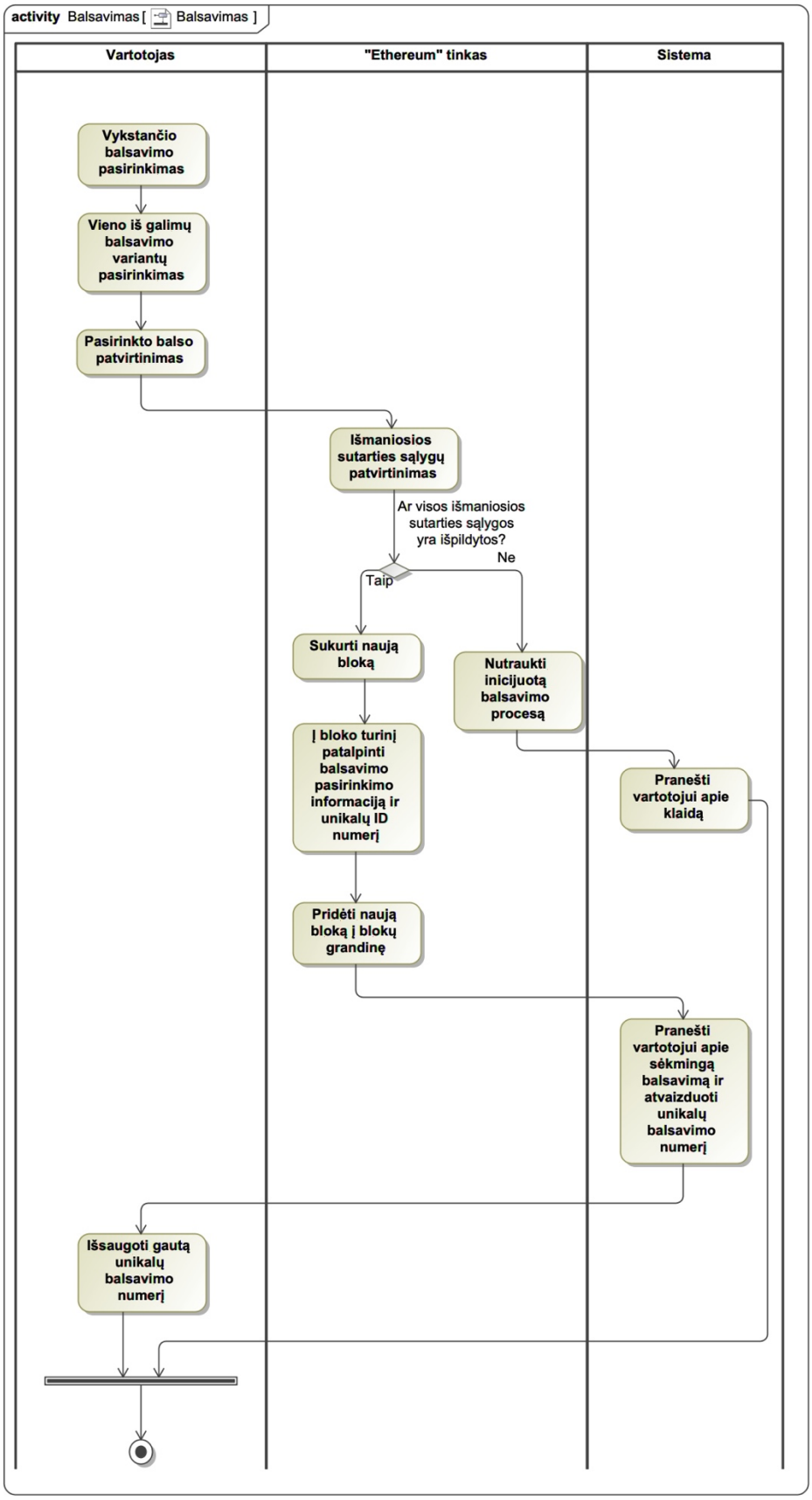
Šiame poskyryje pateikiamas detali siūlomo metodo kiekvieno žingsnio veiklos diagrama, kuriose atsispindi esminis kuriamos sistemos veikimo principas ir vykdomi procesai tokie kaip: prisijungimas prie sistemos naudojantis "Elektroninės valdžios vartų" sistema, balsavimas ir balso patikrinimas.

Prisijungimas naudojantis „Elektroninės valdžios vartų“ sistema.



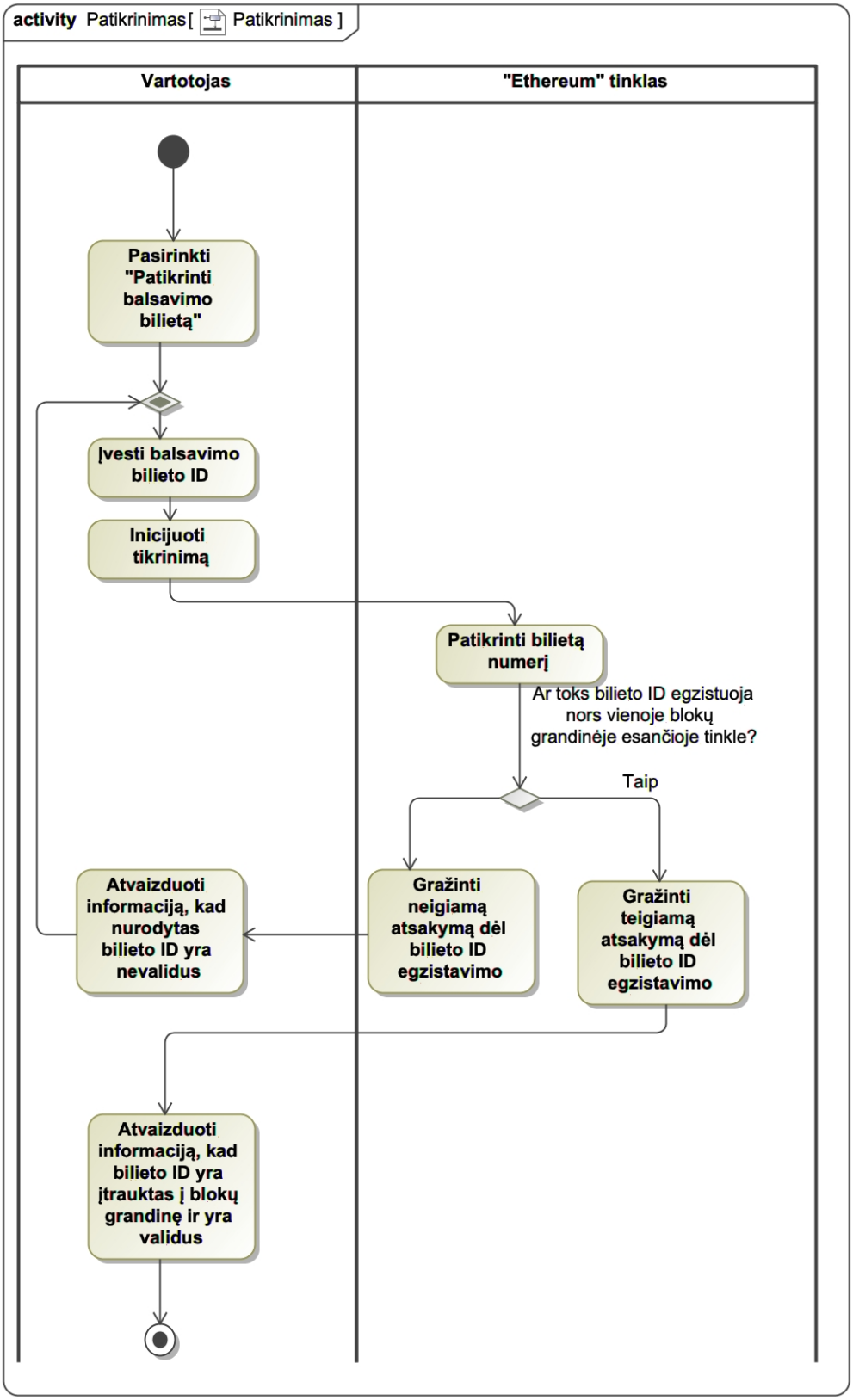
9 pav. Prisijungimo prie sistemos naudojantis „Elektroniniai valdžios vartai" sistema veiklos diagrama

Balsavimo procesas.



10 pav. Balsavimo proceso veiklos diagrama

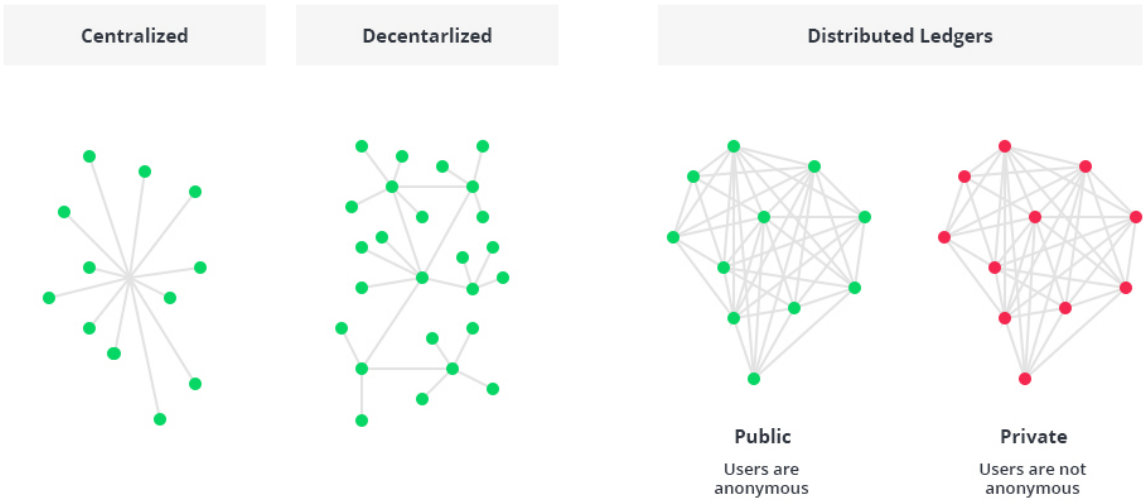
Balso patikrinimo procesas.



11 pav. Balso patikrinimo proceso veiklos diagrama

2.5.3 Elektroninio balsavimo tinklo architektūra

Kadangi blokų grandinės technologijos pagrindinė idėja yra visos sistemos decentralizavimas ir jos paskirstymas, todėl kuriant naują elektroninio balsavimo sprendimą būtina atsižvelgti į šį tinklo unikalumą. Kadangi centralizuotos sistemos architektūra nėra tinkama elektroninio balsavimo skaidrumo problemai spręsti, nes duomenys yra saugomi tik vienoje centralizuotoje duomenų bazėje, apdorojami taip pat centriniame serveryje, dėl to atsiranda saugumo problema. Šiai problemai spręsti tinkamesnis sprendimas yra decentralizuota paskirstyta architektūra, kurioje nėra tradicinio centrinio serverio, o tinkle esantys mazgai komunikuoja tarpusavyje. Tokia architektūra yra vadinama – paskirstytųjų duomenų technologija (*angl. Distributed Ledgers Technology*), dėl šios architektūros visi mazgai yra sujungti vienas su kitu ir gali tarpusavyje komunikuoti, t. y. vykdyti transakcijas ir jas verifikuoti naudojant darbo įrodymo technologiją, kurios pagrindu ir yra užtikrinimas saugumas blokų grandinės tinkle.



12 pav. Tinklo architektūrų diagramos (Blockchain Architecture Explained: How It Works & How to Build, n.d.)

Tokia architektūra užtikrina, kad viena iš mazgų sugedus komunikacija nutraukiam tik su vienu mazgu, tačiau su kitais ji išlieka, kas yra priešingai su centralizuota architektūra, kur už komunikaciją atsako centrinis mazgas.

Kadangi blokų grandinės tinklas yra viešai prieinamas visiems, toks sprendimas elektroninio balsavimo sistemai nėra visiškai tinkamas sprendimas. Pirmoji priežastis yra ta, kad „Ethereum“ transakcijas galima inicijuoti vartotojams tik turintiems „Ether“ kriptovaliutos, kuri gali būti gaunama viešame blokų grandinės tinkle iš kitų paskyrų atlikus

transakciją arba vykdant kasybą ir gaunant apdovanojimas už sėkmingą darbą. Norint įgyvendinti elektroninį balsavimą, kompanijos arba šalies ribose reikalingas privatus blokų grandinės tinklas, kuris balsavimo kūrėjams leis, autorizuoti balsuojančius piliečius, jiems išskirti žetonus balsavimui t. y. suteikti galimybę balsuoti arba pakeisti savo balsą eigoje. Privačiame blokų grandinės tinkle esanti kriptovaliuta irgi yra izoliuota jame ir negali būti naudojama viešajame „Ethereum“ tinkle.

Blokų grandinės tinklo izoliavimas ir sumažinimas leis padidinti transakcijų greitį ir jų patikimumą. Tačiau kuo mažiau mazgų yra blokų grandinės tinkle tuo silpnesnis jis tampa, dėl šios priežasties privatus blokų grandinės tinklas turi būti kuriamas tuo atveju jeigu jame bus daugiau negu 1000 mazgų, kurie kartu galės vykdyti darbo įrodymo procesą, kad užtikrinti blokų grandinės vientisumą. Sekantis privalumas privataus blokų grandinės tinklo yra tas, kad jame esantys mazgai yra apribojami tam tikromis taisyklėmis, kurias nustato tinklo savininkas/organizacija, elektroninio balsavimo atveju tai padės užtikrinti tinkamą balsavimo duomenų srautą.

