LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

BLOKĶĒDES TEHNOLOĢIJA, TĀS PIELIETOJUMS UN POTENCIĀLS

BAKALAURA DARBS

Autors: **Rainers Eduards Strautiņš**
Studenta apliecības Nr.: rs14022
Darba vadītājs: Dr. sc. administr. Imants Gorbāns

RĪGA 2018

ANOTĀCIJA

Darba mērķis ir blokķēdes tehnoloģijas un tās potenciālo pielietojuma iespēju izpēte. Darba ietvaros tika veikta blokķēdes tehnoloģijas un šajā sfērā esošo vadošo tehnoloģisko risinājumu apskate, eksistējošo tehnoloģijas ierobežojumu un iespēju analīze. Papildus tika veikta padziļinātāka Ethereum blokķēdes platformas apskate, jo pēc veiktās izpētes, tā tika izvirzīts par mūsdienu blokķēdes sfērā esošo tehnoloģisko līderi.

Pēc iegūto izpētes rezultātu analīzes darba praktiskajā daļā tika veikta uz Ethereum balstītas decentralizētas lietojumprogrammas prototipa un tā pamatā esošā viedā līguma izstrāde, kas tehniski veic automatizētu auto stāvlaukumu nomas realizāciju.

Atslēgvārdi: Blokķēde, viedais līgums, Ethereum, kriptovalūta, decentralizēta lietojumprogramma.

ABSTRACT

BLOCKCHAIN TECHNOLOGY, ITS USE AND POTENTIAL

The main goal of this paper is to research blockchain technology and its potential use. The research contains technical review of the blocchain technology, overview of the current leaders in this technology and an analysis of the curently existing limitations and possibilities. In addition there was also done more in depth overview of the Ethereum blockchain platform, because it is curently the technological leader in the blockchain field.

After the research result analysis there was created decentralized application prototype based on smart contract on the Ethereum blockchain, that can be used to automate the process of renting a car parking space.

Keywords: Blockchain, smart contract, Ethereum, cryptocurrency, decentralized application.

SATURS

APZĪMĒJUMU SARAKSTS

Bitcoin – ir vienādranga apmaksas sistēma un kriptovalūta, kas balstīta uz atvērtā pirmkoda protokolu un izmanto publisku transakciju žurnālu.

Blokķēde – ir izkliedēta datubāze, kas ir nepārtraukti augošs sakārtotu ierakstu jeb bloku saraksts, kuros esošie datu ieraksti ir aizsargāti pret viltošanu un pārrakstīšanu izmantojot kriptogrāfijas tehnoloģiju.

Bloks – datu ierakstu kopums.

DAO – decentralizēta autonoma organizācija.

DAPP – decentralizēta lietojumprogramma.

Decentralizēta datubāze – datubāze, kuras datus vienlaicīgi uztur vairākas kopējai sistēmai pieslēgtas ierīces.

DDoS uzbrukums - Kiberuzbrukuma veids, kas tiek veikts, lai traucētu vai pilnībā pārtrauktu tīmekļa vietnes vai servera darbību.

Ether – Ethereum platformas pamatā esošā kriptonauda

Ethereum –blokķēdes platforma, kurā ir integrēta Tjuringa-pilnīga programmēšanas valoda, kas ļauj veidot viedos līgumus.

JavaScript – skriptu valoda, kas balstīta uz prototipu koncepta.

Kriptovalūta – ir vērtību nesošs maiņas līdzeklis, kas izmanto kriptogrāfijas tehnoloģiju, lai aizsargātu darījumus pret viltošanu un pārrakstīšanu.

Marķieris – viedajos līgumos realizējama vērtība, kas var tikt realizēta kā kriptovalūta vai ka kāda cita abstrakta struktūru, ko ir iespējams izvietot blokķēdes tīklā. Ar to savstarpēji var mainīties tīkla lietotāji.

Merkle root – koka datu struktūra kriptogrāfijā, kura esoša mezglos glabājas visu apakšmezglu jaucējkods

Nonce – Ethereum konta skaitlis

Solidity – Ethereum platformas programmēšanas valoda, kas ļauj veidot

viedos līgumus.

Tjūringa pilnīga valoda – valoda, kas funkcionāli ļauj pilnībā simulēt visas Tjūringa mašīnas iespējas.

Viedais līgums – ir datoru protokols, kas nodrošina līgumu noslēgšanu un automatizētu izpildi starp divām pusēm bez trešās puses iesaistes.

Lightning un Raiden tīkls - decentralizēts tīkls, kurš ļauj veidot un veikt maksājumu apstiprināšanu nekavējoties.

IEVADS

Blokķēdes tehnoloģija mūsdienās aizvien vairāk gūst strauju popularitāti cilvēku vidū un ļoti daudzi informācijas tehnoloģiju nozares eksperti uzskata, ka šī tehnoloģija ir spējīga revolucionarizēt daudzas tādas mūsdienu sfēras kā piemēram banku sistēmu, izklaides un datorspēļu industriju, enerģijas industriju, apdrošināšanas sistēmas, loģistikas un piegādes ķēžu sistēmas izmantojot viedos līgumus un decentralizāciju.