2.5.4 Siūlomos elektroninio balsavimo sprendimo įgyvendinimo technologijos

Elektroninio balsavimo išmaniosioms sutartis programuoti siūloma „Solidity“ aukšto lygio programavimo kalba. Šiuo metu tai pati populiariausia išmaniųjų sutarčių kūrimo programavimo kalba, kuri lengvai leidžia kurti sutartis naudojant objektiškai orientuoto programavimo paradigmą. Privataus „Ethereum“ blokų grandinės tinklo kūrimui pasirinktas „Ganache“ blokų grandinės tinklo serveris, kuris leidžia susikurti virtualų blokų grandinės tinklą. Išmaniųjų sutarčių testavimui pasirinktas „Ethereum“ komandos sukurtas „Remix“ įrankis, skirtas testuoti išmaniąsias sutartis, nekeliant jų į viešą, globalų blokų grandinės tinklą, o naudojant „Ganache“ privatų tinklą. „Remix“ įrankis leidžia lengvai vykdyti išmaniąsias sutartis, jas analizuoti ir aptikti neatitikimus jų taisyklėse prieš pradedant juos vykdyti tikrame tinkle.

2.6 Skyriaus apibendrinimas ir rezultatai

Šiame skyriuje pateiktas siūlomas problemos sprendimas, giliau supažindinta su „Ethereum“ blokų grandinės protokolu ir išmaniosiomis sutartimis. Pateikta sistemos architektūra bei jos komponentai. Taip pat pasirinktas galutinis sprendimas dėl blokų grandinės tinklo architektūros ir technologijų rinkinys reikalingas sutartims sukurti ir ištestuoti.

2.7 Skyriaus išvados

Išanalizavus „Ethereum“ blokų grandinės protokolą ir radus tinkamą sprendimą esančiai problemai spręsti, pasirinktas „Ethereum“ blokų grandinės privatus tinklas ir išmaniosios sutartys įgyvendinti balsavimo logikai, bei pasirinkta „Elektroniniai valdžios vartai“ programavimo sąsaja, reikalinga Lietuvos Respublikos piliečių identifikavimui, o duomenų nuasmeninimo proceso pagalba padaryti sistemą saugią visiems balsuojantiems. Toks sprendimas leis užtikrinti balsavimo proceso saugumą, skaidrumą, blokų grandinės vientisumą ir suteiks galimybę įsitikinti, kad balsas tikrai buvo įtrauktas į galutinį rezultatą.

3 Elektroninio balsavimo sistemos prototipo realizacija ir testavimas

3.1 Sistemos reikalavimų specifikacija

3.1.1 Sistemos naudotojai

Sistemoje egzistuoja du vaidmenys, kurie abu yra vienodai svarbūs sistemoje, nes atlieka pagrindines sistemos funkcijas, kurios reikalingos sistemos gyvavimui:

- Administratorius;
- Rinkėjas.

Žemiau lentelėse pateikiamas detalus kiekvieno vaidmens aprašymas.

3 lentelė. Naudotojo „Savininkas“ detalizavimas

Vaidmuo	Administratorius
Funkcijos	Naudotojas yra atsakingas už sukurtą balsavimą, turi visas teises sustabdyti arba nutraukti balsavimą, bei suteikti balsuojantiems teisę balsuoti.
Svarba	Šio vaidmens naudotojas sukuria naujus balsavimus, kuriuose vėliau balsuoja piliečiai turintys teisę balsuoti.

4 lentelė. Naudotojo „Balsuotojas" detalizavimas

Vaidmuo	Rinkėjas
Funkcijos	Elektroninės valdžios vartų identifikuotas naudotojas tampa balsuotoju, kurio pagrindinės funkcijos yra: balsuoti, įsitikinti, kad balsas buvo užregistruotas ir peržiūrėti balsavimo rezultatus.
Svarba	Šio vaidmens naudotojas tęsia balsavimo procesą, išreikšdamas savo nuomonę balsavime.

3.1.2 Funkciniai reikalavimai

Šiame poskyryje lentelėse pateikiami sistemos funkciniai reikalavimai, kurie privalo būti įgyvendinti kuriant sistemą ir norint užtikrinti, kad sistema funkcionuotų tinkamai t. y. atitinka toliau pateikiamą reikalavimų specifikaciją.

Vartotojo autorizavimas: Sistemoje turi būti galimybė visų vaidmenų vartotojams prisijungti.

5 lentelė. Funkcinis reikalavimas „Vartotojo prisijungimas"

Reikalavimo Nr.	FR-1
Pagrindimas	Prisijungęs vartotojas gali naudotis sistemos funkcionalumu, kuris yra apribotas pagal naudotojo vaidmenį.
Aktoriai	Visi
Tikimo kriterijus	Naudotojas prisijungia prie sistemos naudojant „Elektroniniai valdžios vartai“ programos sąsają.
Prieš sąlyga	-

Balsuoti: Sistemoje turi būti galimybė naudotojui balsuojant išreikšti savo nuomonę atitinkamu klausimu.

6 lentelė. Funkcinis reikalavimas „Balsuoti"

Reikalavimo Nr.	FR-2
Pagrindimas	Pagrindinis sistemos procesas.
Aktoriai	Rinkėjas
Tikimo kriterijus	Naudotojas pasirinkęs balsuoti už atitinkamą klausimą, savo balsą gali užregistruoti sistemoje.
Prieš sąlyga	Naudotojas turi būti autorizuotas naudojant „Elektroniniai valdžios vartai“ programos sąsają.

Peržiūrėti vykstančius balsavimus: Sistemoje turi būti galimybė peržiūrėti dabar vykstančius balsavimus, pateikiant jų pagrindinę informaciją ir galimus klausimo sprendimo pasirinkimus.

7 lentelė. Funkcinis reikalavimas „Peržiūrėti vykstančius balsavimus“

Reikalavimo Nr.	FR-3
Pagrindimas	Visi aktyvūs balsavimai turi būti matomi visiems sistemos naudotojams ir lankytojams, kadangi balsavimas yra viešas procesas ir jame sprendžiami klausimai viešai žinomi.
Aktoriai	Visi
Tikimo kriterijus	Naudotojas sistemoje mato visus aktyvius balsavimus, bei jų pagrindinę informaciją (sprendžiamas klausimas, galimi pasirinkimo variantai, pradžios data ir laikas, pabaigos data ir laikas, balsavimo savininkas).
Prieš sąlyga	-

Peržiūrėti pasibaigusius balsavimus: Sistemoje turi būti galimybė peržiūrėti jau pasibaigusius balsavimus, pateikiant jų pagrindinę informaciją, galimus klausimo sprendimo pasirinkimus ir galutinius balsavimo rezultatus.

8 lentelė. Peržiūrėti pasibaigusius balsavimus

Reikalavimo Nr.	FR-4
Pagrindimas	Pasibaigus balsavimui naudotojams yra aktualu sužinoti, kokiu rezultatu tam tikras balsavimas pasibaigė, todėl informacija apie pasibaigusį balsavimą turi būti prieiname vartotojams.
Aktoriai	Visi
Tikimo kriterijus	Naudotojas sistemoje mato visus pasibaigusius balsavimus, bei jų pagrindinę informaciją (sprendžiamas klausimas, galimi pasirinkimo variantai, pradžios data ir laikas, pabaigos data ir laikas, balsavimo savininkas ir galutinis rezultatas).
Prieš sąlyga	Yra pasibaigęs nors vienas balsavimas

Sukurti balsavimą: Sistemoje turi būti galimybė organizatoriui sukurti naują balsavimą, kuriame galėtų dalyvauti prisijungę, balsavimo teisę turintys sistemos naudotojai.

9 lentelė. Funkcinis reikalavimas „Sukurti balsavimą“

Reikalavimo Nr.	FR-5
Pagrindimas	Rinkimų procesas prasideda nuo pirmo žingsnio t. y. balsavimo iniciavimo, kuriame yra nurodomas sprendžiamas klausimas ir galimi jo sprendimo variantai.
Aktoriai	Administratorius
Tikimo kriterijus	Administratorius gali sukurti naują balsavimą.
Prieš sąlyga	Vartotojas prisijungęs prie sistemos kaip administratorius.

Pradėti balsavimą: Sistemoje turi būti galimybė neaktyvų tačiau sukurtą balsavimą padaryti aktyviu ir matomu rinkėjams.

10 lentelė. Funkcinis reikalavimas „Pradėti balsavimą“

Reikalavimo Nr.	FR-6
Pagrindimas	Sukurtas naujas balsavimas nėra iškarto matomas naudotojams sistemoje, kad per klaidą per anksti nebūtų paviešinta atitinkama informacija arba galima būtų balsavimą paruošti dar jam neprasidėjus.
Aktoriai	Administratorius
Tikimo kriterijus	Naudotojas gali padaryti neaktyvų balsavimo įrašą aktyviu.
Prieš sąlyga	Sukurtas neaktyvus balsavimas.

Sustabdyti balsavimą: Sistemoje turi būti galimybė sustabdyti vykstantį balsavimą.

11 lentelė. Funkcinis reikalavimas „Sustabdyti balsavimą“

Reikalavimo Nr.	FR-7
Pagrindimas	Vykstant balsavimui gali atsirasti būtinybė sustabdyti balsavimą, kad naudotojai negalėtų balsuoti, tai reikalinga balsavimo metu atsiradus priežasčiai reikalaujančiai balsavimo sustabdymo.
Aktoriai	Administratorius
Tikimo kriterijus	Naudotojas gali padaryti aktyvų balsavimą nebeaktyviu.
Prieš sąlyga	Sukurtas balsavimas ir jis yra aktyvus.

Įsitikinti, kad balsas yra įtrauktas: Sistemoje turi būti galimybė naudotojui įsitikinti, kad jo balsas buvo įtrauktas į bendra balsavimo rezultatą.

12 lentelė. Funkcinis reikalavimas „Įsitikinti, kad balsas yra įtrauktas“

Reikalavimo Nr.	FR-8
Pagrindimas	Kad užtikrinti balsavimo proceso skaidrumą naudotojams turi būti suteikta galimybė įsitikinti, kad balsas tikrai buvo įtrauktas į balsavimo rezultatą.
Aktoriai	Rinkėjas
Tikimo kriterijus	Naudotojas įvedęs unikalų balsavimo numerį gauna informaciją apie šio balsavimo bilieto statusą.
Prieš sąlyga	Išduotas balsavimo bilietas

3.1.3 Nefunkciniai reikalavimai

Šiame poskyryje lentelėse pateikiami sistemos nefunkciniai reikalavimai, kurie privalo būti įgyvendinti kuriant naują sistemą, kad užtikrinti sistemos saugumą, panaudojimą ir suderinamumą su technologijomis.

3.1.3.1 Saugumo reikalavimai

Saugumo sertifikatai: Serveris, kuriame bus patalpinta sistema turi naudoti „TLS“ saugumo sertifikatą.

13 lentelė. Nefunkcinis saugumo reikalavimas „Saugumo sertifikatas“

Reikalavimo Nr.	NSR-1
Pagrindimas	Duomenys perduodami iš serverio per kanalą turi būti koduojami, kad jų negalima būtų perėmus perskaityti.
Tikimo kriterijus	Serveris naudoja „TLS" saugumo sertifikatą.

Duomenų nuasmeninimas: Duomenys privalo būti nuasmeninami prieš siunčiant juos į blokų grandinės tinklą.

14 lentelė. Nefunkcinis saugumo reikalavimas „Duomenų nuasmeninimas“

Reikalavimo Nr.	NSR-2
Pagrindimas	Duomenys blokų grandinės tinkle turi būti nuasmeninti, kad tokiu atveju jeigu kažkam jas pavyktų perskaityti, negalėtų nustatyti kam priklauso tam tikras balsas, tai labai svarbu, kadangi blokų grandinės yra saugomas kiekviename iš naudotojų (tinkle naudotojai suprantami, kaip mazgai) įrenginiuose pagal paskirstytos sistemos principą.
Tikimo kriterijus	Bloke saugomame blokų grandinės tinkle nėra jokios asmeninės informacijos, kuri leistu nustatyti rinkėją.

Administracijos naudotojų sesijų valdymas: Administracijos naudotojų sesija turi būti valdoma naudojant JSON internetinių aplikacijų prieigos raktas (*angl. JSON Web Token – JWT*).

15 lentelė. Nefunkcinis saugumo reikalavimas „Administracijos naudotojų sesijų valdymas“

Reikalavimo Nr.	NSR-3
Pagrindimas	Administracijos naudotojai yra identifikuojami naudojant privatų serverį ir duomenų bazę, kurioje yra užregistruojami tik administracinėms funkcijoms atlikti skirti naudotojai ir jų

	užklausoms vykdyti turi būti naudojamas sesijos valdymo mechanizmas.
Tikimo kriterijus	Prisijungus prie sistemos kaip naudotojas „Administratorius“ vidinis serveris sugeneruoja prieigos raktą, kuris yra saugomas tik naudotojo naršyklėje ir galio 2 valandas.

Išmaniųjų sutarčių testavimas: Išmaniųjų sutarčių programinis kodas turi būti padengtas 100% komponentų testais.

16 lentelė. Nefunkcinis saugumo reikalavimas „Išmaniųjų sutarčių testavimas“

Reikalavimo Nr.	NSR-4
Pagrindimas	Išmaniosios sutartys yra kritinis sistemos komponentas, kuriame yra aprašyta visa automatinė balsavimo vykdymo logika ir tokia kritinė dalis turi būti ištestuota maksimaliai.
Tikimo kriterijus	Išmaniosios sutartys turi joms parašytus testus pagal dokumentacijoje pateiktus testavimo scenarijus ir komponentų testavimo rezultatas turi būti 100%.

Išmaniųjų sutarčių testavimo aplinka: Išmaniųjų sutarčių testavimo aplinka turi būti lokali ir izoliuota nuo tikro „Ethereum“ tinklo, kuriame sutartis ir bus vykdoma.

17 lentelė. Nefunkcinis saugumo reikalavimas „Išmaniųjų sutarčių testavimas“

Reikalavimo Nr.	NSR-5
Pagrindimas	Išmaniosios sutartys testuojamos specialiai sukurtoje lokalioje aplinkoje naudojant fiktyvias blokų grandinės tinklo paskyras.
Tikimo kriterijus	Testavimo scenarijams įgyvendinti naudojami lokali testavimo aplinka

Unikalus balsavimo patikrinimo numeris: Po balsavimo rinkėjui turi būti išduotas unikalus numeris, kurio dėka naudotojas gali sužinoti ar balsas yra įtrauktas į balsavimą.

18 lentelė. Nefunkcinis saugumo reikalavimas „Unikalus balsavimo patikrinimo numeris“

Reikalavimo Nr.	NSR-6
Pagrindimas	Rinkėjui prabalsavus išduodamas unikalus numeris, kurio pagalbas visada galima pasitikrinti ar balsas yra įtrauktas į galutinį rezultatą. Šis unikalus numeris neturi jokio ryšio su naudotoju ir su juo nėra galimybės nustatyti kam šis numeris priklauso, tokiu būdų yra užtikrinimas duomenų anonimiškumas, nes kad atitinkamas numeris priklauso atitinkam žmogui, žinot tik jis pats.
Tikimo kriterijus	Naudotojui sėkmingai prabalsavus pateikiamas unikalus 42 simbolių numeris šešioliktainiu formatu (pvz. 0xAE55bc6273a1d8A2D443646d044f3DE3d21a1d34).

3.1.3.2 Panaudojamumo reikalavimai

Sistemos veikimas nepriklausomai nuo įrenginio: Sistema turi veikti nepriklausomai nuo įrenginio operacinės sistemos

19 lentelė. Nefunkcinis panaudojamumo reikalavimas „Sistemos veikimas nepriklausomai nuo įrenginio“

Reikalavimo Nr.	NPR-1
Pagrindimas	Vartotojas galės sistema naudotis bet kokiu įrenginiu turinčių prieigą prie interneto.
Tikimo kriterijus	Sistema tinkamai veikia, naudojantis bet kuria operacine sistema.

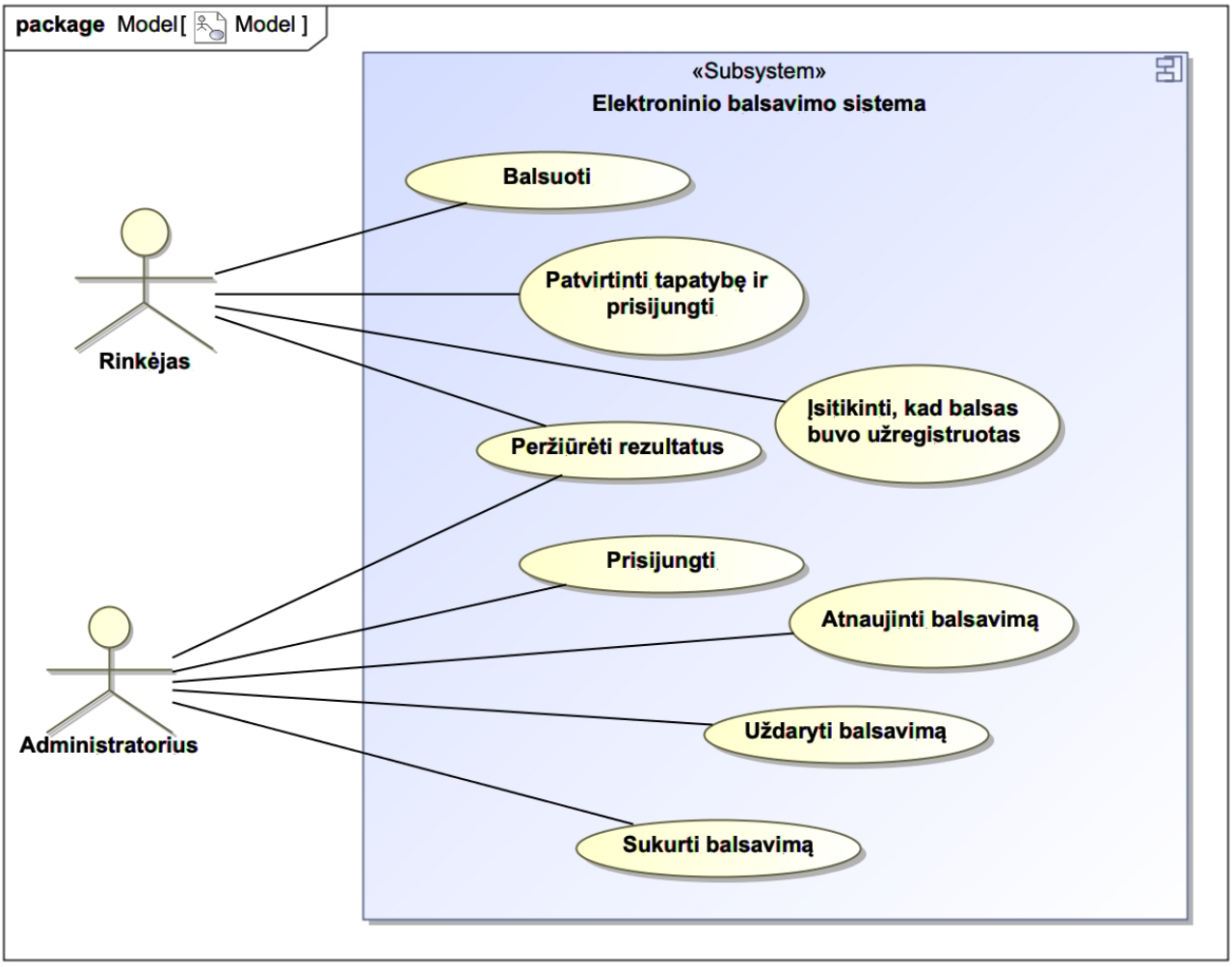
Balsavimo rezultatų atvaizdavimas: Balsavimo rezultatai tūri būti sveikieji skaičiai išreikšti procentine forma.

20 lentelė. Nefunkcinis panaudojamumo reikalavimas „Balsavimo rezultatų atvaizdavimas“

Reikalavimo Nr.	NPR-2
Pagrindimas	Saugant sistemoje sveikuosius skaičius lengviau juos lyginti ir atlikti skaičiavimus.
Tikimo kriterijus	Balsavimo rezultatai pateikiami suapvalinti ir procentine forma, pvz. 25%, 60%, 100%.

3.1.4 Sistemos panaudos atvejai

Šiame poskyryje pateikiama anksčiau aprašytų naudotojų panaudos atvejų diagrama, kurioje parodomi visi naudotojų veiksmai, kuriuos jie gali atlikti sistemoje ir detalus jų aprašymas.



13 pav. Sistemos naudotojų panaudos diagrama

Toliau lentelėse yra, aprašyti ir detalizuoti visi, diagramoje pateikti naudotojų panaudos atvejai.

Balsuoti: Naudotojas pasirinkęs vieną iš balsavimo teikiamų pasiūlymų gali savo pasirinkimą užregistruoti.

21 lentelė. Panaudos atvejis „Balsuoti"

Nr.	1
Aktorius	Rinkėjas

Prieš sąlyga	Naudotojas turi patvirtinta tapatybę ir turi teisę balsuoti, kurią suteikia balsavimo savininkas.
Vykdymo sąlyga	Pasirinktas balsavimo pasiūlymas.
Po sąlyga	Pasirinkimas užregistruotas blokų grandinės tinkle, kaip naujas blokas.

Peržiūrėti rezultatus: Naudotojas gali pasibaigusio balsavimo galutinius rezultatus.

22 lentelė. Panaudos atvejis „Peržiūrėti rezultatus "

Nr.	2
Aktorius	Visi
Prieš sąlyga	Balsavimas pasibaigęs.
Vykdymo sąlyga	Pasirinkta peržiūrėti balsavimo rezultatus.
Po sąlyga	Pateikiami galutiniai balsavimo rezultatai.

Įsitikinti, kad balsas buvo užregistruotas: Naudotojas gali įsitikinti, kad jo balsas buvo užregistruotas balsavime.

23 lentelė. Panaudos atvejis „Įsitikinti, kad balsas buvo užregistruotas "

Nr.	3
Aktorius	Rinkėjas
Prieš sąlyga	Naudotojas yra prabalsavęs, gautas unikalus balso patvirtinimo numeris.
Vykdymo sąlyga	Pasirinkta patvirtinti balso tinkamumą ir įvedamas unikalus balso patvirtinimo numeris.
Po sąlyga	Pateikiama informacija apie balso statusą.

Patvirtinti tapatybę ir prisijungti: Naudotojas gali patvirtinti tapatybę naudodamasis „Elektroniniai valdžios vartai“ programos sąsaja, kad prisijungti prie elektroninio balsavimo sistemos.

24 lentelė. Panaudos atvejis „Patvirtinti tapatybę ir prisijungti "

Nr.	4
Aktorius	Rinkėjas
Prieš sąlyga	Naudotojas turi prieigą prie „Elektroniniai valdžios vartai“ sistemos ir turi teisę balsuoti.
Vykdymo sąlyga	Prisijungiama prie „Elektroniniai valdžios vartai“ sistemos naudojant pasirinktą metodą.
Po sąlyga	Naudotojo tapatybė patvirtinta ir vartotojas prijungiamas prie sistemos

Prisijungti: Administratorius prisijungti prie sistemos gali naudodamas unikalaus numerio ir slaptažodžio kombinaciją.

25 lentelė. Panaudos atvejis „Prisijungti"

Nr.	5
Aktorius	Administratorius
Prieš sąlyga	Naudotojas yra įtrauktas į privačią elektroninio balsavimo sistemos duomenų bazę.
Vykdymo sąlyga	Suvedamas unikalus prisijungimo numeris ir atitinkamas slaptažodis.
Po sąlyga	Naudotojas sėkminga prisijungia prie sistemos.

Sukurti balsavimą: Naudotojas gali sukurti naują, nepriklausomą balsavimą.

26 lentelė. Panaudos atvejis „Sukurti balsavimą"

Nr.	6

Aktorius	Administratorius
Prieš sąlyga	Naudotojas yra prisijungęs, kaip administratorius.
Vykdymo sąlyga	Nurodomas balsavimo pavadinimas ir nurodomi pasirinkimo variantai, bei pabaigos data ir laikas.
Po sąlyga	Sukurtas balsavimas matomas, kaip neaktyvus tik administratoriui.

Atnaujinti balsavimą: Naudotojas gali pakeisti balsavimo statusą iš „Neaktyvus“ į „Aktyvus“, kad balsavimas būtų matomas rinkėjams.

27 lentelė. Panaudos atvejis „Atnaujinti balsavimą"

Nr.	7
Aktorius	Administratorius
Prieš sąlyga	Naudotojas yra prisijungęs, kaip administratorius. Balsavimas yra dar neaktyvuotas.
Vykdymo sąlyga	Pasirenkama paviešinti balsavimą.
Po sąlyga	Balsavimas matomas visiems naudotojams.

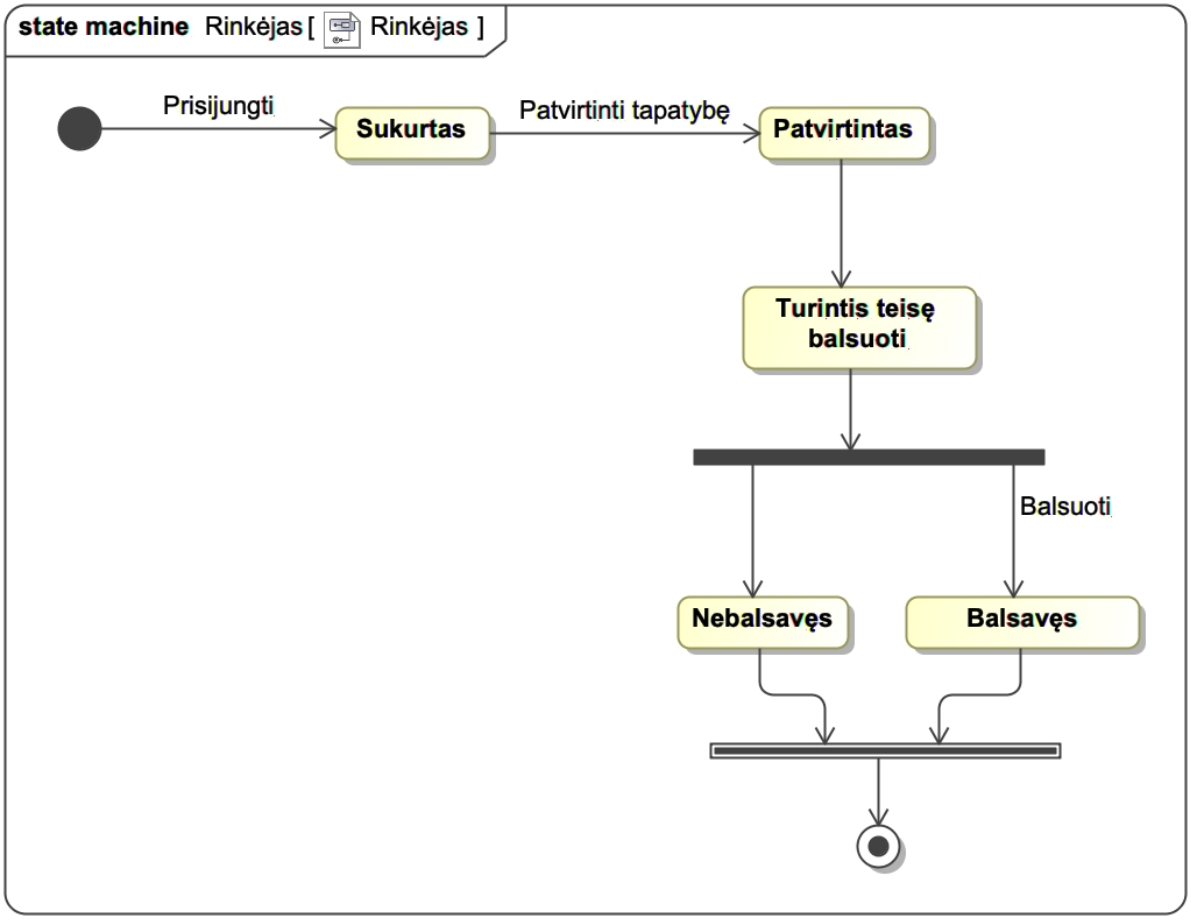
Uždaryti balsavimą: Administratorius gali uždaryti balsavimą jam oficialiai pasibaigus arba iškilus būtinybei tai padaryti.