Blokķēdi iespējams uzskatīt par jaunu tehnoloģiju, jo tā aizsākusies 2009.gadā, kad joprojām anonīmā persona vai cilvēku grupa, kas izmantoja segvārdu Satoshi Nakomoto, izlaida klajā Bitcoin kriptovalūtu [1]. Šī tehnoloģija strauji kļūst populāra un piesaista jaunus lietotājus pateicoties tās uzticamībai, kas tiek nodrošināta glabājot publiskā veidā visus veiktos datu ierakstus ar veiktajiem kriptovalūtas pārskaitījumiem datu blokos, kas ir savstarpēji saistīti viens ar otru un kuru pievienošanu, saturošo datu pārbaudi un apstiprināšanu veic blokķēdes tīkla lietotāji. Kopš 2009. gada šī tehnoloģija ir strauji attīstījusies un tajā ir veikta viedo līgumu integrācija ar kuru palīdzību iespējams veidot automatizētas decentralizētas sistēmas.

Pirmo reizi viedo līgumu ideja tika minēta 1996.gadā, kur Niks Szabo tos aprakstīja kā juridisko līgumu alternatīvu, kas eksistētu digitālā veidā un kur daļa no tā būtu realizēta kā automatizēta sistēma. Tieši Vitaļika Buterina izveidotā Ethereum platforma, kurā ir iebūvēta Tjuringa pilnīga programmēšanas valoda, deva iespēju jebkuram izstrādātājam veidot decentralizētas programmas un padarīja viedos līgumus populārus.

Darba autors izvirzīja mērķi veikt blokķēdes tehnoloģijas sfēras un tās pielietojuma iespēju izpēti un vadoties pēc iegūtajiem rezultātiem izvirzīt vadošās potenciālās pielietošanas sfēras. Vadoties pēc veiktajā izpētē iegūto rezultātu analīzes veikt decentralizētas lietojumprogrammas prototipa izstrādi, ko būtu iespējams pielietot kādam konkrētam gadījumam, kā arī veikt reālo viedā līguma tehniskās realizācijas apraktu.

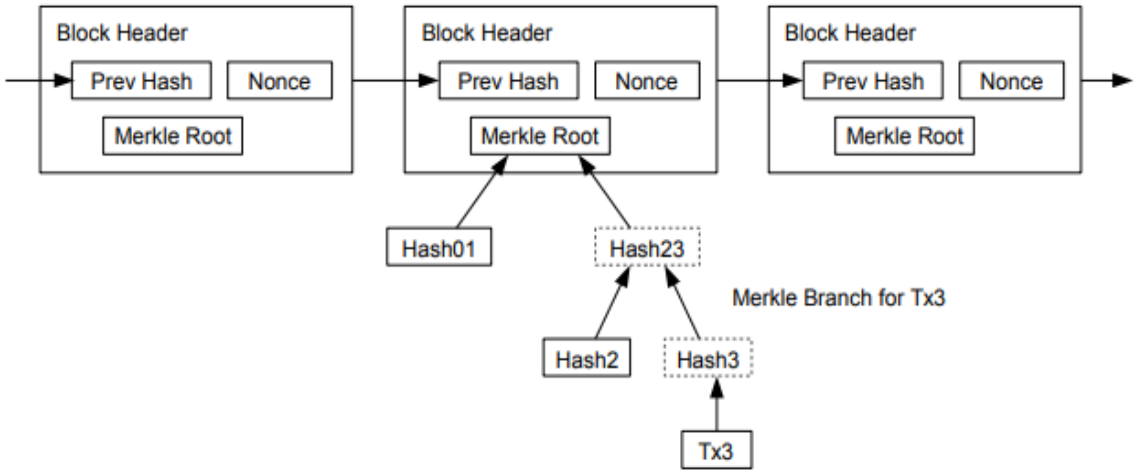
Darba realizēšanas laikā kā pētniecības metodes tika izmantotas pieejamās literatūras analīze un prototipēšana.

1. BLOKĶĒDES TEHNOLOĢIJA

Sākotnēji blokķēdes tehnoloģija tika izveidota ar mērķi pildīt alternatīvas un uzticamas elektroniskās naudas funkcionalitāti, bet mūsdienās šī tehnoloģija ir strauji attīstījusies un to ir iespējams izmantot daudz plašāk kā tikai vienkāršu norēķinu līdzekli, pateicoties šajā tehnoloģijā implementētajai viedo līgumu veidošanas funkcionalitātei.

1.1 Vispārējā uzbūves struktūra un darbība

Blokķēdes pamatā, kā jau pēc nosaukuma iespējams noprast, eksistē ķēde jeb virkne, kas sastāv no datu blokiem. Jaunizveidotie bloki tiek pievienoti virknes beigās un katrs bloks ir saistīts ar iepriekšējo bloku virknē izmantojot bloka aprakstošo jaucējfunkcijas izpildes laikā iegūto rezultātu. Katrā blokā ir galvene kurā glabājas bloku aprakstošie dati un ķermenis, kur atrodas informācija par blokā iekļautajiem apstiprinātajiem darījumiem.