28 lentelė. Panaudos atvejis „Uždaryti balsavimą“

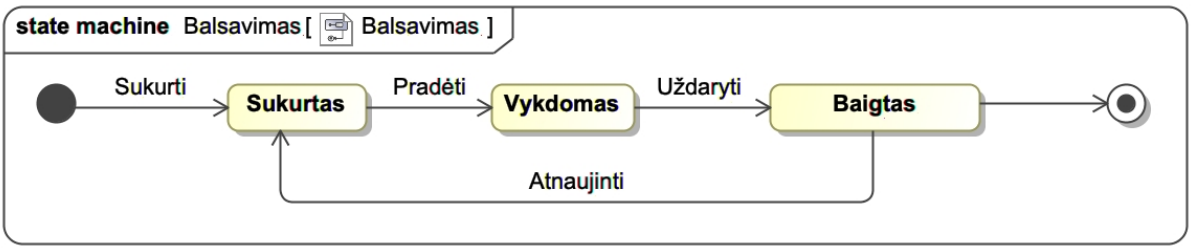
Nr.	8
Aktorius	Administratorius
Prieš sąlyga	Balsavimas yra prasidėjęs.
Vykdymo sąlyga	Pasirenkama sustabdyti balsavimą.
Po sąlyga	Balsavimas uždarytas, rinkėjai negali daugiau balsuoti balsavime.

3.1.5 Sistemos objektų būsenų diagramos

Šiame poskyryje pateikiamos balsavimo ir balsuotojo objekto būsenų diagramos, kuriose parodomi, kokias būsenas gyvavimo metu gali tūrėti kiekvienas iš objektų.



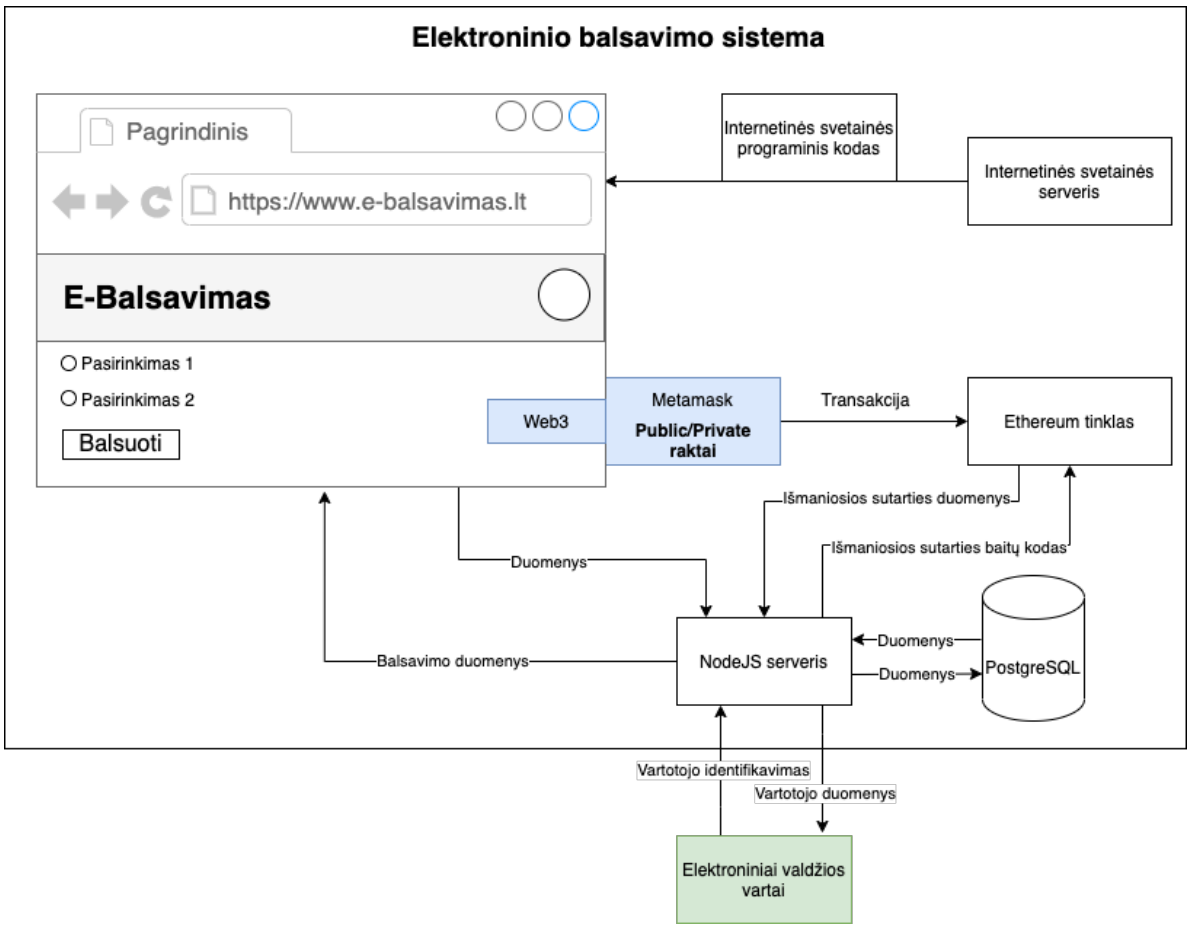
14 pav. Objekto „Rinkėjas" būsenų diagrama



15 pav. Objekto „Balsavimas" būsenų diagrama

3.2 Bendra sistemos architektūra

Bendroje sistemos architektūros diagramoje yra atvaizduojamos naudojamos technologijos ir integracijos su kitomis sistemomis ir ryšiai tarp jų. Visos šios sistemos dalys yra reikalingos bendram elektroninio balsavimo sistemos veikimui.



16 pav. Bendra elektroninio balsavimo sistemos architektūra

Šioje architektūros diagramoje atsispindi pagrindiniai duomenų mainai tarp posistemių ir išorinių sistemų.

3.2.1 Naudojamos technologijos

Šiame poskyryje pateikiamos visos naudojamos technologijos elektroninio balsavimo sistemai įgyvendinti. Kadangi „Ethereum“ išmaniųjų sutarčių technologija yra ganėtinai nauja ir greitai, dažnai atnaujinama įrankiai skirti su jomis dirbti taip pat yra dažnai naujinami, todėl tarpusavyje įrankių versijos turi daugybė didelių pakeitimų, kurie tarpusavyje yra nesuderinami.

29 lentelė. Naudojamos technologijos, jų versijos ir aprašymai

Technologijos/įrankio pavadinimas	Versija	Aprašymas
„Solc“ (“Solidity — Solidity 0.8.0 documentation,” n.d.)	0.4.17	„Solidity“ programinio kodo kompiliatorius.
„Web3js“	1.2.0	Bibliotekų rinkinys, leidžiantis sąveikauti su vietiniu arba nutolusiu „Ethereum“

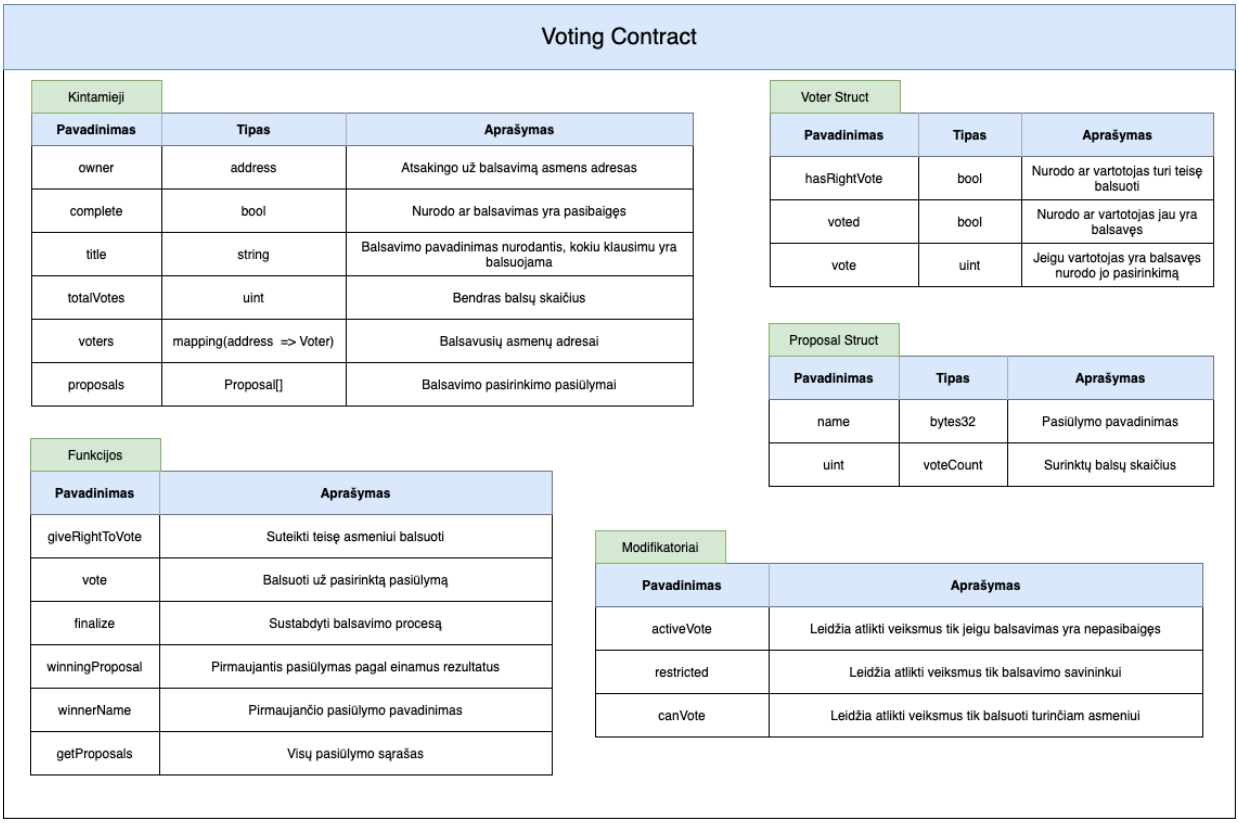
		tinklu naudojant „Javascript“ programavimo kalbą.
„Ganache“	6.12.2	Įrankis skirtas pilnai imituoti „Ethereum“ tinklą ir naudotojų veiksmus jame, skirtas išmaniųjų sutarčių testavimui.
„Mocha“	8.2.1	Komponentų testavimo įrankis, skirtas išmaniųjų sutarčių panaudos atvejams testuoti.
„Rinkeby Ethereum“ tinklas	-	„Ethereum“ testavimo tinklas, skirtas išmaniųjų sutarčių testavimui tikroje aplinkoje naudojant netikrus „Ether“ pinigus atlikti transakcijoms.
„Remix Ethereum“	-	„Ethereum“ išmaniųjų sutarčių kūrimui ir paleidimui.
VueJS	2.6.11	Vartotojo sąsajos dalies programavimo karkasas.
„Truffle hdwallet provider“	0.0.3	Tarpininkas tarp „Web3“ ir „Metamask“, skirtas gauti ir perduoti informaciją apie naudotojo „Ethereum“ paskyrą.
„Metamask“	-	Tarpininkas tarp naudojo ir „Ethereum“ tinklo ir jame esančių išmaniųjų sutarčių.
„Infura“	-	Sukuria ryšį tarp nutolusio „Ethereum“ blokų grandinės tinklo, kuris vėliau naudojamas išmaniųjų sutarčių paleidimui.
„NodeJS“	12.16.1	„Javascript“ programinio kodo vykdymo aplinka.
ExpressJS	4.7.11	Vidinės sąsajos programavimo karkasas.
„MongoDB“	4.2	„NoSQL“ nereliacinė duomenų bazė.

3.3 Išmaniosios sutartys

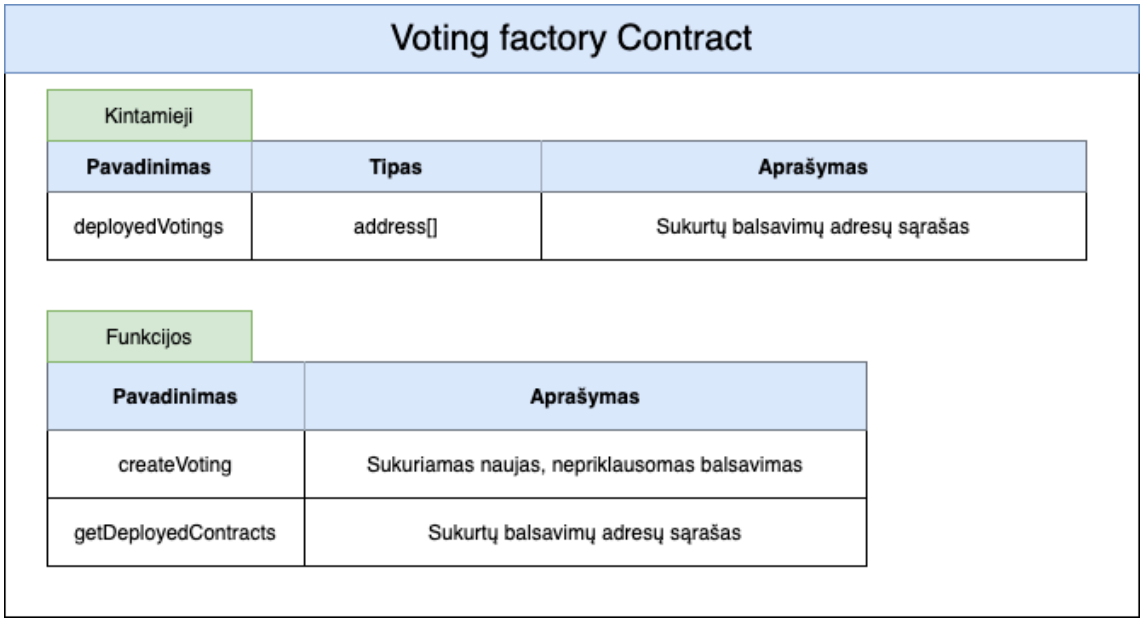
Elektroninio balsavimo sistemos „Ethereum“ blokų grandinės tinkle naudojamos dvi išmaniosios sutartys:

1. „Voting Contract“;
2. „Voting factory Contract“.

Iš kurių pagrindinė yra „Voting Contract“, nes joje yra saugoma visa informacija apie vykstantį balsavimą. Antroji „Voting factory Contract“ išmanioji sutartis yra pagalbinė naudojama inicializuoti naujus balsavimus „Ethereum“ blokų grandinės tinkle. Tokiu būdų nereikia pakartotinai rankiniu būdu kompiliuoti „Voting Contract“ ir jo išskleisti tinkle, tai galima padaryti pasinaudojus „createVoting“ funkcija „Voting factory“ išmaniojoje sutartyje.

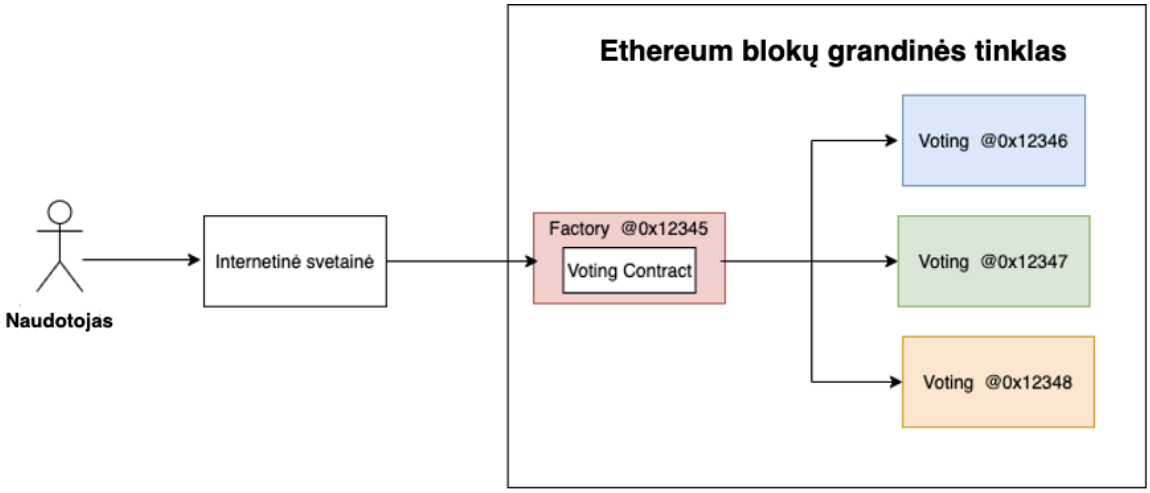


17 pav. „Voting" išmaniosios sutarties struktūra



18 pav. „Voting factory" išmaniosios sutarties struktūra

Sekančioje diagramoje parodoma, kaip dvi ankščiau minėtos sutartys yra susijusios tarpusavyje.



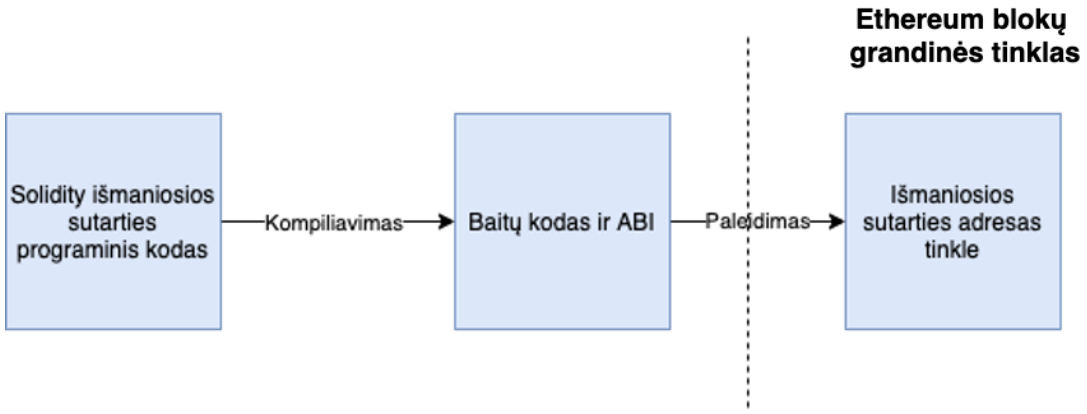
19 pav. „Voting" ir „Voting Factory" išmaniųjų sutarčių tarpusavio ryšys

Pateikta diagrama parodo panaudos atvejį, kai naudotojas naudodamasis elektroninio balsavimo sistemos internetine svetaine „Ethereum“ blokų grandinės tinkle sukuria kelias „Voting“ išmaniąsias sutartis, kurioms sukurti yra naudojama tik viena „Voting Factory“ sutartis į kurią yra paduodama „Voting“ išmanioji sutartis t. y. jos programinis kodas.

3.3.1 Išmaniųjų sutarčių kompiliavimas

Išmaniųjų sutarčių kompiliavimo ir paleidimo procesas yra vienas svarbiausių, kuriuo metu turi būti užtikrinta, kad išmaniosios sutarties programinis kodas yra tinkamas naudojimui ir yra saugus. Taip pat svarbu užtikrinti, kad programinio kodo vykdymas „Ethereum“ blokų grandinės tinkle bus saugus ir logiškas, nes kitu atveju kenkėjiškas vartotojas galės pasinaudoti saugumo spragomis ir pasipelnyti iš išmaniosios sutarties. Sekantis svarbus aspektas prieš paleidžiant išmaniąją sutartį paskaičiuoti ir įsitikinti kiek išviso kainuos visos sutarties vykdymas didėjant naudotojų skaičiui. Kiekviena operacija blokų grandinės tinkle yra apmokestinama „dujomis“, viena operacija kainuoja viena labai mažą „Ehter“ dalį, tačiau jeigu naudotojų skaičius išauga tai išauga ir kaina kiekvienai operacijai atlikti.

Naudojant „Solidity“ kompiliatorių išmaniosios sutarties kodas parašytas „Solidity“ programavimo kalba yra sukompiliuojamas į baitų kodą (šis bus vykdomas „Ethereum“ blokų grandinės tinkle) ir į programos dvejetainė sąsaja (*angl. Application Binary Internface – ABI*). Ši dvejetainė programos sąsaja reikalinga tam, kad galima būtų sąveikauti su išmaniąja sutartimi esančia „Ethereum“ blokų grandinės tinkle, ši sąsaja ir padeda išlaikyti ryšį tarp baitų kodo ir išmaniosios sutarties savininko.



20 pav. Išmaniosios sutarties kompiliavimo ir paleidimo diagrama

Išmaniųjų sutarčių kompiliavimo procesas vykdomas naudojant „Javascript“ programavimo kalbą ir „Solc“ įrankį, sukompiliuotas kodas išsaugomas JSON failo formatu „**./ethereum/build**“ kataloge, kuris vėliau gali būti naudojamas naujų sutarčių paleidimui arba jų testavimui lokaliame „Ethereum“ tinkle naudojant „Mocha“ ir „Ganache“.

Išmaniųjų sutarčių kompiliavimo programinis kodas:

```
const path = require('path');
const solc = require('solc');
const fs = require('fs-extra');

const buildPath = path.resolve(__dirname, 'build');

fs.removeSync(buildPath);

const contractPath = path.resolve(__dirname, 'contracts', 'Voting.sol');
const source = fs.readFileSync(contractPath, 'utf8');
const output = solc.compile(source, 1).contracts;

fs.ensureDirSync(buildPath);

for (let contract in output) {
    fs.outputJsonSync(
        path.resolve(buildPath, `${contract.replace(':', '')}.json`),
        output[contract]
    );
}
```

Išmaniųjų sutarčių paleidimo programinis kodas:

```
const HDWalletProvider = require('truffle-hdwallet-provider');
const Web3 = require('web3');
const compiledFactory = require('./build/VotingFactory.json');

const provider = new HDWalletProvider(
    'wait three visual cinnamon wood loan giraffe love artist output top grass',
    'https://rinkeby.infura.io/v3/e945d00d15c64c12b4e2ac8f67832dba'
);

const web3 = new Web3(provider);

const deploy = async () => {
    const accounts = await web3.eth.getAccounts();
    const [account] = accounts;

    const result = await new web3.eth.Contract(JSON.parse(compiledFactory.interface))
        .deploy({ data: compiledFactory.bytecode, arguments: [] })
        .send({ gas: '1000000', from: account });

    console.log('Contract deployed to', result.options.address);
};

deploy();
```

3.3.2 Išmaniųjų sutarčių saugumo užtikrinimas

Išmaniųjų sutarčių „Voting“ ir „VotingFactory“ funkcijų saugumui užtikrinti yra naudojami „Solidity“ programavimo kalbos modifikatoriai. Jų veikimo principas yra paprastas, naudojant „require“ funkciją yra nurodoma, kokią sąlygą privalo atitikti vartotojas arba kitų objektų būsenos, kad leisti įvykdyti veiksmą, kuris yra apsaugotas modifikatorių pagalba. Jeigu „require“ funkcijos sąlyga yra neišpildoma transakcija yra nutraukiama ir sutarties būsena gražinama į pradinę.„Voting“ išmaniojoje sutartyje yra naudojami šie modifikatoriai:

- „activeVote“ – leidžia atlikti veiksmus tik tuo atveju jeigu balsavimo būsena yra aktyvi;
- „restricted“ – leidžia atlikti veiksmus tik balsavimo savininkui;
- „canVote“ – leidžia atlikti veiksmus tik balsuotojui, kuris turi teisę balsuoti ir turi patvirtintą tapatybę.

Šis saugumo mechanizmas leidžia užtikrinti, kad kenkėjiškas vartotojas negalės pasinaudoti išmaniosios sutarties funkcijomis jeigu neatitiks jose aprašytų sąlygų.

3.4 „Elektroniniai valdžios vartai“ sistemos integracija

Integracija su „Elektroniniai valdžios vartai“ sistema (*toliau EVVS*) yra įgyvendinta naudojantis jau sukurta testavimo aplinka, kurios dokumentacija ir visi reikalingi programiniai kodai yra pateikiama oficialioje „Elektroniniai valdžios vartai“ svetainėje skirtoje sistemų kūrėjams (“Elektroniniai valdžios vartai,” n.d.).