1.1. att. **Bitcoin blokķēdes uzbūves arhitektūra [1]**

Būtība blokķēdi var uzskatīt par decentralizētu datubāzi, kuras pamatā ir atklātība, publiska pieejamība, aizsardzība pret saglabāto datu ļaunprātīgu manipulāciju un pārrakstīšanu. Tieši decentralizācijas īpašība piešķir blokķēdei nepieciešamo drošību pret ļaunprātīgu datu izmaiņām, jo to uztur visi tās lietotāji jeb blokķēdes mezgli, kas kopā veido vienotu datoru tīklu. Katra lietotāja, kas pieslēdzies blokķēdes rakšanas tīklam, datorā glabājas pilna vai atsevišķos gadījumos daļēja konkrētās blokķēdes kopija, tādējādi blokķēde veiksmīgi darbosies kamēr eksistēs vismaz viens aktīvs blokķēdes mezgls. Augot lietotāju skaitam blokķēdes tīkls kļūst arvien nostiprinātāks un drošāks. Izveidotajā blokķēdes mezglu tīklā visiem lietotājiem tiek piešķirtas vienlīdzīgas tiesības.

1.2 Blokķēdes lietotāji

Blokķēdes lietotāji iedalās divos veidos:

- Parasts lietotājs - kas veido jaunus ierakstus veicot valūtas pārskaitījumus
- Kriptovalūtas racēji jeb tā saucamie mezgli – veic jauno datu ierakstu apkopošanu, pārbaudīšanu un to ierakstīšanu datu blokos. Lai veiktu minētas darbības lietotājam ir nepieciešams kļūt par blokķēdes tīkla mezglu un glabāt lokālo blokķēdes kopiju.

Kriptovalūtas racēja uzdevums ir apkopot jaunos ierakstus un ģenerēt jaunā bloka galvenē esošo jaucējkodu, kuram ir jāatbilst tai brīdi tīklā esošajiem nosacījumiem. Gadījuma, ja pirmais uzģenerētais jaucējkods neatbilst noteiktajam blokķēdes tīkla sarežģītības prasībām racēja lietotājs izmaina bloka galvenē iekļaujamā nonce lauka vērtību un veic atkārtotu jaucējkoda funkcijas izpildi un ģenerētā rezultāta pārbaudi. Ja racējam veiksmīgi izdodas ģenerēt jaucējkodu, kas atbilst prasībām, jaunais bloks tiek izveidots un pārsūtīts tīklā, kur citi mezgli validē to un pievieno to klāt savai lokālajai blokķēdes kopijai. Līdz brīdim kamēr jaunie datu ieraksti nav pārbaudīti un ievietoti datu blokā, tie nav uzskatāmi par uzticamiem, jo tie var būt nekorekti un tikt atcelti.

Kā trešo lietotāja veidu, kas eksistē tikai blokķēdēs, kurās ir integrēta Tjuringa-pilnīga programmēšanas valoda, kā piemēram Ethereum blokķēdē [2], var minēt

- Viedais līgums – lietotājs, kas realizēts izmantojot programmatūras kodu. Viedais līgums ir spējīgs veikt darbības ar tam pieejamajiem kriptovalūtas līdzekļiem balstoties uz tajā ieprogrammēto loģiku.

1.3 Blokķēdes datu bloks

Blokķēdes bloks sastāv no divām daļām, kur pirmā ir galvene, kas satur bloka aprakstošos metadatus, kuros Bitcoin gadījumā ietilpts tādi dati kā esošā un iepriekšējā bloka jaucējkods, izveidošanas laiks, merkle root datu struktūra, kas apraksta blokā ķermenī iekļautos kriptovalūtas darījumu ierakstus, nonce lauks un esošais tīkla sarežģītības līmenis. Šīs ir arī uzskatāmas kā vispārīgās bloka sastāvdaļas, kas ietilpts vairākuma dažādu veidu blokķēžu blokos, bet bez pieminētā blokam iespējams saturēt papildus arī katrai blokķēdei specifisku informāciju.

1.4 Jaucējkods

Katram blokam eksistē tā identificējošais jaucējkods. Jaunu bloku jaucējkods tiek aprēķināts izmantojot blokā iekļautos datus un iepriekšējā bloka jaucējkodu, tādējādi katra bloka atslēgā savā veidā tiek iešifrētas atsauce uz visiem iepriekšējiem blokiem. Izmantojot jaucējkodu nav iespējams atrast sākotnējo datu kopu no kuras tas tika ģenerēts, bet izmantojot datu kopu iespējams pārbaudīt, vai tā atbilst ģenerētajam jaucējkodam. Bloka jaucējkods nodrošina aizsardzību pret manipulācijām ar blokā esošajiem datiem, jo veicot izmaiņas blokā esošajos datos tiks izmainīts bloka identificējošais jaucējkods, kas veidojas attiecīgi no bloka saturošajiem datiem. Situācijā, ja tiktu mēģināts izmainīt kādā blokā esošos datus, tiktu izmainīts jaucējkods, kas prasītu veikt izmaiņas arī visos nākošajos blokos, jo to jaucējkodos tiek ietverta informācija par iepriekšējo bloku jaucējkodiem. Šī šifrēšanas aizsardzība nodrošina pret iespējamajiem bloku datu izmaiņu mēģinājumiem, secības maiņām vai to dzēšanas.

Bitcoin blokķēdē jaucējkoda ģenerēšanai tiek izmantota 256-bit jaucējfunkcija [3].
Vēl citas mūsdienās populāras blokķēdēs izmantotas jaucējfunkcijas [4] ir :

- Ethhash
- CryptoKnight
- Equihash
- Scrypt

1.5 Publiskās un privātās atslēgas

Lai veiktu kriptovalūtas glabāšanu, saņemšanu un sūtīšanu ir nepieciešamais digitālais kriptovalūtas maks [5], kurš pēc uzbūves sastāv no :

- Publiskās atslēgas - var tik uzrādīta citiem tīkla lietotājiem. Publiskā atslēga neietver sevi nekādu personīgo informāciju par maka lietotāju un tādā veidā ir anonīma, neatklājot maka īpašnieka identitāti. Tā tiek izmantota, lai pierādītu, ka darījuma veikšana no konkrētās adreses ir veikta korekti.
- Privātā atslēga – ir zināma tikai maka īpašniekam un tikai izmantojot to ir iespējams veikt darbības ar maku un tajā esošo kriptovalūtu. Privātā atslēga tehniski tiek izmantota lietotāja veikto darbību parakstīšanai, jo savā ziņā strādā kā parole, bez kuras nav iespējam veikt darījumus. Izmantojot privāto atslēgu ir iespējams noskaidrot attiecīgā maka publisko atslēgu un adresi, bet no publiskās atslēgas nav iespējams noskaidrot privāto atslēgu.

- Maka adrese – tiek izmantota kriptovalūtas saņemšanai. Brīdī kad kāds cits lietotājs vēlas nosūt kādam citam lietotājam kriptovalūtas pārskaitījumu, tas saņēmēja adresē norāda ar jaucējkodu apstrādātu saņēmēja publisko atslēgu, šīs darbības rezultātā tiek iegūta kriptovalūtas maka adrese.

Pēc uzbūves gan privātā, gan publiskā atslēga ir lieli integer tipa skaitļi [6], bet tā kā šie skaitļi ir ļoti lieli, tad to saīsinātai reprezentēšanai izmanto tā saucamo WIF jeb 'Walet import format', kas satur gan burtus, gan ciparus.

1.6 Viedie līgumi

Pirmo reizi viedo līgumu ideja tika minēta 1996.gadā, kur Niks Szabo tos aprakstīja kā juridisko līgumu alternatīvu, kas eksistētu digitālā veidā un kur daļa no tā būtu realizēta kā automatizēta sistēma. Tieši Vitaļika Buterina izveidotā Ethereum platforma, kurā ir iebūvēta Tjuringa pilnīga programmēšanas valoda, deva iespēju jebkuram izstrādātājam veidot decentralizētas programmas un padarīja viedos līgumus populārus.

Viedais līgums ir datu protokols, kura mērķis ir nodrošināt līgumu noslēgšanas funkcionalitāti un automatizētu izpildi starp divām iesaistītām pusēm bez trešās puses iesaistīšanas. Tas būtībā ir datorprogramma, kas tiek glabāta uz blokķēdes. Brīdī, kad viedais līgums tiek saglabāts blokķēdē to vairs nav iespējams rediģēt un tas ir visiem publiski pieejams.

1.7 Publiskās un privātas blokķēdes

Blokķēdes pamatideja ir būt pilnībā decentralizētai sistēmai, bet gadījumā, ja tehnoloģijas izmantošana nepieciešama uzņēmuma iekšējām vajadzībām, tehniski ir arī iespējams veikt blokķēdes pieeju ierobežošanu tādējādi sniedzot lielāku kontroli uzņēmumam, kas pārvaldība blokķēdi iekšējā tīklā.

Šī iemesla dēļ bloķēdes tehnoloģiju ir iespējams iedalīt 3 lielās kategorijās, kur katrai no tām ir dažādi realizētie pieejas līmeņi.

Tā iedalās:

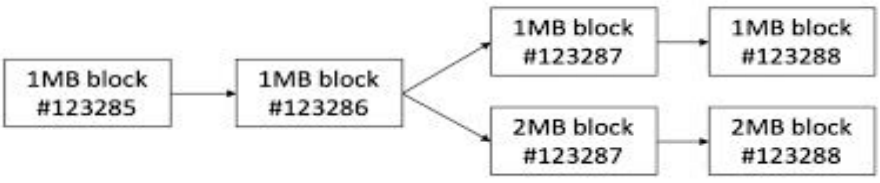
- Publiskais blokķēdes - šis veids ir visiem vislabāk zināmais un tradicionālais veids, kur visa glabātā informācija ir pieejama jebkuram lietotājam pasaulē un katram lietotājam ir iespējams veikt jaunu datu ierakstu veidošanu. Visiem pazīstamākās Bitcoin un Ethereum kriptovalūtas arī pieder pie publisko blokķēžu veida. Lai

novērstu blokķēdes pārvaldes procesu centralizāciju, tajā ir iestrādāta atlīdzības sistēma kurā lietotāji piedalās jaunos datu ierakstu verifikācijā un jaunu bloku veidošanā, kas tiek pievienoti pie blokķēdes, par to saņemot atlīdzību konkrētās kriptovalūtas veidā, tādējādi piesaistot jaunus lietotājus blokķēdes tīklam.