Pagal pateiktą dokumentaciją, naudojantis jau sukurta „AuthGenerator“ klase, buvo pagaminta „SOAP XML“ užklausa, skirta testavimui, kurios dėka gautas unikalus testavimo numeris reikalingas, integracijai įgyvendinti. Toliau pateikiamas sugeneruotos užklausos kodas.

```
<?xml version="1.0" encoding="UTF-8"?>
<authentication:authenticationRequest
xmlns:authentication="http://www.epaslaugos.lt/services/authentication"
xmlns:dsig="http://www.w3.org/2000/09/xmldsig#"
xmlns:ns3="http://www.w3.org/2001/10/xml-exc-c14n#" id="uniqueNodeId">
    <authentication:pid>VSID000000000113</authentication:pid>

<authentication:authenticationProvider>auth.lt.identity.card</authenticatio
n:authenticationProvider>

<authentication:authenticationProvider>auth.lt.bank</authentication:authent
icationProvider>
    <authentication:authenticationAttribute>lt-personal-
code</authentication:authenticationAttribute>
    <authentication:authenticationAttribute>lt-company-
code</authentication:authenticationAttribute>
    <authentication:authenticationAttribute>eidas-
eid</authentication:authenticationAttribute>

<authentication:userInformation>firstName</authentication:userInformation>

<authentication:userInformation>lastName</authentication:userInformation>

<authentication:userInformation>companyName</authentication:userInformation
>

<authentication:postbackUrl>https://localhost</authentication:postbackUrl>
    <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
        <SignedInfo>
            <CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                <InclusiveNamespaces xmlns="http://www.w3.org/2001/10/xml-
exc-c14n#" PrefixList="authentication" />
            </CanonicalizationMethod>
            <SignatureMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
            <Reference URI="#uniqueNodeId">
                <Transforms>
                    <Transform
Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
                    <Transform Algorithm="http://www.w3.org/2001/10/xml-
exc-c14n#">
                        <InclusiveNamespaces
xmlns="http://www.w3.org/2001/10/xml-exc-c14n#" PrefixList="authentication"
/>
                    </Transform>
                </Transforms>
                <DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
                <DigestValue>iuNrswB033tjbXtqpsy+GrT57uY=</DigestValue>
            </Reference>
        </SignedInfo>
```

```
<SignatureValue>hoLyi6hkt5v6M6qNFYARkVW/0Ioq6Z0V5rXFL+itORFtXCpTjzfXGRazPcX
gW9OzTOES3gem0vRf&#xD;

tv26hPsla5K4goRy9X35V//aYKMRJljBFhzjfwUo39/ZAQE3O+98w62zw2PwNF81h7Ou9S3DXKR
6&#xD;

kf4B7qjfYS2cleonreWm9RSJd0/xYyYjsScj+rB7tnFNG9nAamh79PiFtEyxXrwirDdjR6+vvrr
c&#xD;

OCtm7JpsOZhmz5tuQtZGC2Syf8imnibpBtHjfY675opbk2GRQ6AswLnG6f+KEJdryz2ZqS/ZIwS
b&#xD;
            ko16Of+zFjI9CLonCTAfptPV2niQxUwaTicCbw==</SignatureValue>
        <KeyInfo>
            <KeyValue>
                <RSAKeyValue>

<Modulus>m50I4AZwBGxKhrYDS6qwoGCry1n+kZCL4rSI5ikq6srt8a0Amg6JCgX3akRcdf3Ktf
3XZ+1xZSSQ&#xD;

iWNpCen4Hi7K/2tsSmmLWhCp4Hq+kEAXI5JrmCgH51YJpW8Z2sw7QKWkbvQ+7rdwqXAnEGxp1oM
W&#xD;

3559l2WDRzDaC/iVmiFo8S4I3ak/wSHFiCr+MxW4vQ/7WirW2FqfUIFKVtCC4IuqJm0wKvsupdO
A&#xD;

c6LF6CKeTM5dc8hHS67cPLdc3ev7pR5tUD4xQECCeMd81s+9eZEUoH84djZLRnXx3WhxFgDRwfc
c&#xD;
                        VUYYup4Za2E8SEHSeXkwTueSBiIx8qibANDAmw==</Modulus>
                    <Exponent>AQAB</Exponent>
                </RSAKeyValue>
            </KeyValue>
        </KeyInfo>
    </Signature>
</authentication:authenticationRequest>
```

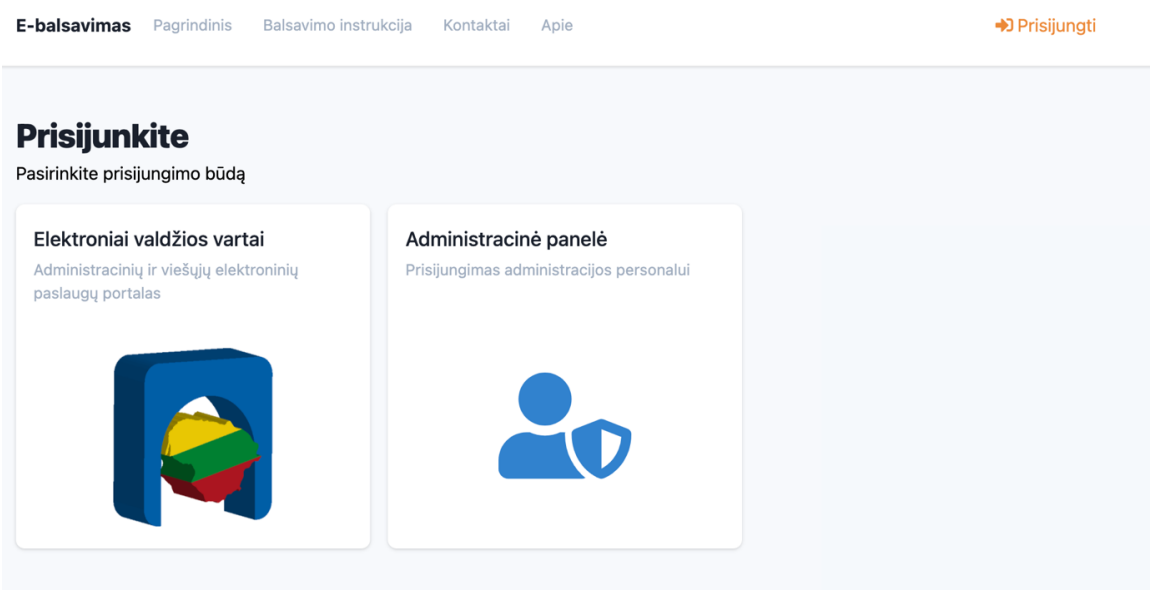
Atlikus užklausą į serverį yra gaunamas unikalus kodas, kurį sudaro 36 simboliai (pvz. 8c3b503c-bbee-11eb-8529-0242ac130003), kuri reikia pateiktį į EVVS testavimo serverį, kad galima būtų pradėti naudotis šia integracija.

Dėka šios EVVS testavimo aplinkos galima lengvai imituoti Lietuvos Respublikos piliečių tapatybės nustatymą naudojant atskirą sistemą ir taip identifikuoti asmenis turinčius teisę balsuoti. Elektroninio balsavimo sistemoje yra labai svarbu, kad sistema galėtų naudotis tik teisę balsuoti turintys Lietuvos piliečiai.

3.5 Grafinė vartotojo sąsaja

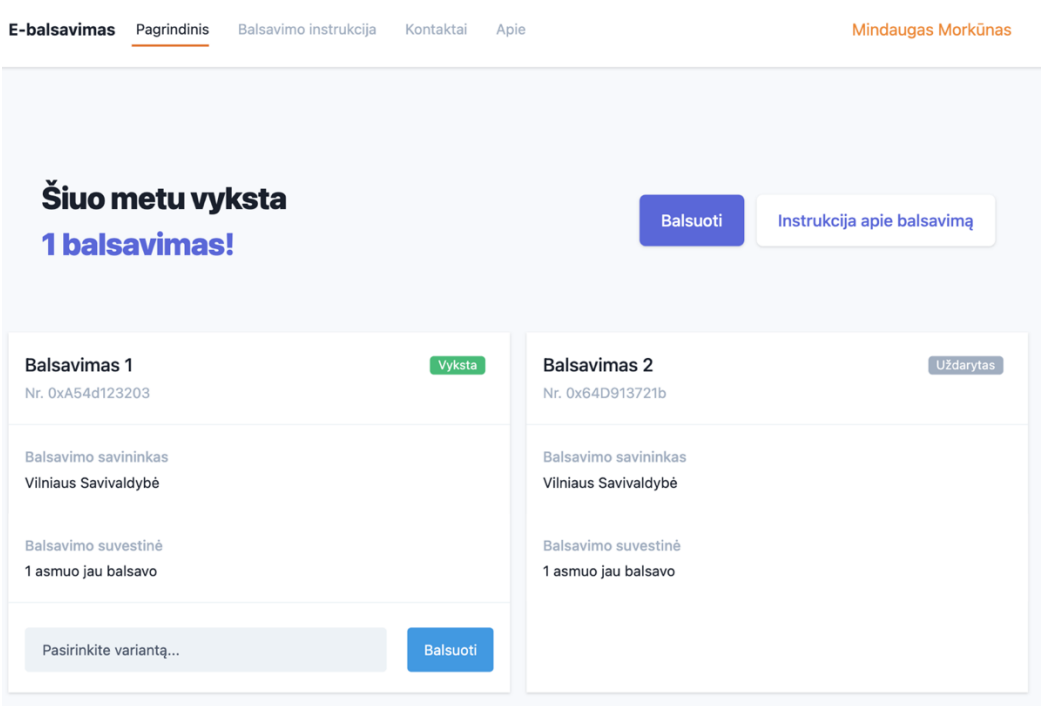
Šiame poskyryje pateikiami grafinės vartotojo sąsajos ekrano vaizdai, kuriuos parodoma, kaip atrodo sukurtos sistemos prisijungimo, balsavimo, balsavimo rezultatų peržiūros ir balso patikrinimo puslapiai.

Prisijungimo puslapis.



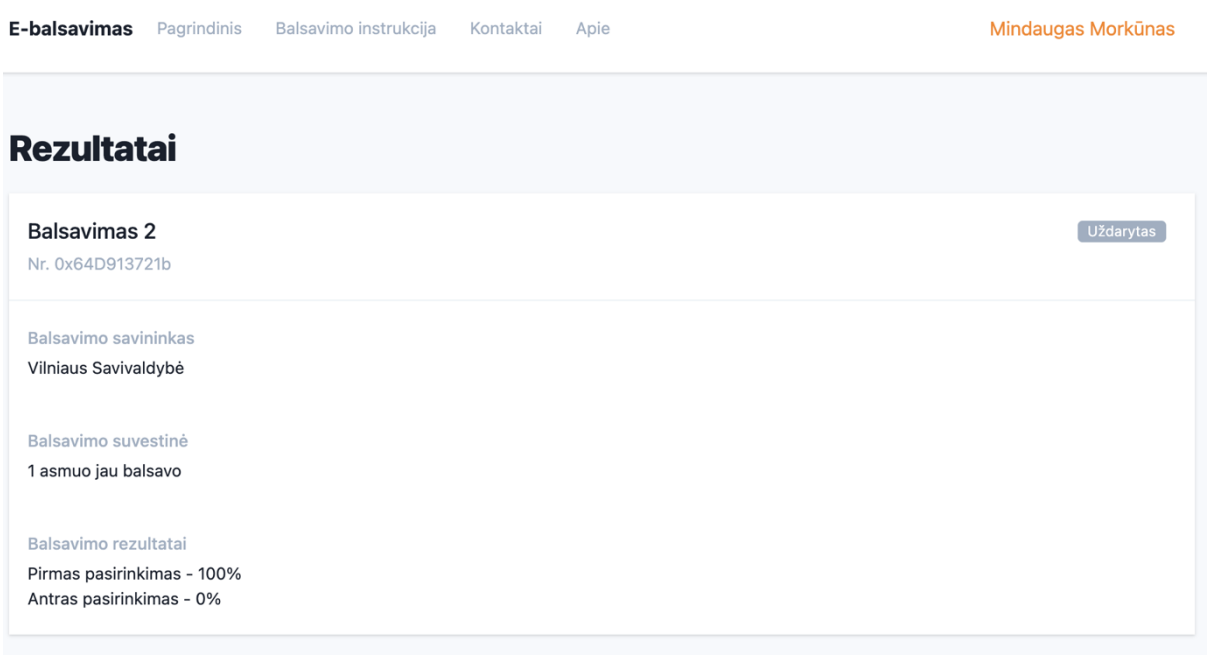
21 pav. Prisijungimo puslapis

Pagrindinis balsavimo puslapis.



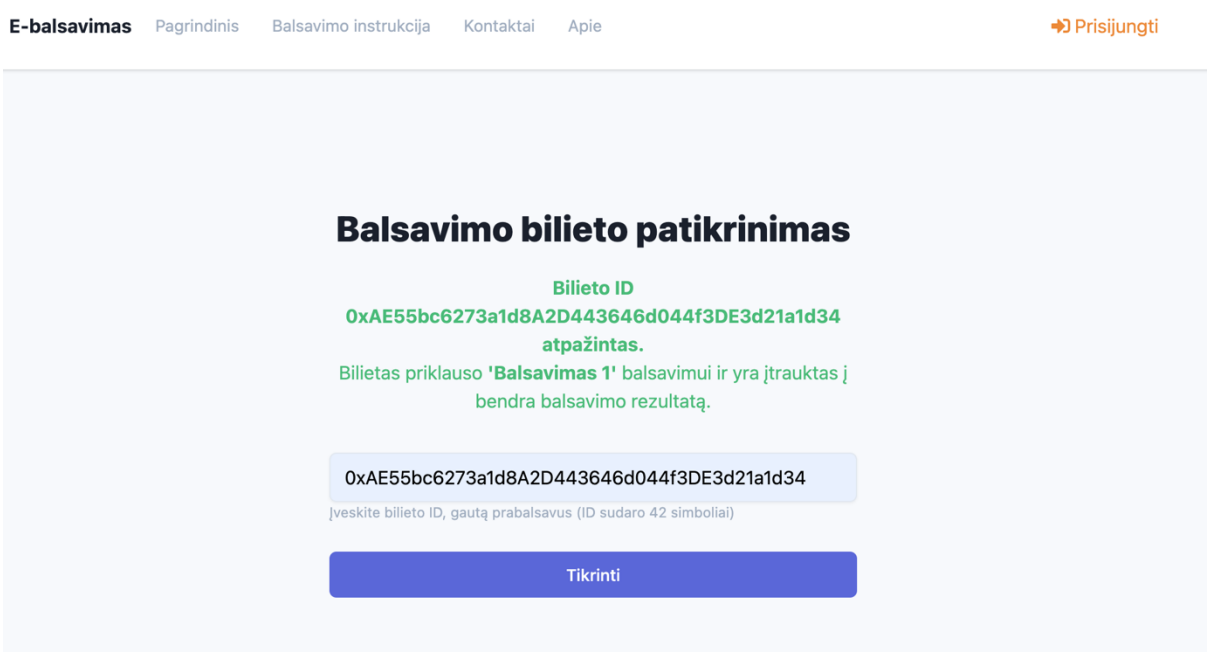
22 pav. Pagrindinis balsavimo puslapis

Balsavimo rezultatų puslapis.