- Konsorcijas blokķēdēs – tajās eksistē piekļuves ierobežojumi, kur procesu pārvaldi veic iepriekš izvēlēta mezglu kopa. Šādu blokķēdi iespējams uzskatīt par daļēji decentralizētu un tā būtu piemērota izmantošanai banku un privātu uzņēmumu sistēmās, kur rakstīšanas iespējas būtu piešķirtas konkrētiem banku vai uzņēmumu mezgliem, bet lasīšanas pieejas visiem lietotājiem.
- Privātā blokķēde – pēc uzbūves ir pati noslēgtākā un centralizētākā, jo tajā rakstīšanas iespējas ir tikai kādai konkrētai organizācijai vai personai un lasīšanas pieejas var būt atstātas publiskas vai arī ierobežotas konkrētiem lietotājiem. Šāda veida blokķēde ir piemērota lietošanai uzņēmuma iekšienē. Veicot šāda veida piekļuves ierobežošanu blokķēde zaudē daudzās no tās pozitīvajām īpašībām, kas piešķir tai uzticamību un drošību, tādējādi principā neievērojot blokķēdes pamatideju kur tā ir pilnībā decentralizēta sistēma.

1.8 Blokķēdes zarošana un uzlabojumu ieviešana

Bitcoin blokķēde balstās uz tā kopienas lietotājiem, kas to lieto. Lai gan tas tika izveidots 2009. gadā, kopš tā laika kopienas lietotāji ir veikuši un publicējuši vairākas izmaiņas Bitcoin protokolā ar mērķi uzlabot tā darbību un efektivitāti. Gadījumos, kad kopiena nespēj vienprātīgi akceptēt piedāvātās izmaiņas, rodas situācija, kad kopiena sadalās, kur daļa no tās atbalsta jaunās protokola izmaiņas un daļa vēlās palikt pie vecā protokola. Šādā gadījumā tiek veikts tā sauktais zarošanās process, kura rezultātā no vienas blokķēdes rodas jauna, alternatīva blokķēde ar mazliet vai ļoti atšķirīgām īpašībām, bet vieniem un tiem pašiem pagātnes ierakstiem. Viena no populārākajām blokķēdēm, kas radusies zarošanas rezultātā ir Bitcoin Cash, kas tika atzarota 2017.gada vidū no oriģinālās Bitcoin blokķēdes.

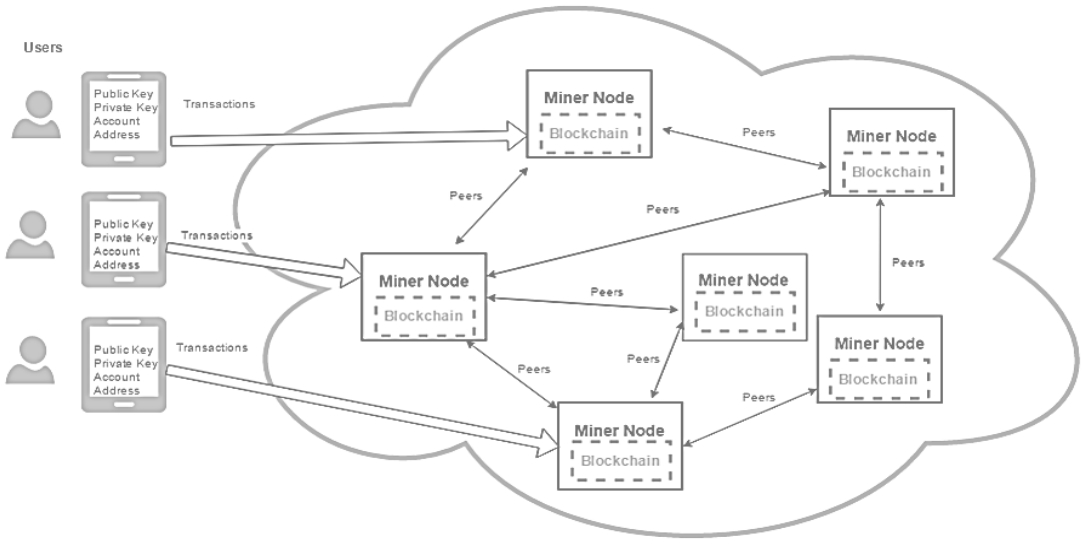


1.2. att. **Cietās zarošanās, kas izmaina maksimālo bloka izmēru no 1MB uz 2MB**

Blokķēdes zarošanu var iedalīt divos veidos [7], kur pirmais ir cietās zarošanas veids un otrais ir mīkstās zarošanas veids. Cietā zarošanās (Angliski – Hard Fork) rezultātā blokķēdē tiek ieviestas tādas izmaiņas kuru dēļ blokķēdes mezgliem veicot bloku validāciju jaunie bloki tiks uzskatīti par nederīgiem, tāpēc visiem mezgliem ir nepieciešams atjaunot to sistēmas, lai blokķēdes sistēma varētu vienprātīgi turpināt apstiprināt un validēt jaunu bloku pievienošanu blokķēdei, ja šadā gadījuma eksistē vairāki mezgli, kas neveic jauno izmaiņu ieviešanu, tad var notikt vienas blokķēdes sadalīšanās divās blokķēdēs, kur viena no tām strādās pēc vecajiem principiem, bet otra no tām pēc jaunajiem. Mīkstais zarošanas veids (Angliski – Soft Fork) ir tādu jaunu izmaiņu ieviešana, kas joprojām mezglu starpā tiek veiksmīgi apstiprinātas balstoties uz vecajiem noteikumiem. Mīkstais zarošanās veids ir tādu izmaiņu ieviešana, kas ir atpakaļ saderīga ar iepriekšējo blokķēdes stāvokli.

1.9 Blokķēdes mezgli

Blokķēdes mezgls ir svarīga sastāvdaļa kopējā blokķēdes tīkla uzturēšanā, jo tajā tiek glabāta lokālā blokķēdes pilna kopija [8], kā arī papildus tam mezgls veic sistēmā veikto darījumu apstiprināšanu un jaunu bloku veidošanu un pievienošanu blokķēdei, par to saņemot pretī kompensāciju kriptovalūtas veidā. Jo vairāk mezglu eksistē blokķēdes tīklā, jo stiprāka kļūst kopējā sistēma. Mezgls veic jauno darījumu apstiprināšanu izmantojot vienprātības protokolu (Angliski – Consensus protokol), kas savā būtībā definēti noteikumi un izpildāmās darbības, ko nepieciešams izpildīt, lai veiktu darījumu pārbaudi un veiksmīgu jaunu bloku veidošanu. Visiem vienas blokķēdes sistēmā esošajiem mezgliem ir jāizmanto viens un tas pats vienprātības protokols, lai tās savstarpēji būtu saderīgas un varētu darboties kā vienota sistēma.



1.3.att **.Blokķēdes mezglu un parasto lietotāju attiecību vizualizācija**

Kā populārākos vienprātības protokolus [9] var minēt:

- Darba pierādījums (Angliski – Proof of work) – kuru izmanto populārākās kriptovalūtas Bitcoin un Ethereum. Procesu kura rezultātā iegūst darba pierādījumu sauc par kriptovalūta rakšanu kuras laikā tiek veikta adekvātas jaucējfunkcijas meklēšana
- Likmes pierādījums (Angliski – Proof of stake) – šis pierādījuma protokols nebalstās uz datora jaudas izmantošanu, jo šajā procesā netiek veikta jaunu kriptovalūtas vienību iegūšana, jo visa pieejamā kriptovalūta jau eksistē. Tā vietā, lai piedalītos šajā darījumu validācijas procesā ir nepieciešams investēt konkrētajā blokķēdē, jo bloka veidošanas uzdevums tiek gadījuma veidā piešķirt kādam no sistēmas lietotājiem, kur lielāka iespēja saņemt šo uzdevumu ir tam lietotājam, kura kontā atrodas vairāk attiecīgās kriptovalūtas. Šeit par veikto bloka pievienošanu blokķēdei, kā kompensācija tiek saņemta nodeva par veiktajiem darījumiem. Arī Ethereum nākotnē plāno pāriet uz šo pierādījuma protokolu.

2. BLOKĶĒDES PIELIETOJUMS UN POTENCIĀLS

Blokķēdes pamatīpašības ir decentralizācija, datu ierakstu atklātība, publiska pieejamība un aizsardzība pret ļaunprātīgām datu izmaiņām. Mūsdienās blokķēde vairākumā gadījumu tiek izmantota kā alternatīva ierastajai fiat naudai, bet aizvien vairāk tiek veidoti jauni projekti, kuru tehnoloģiskā pamat ideja un funkcionalitāte izmanto blokķēdei piemītošās īpašības un viedo līgumu sniegtās iespējas.

Tieši Bitcoin aizsāka blokķēdes tehnoloģijas revolūcijas laikmetu. Sākotnēji blokķēdes pamatā bija tikai kriptovalūta, kura tika izmantota kā alternatīva fiat valūtai, bet laikam ejot tika izveidoti tādi jauni blokķēžu protokoli kā Ethereum, kuros papildus pamatā eksistējošajai kriptovalūtai tika implementētas Tjuringa-pilnīga programmēšanas valoda, kas deva iespēju veidot viedos līgumus, kas tiek saglabāti uz blokķēdes. Viedie līgumi sniedz izstrādātājiem iespēju veidot plašas funkcionalitātes lietojumprogrammas, kas darbojas decentralizētā veidā. Tieši minētās lietojumprogrammas sniedz šai tehnoloģijai iespējas revolucionarizēt vairākas industrijas ieviešot tajās decentralizāciju, automatizāciju un atbrīvošanos no trešajām personām [10].