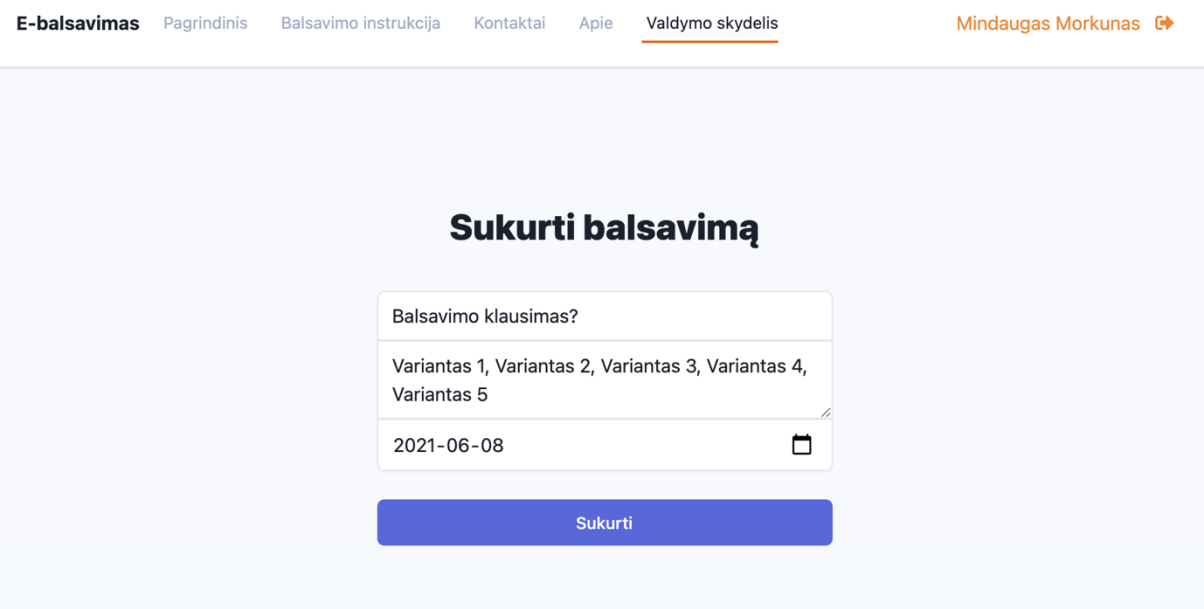
23 pav. Balsavimo rezultatų puslapis

Balsų patikrinimo puslapis.



24 pav. Balsų patikrinimo puslapis

Naujo balsavimo sukūrimo puslapis.



25 pav. Naujo balsavimo sukūrimo puslapis

3.6 Sistemos testavimas

Testavimo metu siekiama patikrinti sukurtą elektroninio balsavimo sistemą pagal įvairius testavimo atvejus ir įsitikinti, kad sistema yra paruošta ir tinkama viešam naudojimui , taip pat nekelia grėsmės vartotojų duomenų saugumui ir atitinka reikalavimų specifikacijoje aprašytus reikalavimus.

Kiekvieno atlikto testavimo atvejo žingsnio statusas yra žymimas: T – teisingas arba N – neteisingas.

3.6.1 Sistemos funkcijų testavimas

Žemiau esančioje lentelėje pateikiami sistemos testavimo scenarijai, kurių sėkmingas įgyvendinimas yra svarbiausias.

Scenarijaus Nr.	Funkcinio reikalavimo Nr.	Scenarijaus pavadinimas	Scenarijaus svarba
TS-01	FR-2	Balsavimas	Kritinė
TS-02	FR-1	Prisijungimas naudojant „Elektroninių valdžios vartų“ programos sąsają	Aukšta
TS-03	-	Prisijungti, kaip administratoriui	Aukšta
TS-04	FR-5	Sukurti naują balsavimą	Aukšta

Visiems lentelėse pateiktiems scenarijams yra aprašomi testavimo atvejai ir jų vykdymo žingsniai, kokio rezultato tikimasi ir koks gautas rezultatas įvykdžius testavimo atvejo žingsnius.

3.6.1.1 „Balsavimas“ (TS-01) scenarijaus testavimo atvejai

Prieš sąlyga: Prisijungti prie sistemos, kaip „Rinkėjas“.

30 lentelė. „Balsavimo“ testavimo atvejai

Testavimo atvejis	Žingsniai	Duomenys	Laukiamas rezultatas	Gautas rezultatas	Statusas
Pasirinktas balsavimo variantas (TS-01.1)	1. Atidaryti balsavimų sąrašą				T
	2. Pasirinkti aktyvų balsavimą				T
	3. Pasirinkti vieną iš balsavimo variantų	Pirmas			T
	4. Paspausti mygtuką „Balsuoti“				T
	5. Patvirtinti balsavimo transakciją naudojant „Metamask“		Atvaizduojamas pranešimas apie sėkmingą balsavimą ir pateikiamas balsavimo numeris	Pateiktas balsavimo numeris	T
Nepasirinktas balsavimo variantas (TS-01.2)	1. Atidaryti balsavimų sąrašą				T
	2. Pasirinkti aktyvų balsavimą				T
	3. Paspausti mygtuką „Balsuoti“		Atvaizduojamos klaidos pranešimas, kad nepasirinktas balsavimo variantas	Atvaizduojamas klaidos pranešimas, balsas neužskaitytas	T

Atmesta transakcija (TS-01.3)	1. Atidaryti balsavimų sąrašą				T
	2. Pasirinkti aktyvų balsavimą				T
	3. Pasirinkti vieną iš balsavimo variantų	Pirmas			T
	4. Paspausti mygtuką „Balsuoti“				T
	5. Atmesti balsavimo transakciją naudojant „Metamask“		Atvaizduojamos klaidos pranešimas, kad transakcija nesėkminga	Atvaizduojamas klaidos pranešimas, balsas neužskaitytas	T

3.6.1.2 „Prisijungti naudojant „Elektroninių valdžios vartų“ (TS-02) programos sąsają scenarijaus testavimo atvejai

Po sąlyga: Atsijungti iš sistemos.

31 lentelė. „Prisijungti naudojant „Elektroninių valdžios vartų“ programos sąsają“ testavimo atvejai

Testavimo atvejis	Žingsniai	Duomenys	Laukiamas rezultatas	Gautas rezultatas	Statusas
Sėkmingas prisijungimas (TS-02.1)	1. Atidaryti prisijungimo puslapį				T
	2. Pasirinkti „Elektroniniai valdžios vartai“				T
	3. Prisijungti prie „Elektroninių valdžios vartų“		Vartotojas gražintas atgal į balsavimo sistemos pagrindinį puslapį ir sukurta sesija	Vartotojas pagrindiniame puslapyje, sukurta prisijungimo sesija	T

Nesėkmingas prisijungimas (TS-02.2)	1. Atidaryti prisijungimo puslapį				T
	2. Pasirinkti „Elektroniniai valdžios vartai“				T
	3. Neprisijungti prie „Elektroninių valdžios vartų“		Vartotojas gražintas atgal į balsavimo sistemą, nauja sesija nesukurta	Vartotojas pagrindiniame puslapyje, nauja sesija nesukurta ir balsuoti negalima.	T

3.6.1.3 „Prisijungti, kaip administratoriui“ (TS-03) scenarijaus testavimo atvejai

Prieš sąlyga: Sukurtas „12345678” vartotojas.

Po sąlyga: Atsijungti iš sistemos.

32 lentelė. „Prisijungti, kaip administratoriui“ testavimo atvejai

Testavimo atvejis	Žingsniai	Duomenys	Laukiamas rezultatas	Gautas rezultatas	Statusas
Visi laukeliai teisingi (TS-03.1)	1. Atidaryti prisijungimo langą				T
	2. Įvesti naudotojo ID	12345678			T
	3. Įvesti slaptažodį	Testas1234			T
	4. Paspausti „Prisijungti“		Prisijungta sėkmingai	Prisijungta sėkmingai	T
Neteisingas naudotojo ID formatas (TS-03.2)	1. Atidaryti prisijungimo langą				T
	2. Įvesti naudotojo ID	Atsitiktinis tekstas			T
	3. Įvesti slaptažodį	Testas1234			T
	4. Paspausti prisijungti		Prisijungti nepavyko	Prisijungti nepavyko	T

Neegzistuojantis naudotojo ID (TS-03.3)	1. Atidaryti prisijungimo langą				T
	2. Įvesti el. paštą	Atsitiktinis tekstas			T
	3. Įvesti slaptažodį	Testas1234			T
	4. Paspausti prisijungti		Prisijungti nepavyko	Prisijungti nepavyko	T
Neteisingas slaptažodis (TS-03.4)	1. Atidaryti prisijungimo langą				T
	2. Įvesti naudotojo ID	12345678			T
	3. Įvesti slaptažodį	Atsitiktinis tekstas			T
	4. Paspausti prisijungti		Prisijungti nepavyko	Prisijungti nepavyko	T
Visi laukeliai tušti (TS-03.5)	1. Atidaryti prisijungimo langą				T
	2. Įvesti naudotojo ID				T
	3. Įvesti slaptažodį				T
	4. Paspausti prisijungti		Prisijungti nepavyko	Prisijungti nepavyko	T

3.6.1.4 „Sukurti naują balsavimą“ (TS-04) scenarijaus testavimo atvejai

Prieš sąlyga: Prisijungti prie sistemos, kaip „Administratorius“

33 lentelė. „Sukurti naują balsavimą“ testavimo atvejai

Testavimo atvejis	Žingsniai	Duomenys	Laukiamas rezultatas	Gautas rezultatas	Statusas
Visi laukeliai teisingi (TS-04.1)	1. Atidaryti valdymo skydelį				T

	2. Įvesti balsavimo klausimą	Klausimas NR.1?			T
	3. Įvesti galimus pasirinkimo variantus	Pirmas, antras, trečias			T
	4. Įvesti balsavimo pabaigos datą	2021-06-08			
	5. Paspausti „Sukurti“				T
	6. Patvirtinti transakciją naudojant „Metamask“		Sukurtas naujas balsavimas	Sukurtas naujas balsavimas	T
Pasirinkta praėjusi data (TS-04.2)	1. Atidaryti valdymo skydelį				T
	2. Įvesti balsavimo klausimą	Klausimas NR.1?			T
	3. Įvesti galimus pasirinkimo variantus	Pirmas, antras, trečias			T
	4. Įvesti balsavimo pabaigos datą	2020-06-08			
	5. Paspausti „Sukurti“		Nesukurtas balsavimas, atvaizduojama klaida	Nesukurtas naujas balsavimas, atvaizduojama, kad data yra neteisinga	T
Nepatvirtinta transakcija (TS-04.3)	1. Atidaryti valdymo skydelį				T
	2. Įvesti balsavimo klausimą	Klausimas NR.1?			T

	3. Įvesti galimus pasirinkimo variantus	Pirmas, antras, trečias			T
	4. Įvesti balsavimo pabaigos datą	2021-06-08			
	5. Paspausti „Sukurti“				T
	6. Atmesti transakciją naudojant „Metamask“		Nesukurtas naujas balsavimas	Nesukurtas naujas balsavimas, atvaizduojamas klaidos pranešimas	T

3.6.2 Išmaniųjų sutarčių testavimo atvejai

Šiame poskyryje pateikiami išmaniųjų sutarčių „Voting“ ir „VotingFactory“ testavimo atvejai. Sekančiose lentelėse yra aprašomi testavimo atvejai ir jų vykdymo žingsniai, kokio rezultato tikimasi ir koks gautas rezultatas įvykdžius testavimo atvejo žingsnius.

3.6.2.1 „Voting“ išmaniosios sutarties komponentų testai

Testų prieš sąlygos:

1. Testavimo vartotojų paskyrų sąrašas;
2. Sukurta „VotingFactory“ išmanioji sutartis;
3. Sukurta „Voting“ išmanioji sutartis naudojantis „VotingFactory“;

34 lentelė. Išmaniųjų sutarčių testavimo atvejai

Nr.	Testavimo atvejis	Įvedimo reikšmė	Laukiamas rezultatas	Statusas
1	„Ethereum“ blokų grandinės tinkle paleidžiamos „Voting“ ir „VotingFactory“ išmaniosios sutartys	-	Abiejų sutarčių adresai	T
2	Sukurti naują balsavimą (*createVoting)*	„Pavadinimas",[„pasiūlymas „bytes32” formatu", „pasiūlymas „bytes32” formatu“], „savininko adresas"	„Voting“ sutarties adresas	T

3	Suteikti balsuotojui teisę balsuoti (*giveRightToVote)*	„balsuotojo adresas“	Balsuotojas turi teisę atlikti balsavimą. (Voter.*hasRightVote = true*)	T
4	Balsuoti (*vote)*	„pasiūlymo numeris“	Balsas užskaitytas, (*Proposal.voteCount++, totalVotes++)*	T
5	Uždaryti balsavimą (*finalize)*	„balsavimo adresas“	Balsavimas uždarytas (*complete =true)*	T
6	Sužinoti balsavimo nugalėtoją (*winningProposal)*	-	Balsavimas surinkęs daugiausiai balsų (*Proposal)*	T

3.7 Sistemos suderinamumo testavimas

Šiame skyriuje testuojamas sistemos suderinamumas su naršyklėmis ir skirtingomis jų versijomis ir vaizdo prisitaikymas prie įrenginio ekrano dydžio.

Žemiau pateiktoje lentelėje aprašyti sistemos reikalavimai, kurie turi būti įgyvendinti remiantis įvestimi, šiuo atveju skirtingos naršyklės ir skirtingos jų versijos.