Ņemot vērā, ka eksistē gan publiskās, gan privātās blokķēdes, šo tehnoloģiju iespējams izmantot netikai visiem interneta lietotājiem publiski pieejamā veidā, bet arī privātā veidā, piemēram uzņēmuma iekšējā tīklā, kur konkrētajai blokķēdei ir definētas piekļuves tiesības, kas ļauj definēt lietotājus, kas ir spējīgi tai piekļūt un darboties ar to. Lai gan privātās blokķēdes zaudē šai tehnoloģijai tik raksturīgās decentralizācijas īpašības un kļūst savā veidā centralizētas uzņēmuma iekšienē, arī šīm blokķēdēm piemīt potenciāls uzlabot datu apmaiņas un uzglabāšanas iespējas uzņēmumu un organizāciju iekšējās sistēmās.

2.1 Blokķēdes pielietošanas sfēras

Kā viens no pamata blokķēdes pielietošanas veidiem ir kriptovalūta, kas ir alternatīva ierastajai fiat valūtai, tādējādi tā pilda vērtības glabāšanas un apmaiņas funkcionalitāti publiskā un decentralizētā veidā. Pateicoties jaunajām viedo līgumu veidošanas iespējām, šo tehnoloģiju ir iespējams pielietot daudz vairāk veidos, ka tikai vienkāršu valūtu.

Lai veiksmīgi noteiktu konkrētas sfēras, kurās potenciāli iespējams izmantot šo tehnoloģiju, nepieciešams izpētīt kuras no mūsdienās eksistējošajām nozarēm varētu gūt labumu no decentralizācijas, publiskas atklātības un datu drošības, kas eksistē blokķēdes pamatā.

Blokķēdes tehnoloģija savā ziņā strādā kā publiska decentralizēta datubāze [11], kas sevī var glabāt aprakstošu informāciju par reālo pasauli un kurā saglabātos datus ir gandrīz neiespējams ļaunprātīgi manipulēt. Tajā saglabātos datus ir iespējams ātri atjaunot un tie ir publiski pieejami visām iesaistītajām pusēm, tādēļ šī tehnoloģija rada uzticību un uzlabo informācijas apmaiņas ātrumu un efektivitāti starp tās lietotājiem. Pateicoties šīm īpašībām to būtu iespējams izmantot nekustamo īpašumu sfērā, auto industrijā, loģistikā, autortiesību un patentu glabāšanā, banku sistēmu aizvietošanā, spēļu un azartspēļu industrijā, labdarības organizācijās, apdrošināšanas nozari, vēlēšanu sistēmās, ka arī citās nozarēs, kas balstās uz informācijas un vērtību apmaiņu.

Industrijas, kas izmanto blokķēdi, kļūst atklātākas, demokrātiskas, decentralizētas, efektīvas, drošas un pilnībā vai daļēji automatizētas pateicoties viedajiem līgumiem.

2.2 Blokķēdes evolūcija

Būtībā mūsdienu blokķēdes tehnoloģijā notiekošo evolūciju iespējams izdalīt 3 soļos, kur:

1. Kriptovalūta
2. Viedie līgumi
3. Decentralizētas lietojumprogrammas jeb DAPPs = viedie līgumi + lietotāja saskarne

Šie soļi kriptovalūtu lietotju vidū tiek dēvētas attiecīgi par blokķēdes 1.0 , 2.0 un 3.0 versijām.

2.2.1 Kriptovalūtas

Bitcoin pieder pie blokķēdes 1.0 versija un tas funkcionāli strādā kā kriptovalūta, kuru ir iespējams izmantot līdzekļu pārsūtīšanai citiem lietotājiem, citu kriptovalūtu pirkšanai vai kā norēķināšanās veids par produktiem un pakalpojumiem, kas pieņem Bitcoin.

Mūsdienās, lai izveidotu savu kriptovalūtu izstrādātājam ir trīs iespējas, pirmā ir veikt jaunas blokķēdes protokola programmēšanu, kas sevī ietvers jaunu kriptovalūtu, otrā ir veikt cieto zarošanos no kādas citas blokķēde tādējādi radot jaunu blokķēdi un kriptovalūtu vai trešais arī izmantot kādu no pieejamajām blokķēdēm kurai ir iespēja izveidot viedo līgumu ar kura palīdzību ir iespējams izveidot savu kriptovalūtas marķieri, kas balstīts uz kādu no pieejamajiem blokķēžu protokoliem un tiks glabāta uz dotā protokola blokķēdes. Šobrīd

populārākie pieejamie protokoli ar viedo līgumu veidošanas iespējām ir Ethereum, Neo, Lisk, EOS, Cardano [12].