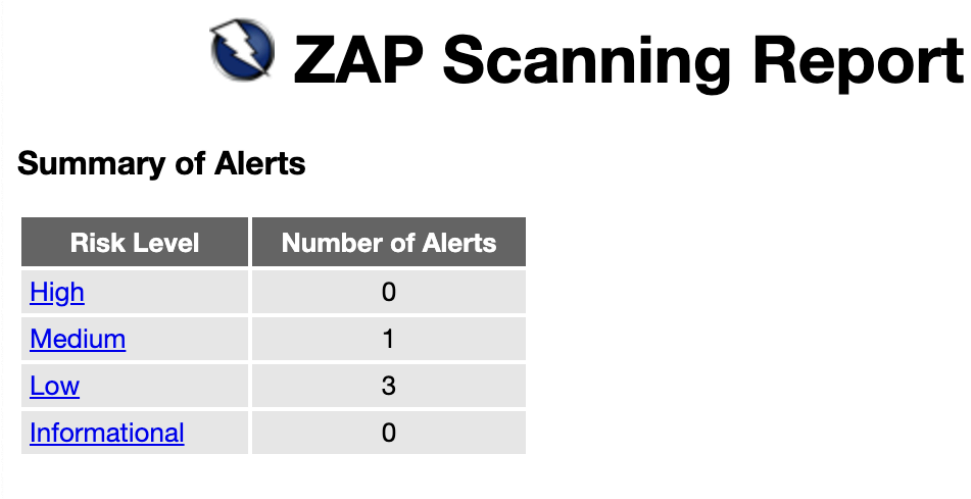
35 lentelė Naršyklių suderinamumo testavimas

Nr.	Reikalavimai/tikslai	Įvestis	Laukiamas rezultatas	Gautas rezultatas	Statusas
ST-1	Tikrinamas sistemos veikimas įvairiose naršyklėse	Naudojama „*Chrome*“ naršyklė versija 90.0.4430.212	Sistema pilnai funkcionuoja be jokių sutrikimų	Sistema pilnai funkcionuoja be jokių sutrikimų	+
ST-2	Tikrinamas sistemos veikimas įvairiose naršyklėse	Naudojama „*Firefox*“ naršyklė versija 87.0	Sistema pilnai funkcionuoja be jokių sutrikimų	Sistema pilnai funkcionuoja be jokių sutrikimų	+
ST-3	Tikrinamas sistemos veikimas įvairiose naršyklėse	Naudojama „*Safari*“ naršyklė versija 14.1	Sistema pilnai funkcionuoja be jokių sutrikimų	Sistema pilnai funkcionuoja be jokių sutrikimų	+

3.8 Vartotojo sąsajos saugumo testavimas

Saugumo testavimas buvo atliekamas naudojant „*OWASP Zed (ZAP)*“ įrankis, kuris skirtas ieškoti svetainių ir vykdomų užklausų pažeidžiamumams.

Atlikus sistemos saugumo testavimą „*ZAP*“ programą sugeneravimo ir pateikė rezultatų išrašą.



ZAP Scanning Report

Summary of Alerts

Risk Level	Number of Alerts
High	0
Medium	1
Low	3
Informational	0

26 pav. Vartotojo sąsajos saugumo testavimo rezultatai

Iš rezultatų galima daryti išvadas, kad sistema yra saugi naudojimui, nes neturi kritiniu spragu, kurios galėtu kelti grėsmę duomenų ir sistemos saugumui. Taip pat „*Zap*“ programa pateikia informaciją apie rastas saugumo spragas ir sprendimo būdus kaip jas pašalinti ir užtikrinti 100 procentinį sistemos saugumą. Tokius sistemos testavimus reiktu atlikti kartą į metus, nes technologijos sensta ir jose atsiranda saugumo spragų.

3.9 Išvados

Sukurta elektroninio balsavimo sistema paremta „Ethereum“ blokų grandinės išmaniųjų sutarčių atitinka jai keliamus funkcinius ir nefunkcinius reikalus, aprašytus reikalavimų specifikacijoje, kurios pagrindiniai aspektai yra duomenų saugumo užtikrinimas ir balsavimo proceso skaidrumas. Atliktas sistemos testavimas parodo, kad sistema ir sukurtas balsavimo protokolas yra tinkamas tolimesniam vystymui ir jo pritaikymui realiame naudojime valstybės demokratinių rinkimų procesuose. Integracija su „Elektroniniai valdžios vartai“ sistema leidžia užtikrinti, kad balsavime galės dalyvauti tik Lietuvos Respublikos piliečiai turintys teisę dalyvauti rinkimuose.

Išvados

1. Remiantis analitinėje dalyje atliktu blokų grandinės technologijos tyrimu ir jau egzistuojančių elektroninio balsavimų sistemų paremtų blokų grandinės technologijos analize, galima daryti išvadą, kad blokų grandinės technologija, ypač „Ethereum“, išmaniųjų sutarčių sprendimas yra tinkamas ne tik finansų, bet ir kitoms taikymo sritims, kurių vienas pagrindinių tikslų yra užtikrinti duomenų saugumą, konfidencialumą ir jų vientisumą bei to proceso skaidrumą. Būtent dėl šių sprendžiamų problemų blokų grandinės technologijos pritaikymas elektroniniam balsavimui labai tinkamas.
2. Buvo pasiūlytas „Ethereum“ blokų grandinės išmaniųjų sutarčių taikymo metodas, siekiant spręsti balsavimo proceso skaidrumo problemą. Pasiūlytą metodą sudaro naudotojų identifikavimas naudojant „Elektroninių valdžios vartų“ programos sąsają, „Ethereum“ blokų grandinės technologijos tinklą, išmaniąsias sutartis. Toks metodas leis užtikrinti balsavimo proceso saugumą, skaidrumą, galimybę įsitikinti, kad balsas tikrai buvo įtrauktas į galutinį rezultatą.
3. Suprojektuota sistema paremta išmaniosiomis sutartimis, veikianti „Ethereum“ blokų grandinės technologijos tinkle, su „Elektroninių valdžios vartų“ programos sąsajos integracija naudotojams autentifikuoti, su galimybe patikrinti, ar balsas buvo įtrauktas į balsavimo blokų grandinę. Atlikto testavimo rezultatai parodo, kad sistema tinkama plačiajam naudojimui, nes atitinka keliamus saugumo ir panaudojamumo reikalavimus ir užtikrina sistemos tinkamą veikimą ją vystant ir administruojant.

Literatūros šaltiniai

1. Agora Vote. (2017). Retrieved January 26, 2020, from https://static1.squarespace.com/static/5b0be2f4e2ccd12e7e8a9be9/t/5b6c38550e2e725e9cad3f18/1533818968655/Agora_Whitepaper.pdf
2. Antonopoulos, A. M., & Wood, G. (n.d.). *Mastering Ethereum : building smart contracts and DApps*.
3. Baliga, A. (2017). *Understanding Blockchain Consensus Models*. Retrieved from www.persistent.com
4. Bardhan, P. (2002, September). Decentralization of governance and development. *Journal of Economic Perspectives*, Vol. 16, pp. 185–205. https://doi.org/10.1257/089533002320951037
5. Bitcoin - Open source P2P money. (2019). Retrieved January 14, 2020, from https://bitcoin.org/en/
6. Blockchain Architecture Explained: How It Works & How to Build. (n.d.). Retrieved June 18, 2020, from https://mlsdev.com/blog/156-how-to-build-your-own-blockchain-architecture
7. Buterin, V. (n.d.). *A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM*.
8. Buterin, V. (2012). *A NEXT GENERATION SMART CONTRACT & DECENTRALIZED APPLICATION PLATFORM*.
9. Curran, K. (2018). E-Voting on the Blockchain. *The Journal of the British Blockchain Association*, *1*(2), 1–6. https://doi.org/10.31585/jbba-1-2-(3)2018
10. Elektroniniai valdžios vartai. (n.d.). Retrieved May 23, 2021, from https://www.epaslaugos.lt/portal/content/1257
11. Gatteschi, V., Lamberti, F., Demartini, C., Pranteda, C., & Santamaria, V. (2018). To Blockchain or Not to Blockchain: That Is the Question. *IT Professional*, *20*(2), 62–74. https://doi.org/10.1109/MITP.2018.021921652
12. Gibson, J. P., Krimmer, R., Teague, V., & Pomares, J. (2016, August 1). A review of E-voting: the past, present and future. *Annales Des Telecommunications/Annals of Telecommunications*, Vol. 71, pp. 279–286. https://doi.org/10.1007/s12243-016-0525-8
13. Home | Ethereum. (2019). Retrieved January 14, 2020, from https://ethereum.org/
14. i-Voting — e-Estonia. (2020). Retrieved January 8, 2020, from https://e-

estonia.com/solutions/e-governance/i-voting/

15. Instantly Move Money to All Corners of the World | Ripple. (2020). Retrieved January 14, 2020, from https://ripple.com/
16. Lindman, J., Tuunainen, V. K., & Rossi, M. (2017). Opportunities and Risks of Blockchain Technologies – A Research Agenda. *Hawaii International Conference on System Sciences 2017 (HICSS-50)*. Retrieved from https://aisel.aisnet.org/hicss-50/da/open_digital_services/3
17. Liu, Y., & Wang, Q. (n.d.). *An E-voting Protocol Based on Blockchain*.
18. Luu, L., Chu, D. H., Olickel, H., Saxena, P., & Hobor, A. (2016). Making smart contracts smarter. *Proceedings of the ACM Conference on Computer and Communications Security*, *24-28-October-2016*, 254–269. https://doi.org/10.1145/2976749.2978309
19. Mentor, R. O. (2016). *The Future of Democracy: Blockchain Voting*. Retrieved from http://www.nytimes.com/2016/12/09/us/
20. Nakamoto, S. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. Retrieved from www.bitcoin.org
21. *Polys-the digital voting system*. (n.d.). Retrieved from www.polys.me
22. Solidity — Solidity 0.8.0 documentation. (n.d.). Retrieved January 26, 2021, from https://docs.soliditylang.org/en/v0.8.0/
23. Springall, D., Finkenauer, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., & Halderman, J. A. (2014). Security analysis of the estonian internet voting system. *Proceedings of the ACM Conference on Computer and Communications Security*, 703–715. https://doi.org/10.1145/2660267.2660315
24. Swan, M. (n.d.). *Blockchain : blueprint for a new economy*.
25. The Online Voting Platform of The Future - Follow My Vote. (2020). Retrieved January 12, 2020, from https://followmyvote.com/
26. Tso, R., Liu, Z.-Y., & Hsiao, J.-H. (2019). Distributed E-Voting and E-Bidding Systems Based on Smart Contract. *Electronics*, *8*(4), 422. https://doi.org/10.3390/electronics8040422
27. Voting results in detail. (2019). Retrieved January 26, 2020, from https://rk2019.valimised.ee/en/voting-result/voting-result-main.html
28. Weber, I., Lu, Q., Tran, A. B., Deshmukh, A., Gorski, M., & Strazds, M. (2019). *A Platform Architecture for Multi-Tenant Blockchain-Based Systems*. Retrieved from http://arxiv.org/abs/1901.11219

29. Yli-Huumo, J., Ko, D., Choi, S., Park, S., & Smolander, K. (2016). Where Is Current Research on Blockchain Technology?—A Systematic Review. *PLOS ONE*, *11*(10), e0163477. https://doi.org/10.1371/journal.pone.0163477
30. Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. *Proceedings - 2017 IEEE 6th International Congress on Big Data, BigData Congress 2017*, 557–564. https://doi.org/10.1109/BigDataCongress.2017.85

Priedai

A Priedas. „Voting“ išmaniosios sutarties programinis kodas

```
contract Voting {
    struct Voter {
        bool hasRightVote;
        bool voted;
        uint vote;
    }

    struct Proposal {
        bytes32 name;
        uint voteCount;
    }

    address public owner;
    bool public complete;
    string public title;
    string public endDate;
    uint public totalVotes;

    mapping(address => Voter) public voters;

    Proposal[] public proposals;

    function Voting(string votingTitle, bytes32[] proposalNames, string votingEndDate,address
votingOwner) public {
        owner = votingOwner;
        title = votingTitle;
        endDate = votingEndDate;
        voters[owner].hasRightVote = true;
        complete = false;

        for (uint i = 0; i < proposalNames.length; i++) {
            proposals.push(Proposal({name: proposalNames[i], voteCount: 0}));
        }
    }

    modifier activeVote() {
        require(!complete);
        _;
    }

    modifier restricted() {
        require(msg.sender == owner);
        _;
    }

    modifier canVote() {
        Voter storage sender = voters[msg.sender];
        require(sender.hasRightVote);
        require(!sender.voted);
        _;
    }

    function giveRightToVote(address voterAddress) public restricted {
        require(!voters[voterAddress].voted);
        require(!voters[voterAddress].hasRightVote);
        voters[voterAddress].hasRightVote = true;
    }

    function vote(uint id) public canVote activeVote {
        Voter storage sender = voters[msg.sender];
        sender.voted = true;
        sender.vote = id;
        totalVotes++;
        proposals[id].voteCount++;
    }

    function winningProposal() public view returns (uint) {
        uint winningVoteCount = 0;
        uint foundWinningProposal = 0;
        for (uint p = 0; p < proposals.length; p++) {
            if (proposals[p].voteCount > winningVoteCount) {
                winningVoteCount = proposals[p].voteCount;
                foundWinningProposal = p;
```

```
            }
        }
        return foundWinningProposal;
    }

    function winnerName() public view returns (bytes32) {
        return proposals[winningProposal()].name;
    }

    function finalize() public restricted {
        complete = true;
    }

    function getProposals() public view returns (Proposal[]) {
        return proposals;
    }
}
```

B Priedas. „VotingFactory“ išmaniosios sutarties programinis kodas

```
contract VotingFactory {
    address[] public deployedVotings;

    function createVoting(string votingTitle, bytes32[] proposalNames, string endDate) public {
        Voting voting = new Voting(votingTitle, proposalNames, endDate, msg.sender);
        deployedVotings.push(voting);
    }

    function getDeployedVotings() public view returns (address[]) {
        return deployedVotings;
    }
}
```

C Priedas. Elektroninio balsavimo sistemos programinis kodas

https://github.com/mindaugas16/e-voting