2.2.2 Viedie līgumi

Mūsdienās blokķēžu tehnoloģija strauji attīstās un kā viens no jaunākajiem pievedumiem šajā tehnoloģijā ir tjuringa-pilnīgas valodas integrācija uz blokķēdes, kas ļauj veidot viedos līgumus. Tieši viedo līgumu sniegtās iespējas dod blokķēdei papildus funkcionālās iespējas, kas ļaus revolucionarizēt vairākas mūsdienu sfēras

Mūsdienās izmantojot viedos līgumus tiek veidoti jauni kriptovalūtu marķierus un decentralizētas lietojumprogrammas, kas tiek uzstādītas uz blokķēdes.

Populārākais šādas blokķēdes piemērs ir Ethereum kura pamatā eksistē Ether (ETH) kriptovalūta, ar kuru iespējams pildīt līdzīgas funkcijas kā ar Bitcoin valūtu, kā arī tajā ir integrēta viedo līgumu veidošanas iespējas.

Tjuringa-pilnīga programmēšanas valoda ļauj programmētājam realizēt jebkāda veida aprēķinus izveidotajās lietojumprogrammās [13]. Viedo līgumu pamatā ir spēja izpildīt transakcijas un pārvietot vērtības vadoties pēc tajos ieprogrammētas loģikas, tie arī ir spējīgi veikt vērtību uzglabāšanu tāpēc vienlaicīgi strādā arī kā autonoma maza datubāze.

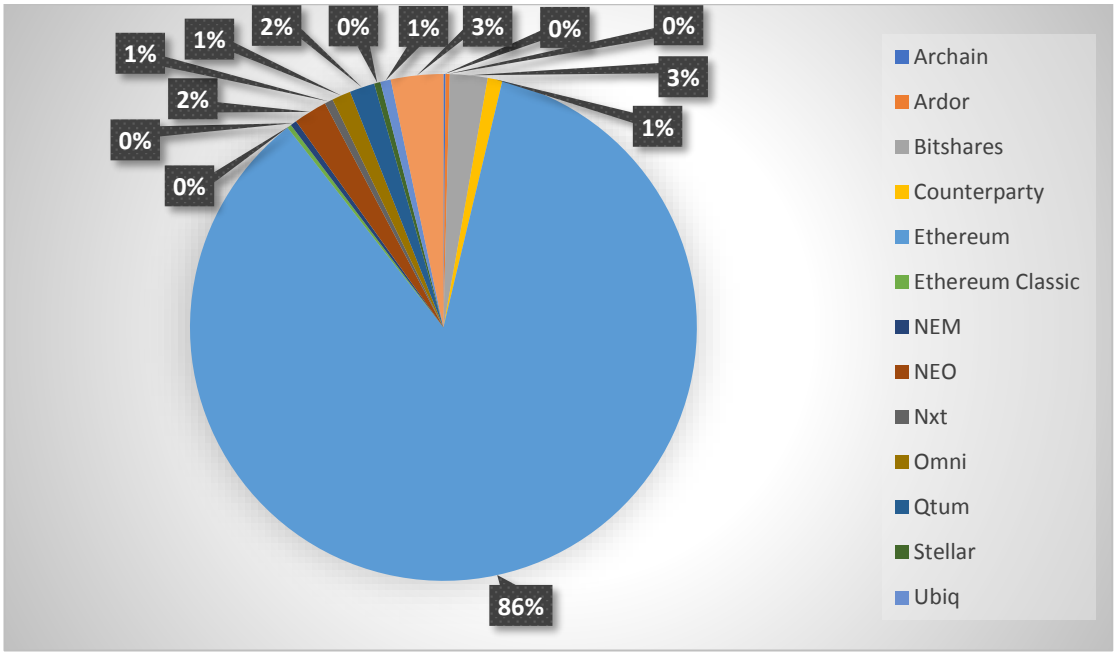
Viedie līgumi sniedz izstrādātajiem iespēju veidot savu kriptovalūtas marķierus, kas tehniski atšķiras no blokķēdes 1.0 apzīmētās kriptovalūtas, jo tā tiek radīta viedā līguma ietvaros, kas tiek izvietots uz jau eksistējošas blokķēdes pamata un ir atkarīga no tās.

Zemāk esošajā tabula redzami blokķēžu platformu nosaukumi, kas piedāvā viedo līgumu izstrādi un izstrādāto marķieru skaits, kas balstās uz https://coinmarketcap.com/tokens/views/all/ pieejamajiem datiem. Šajā vietnē tiek izvietotas kriptovalūtas, kas ir ievietotas tīmekļa vietnēs kurās iespējams tirgoties ar kriptovalūtām. Tādējādi tiek apskatīti dati par visperspektīvākajiem projektiem un platformām uz kurām tie tiek balstīti

2.1. tabula

Blokķēžu platformu, kas piedāvā viedo līgumu izveidi, salīdzinājums balstoties uz pieejamo veiksmīgo projektu skaitu.

Nosaukums	**Skaits**	**%**
Archain	1	>1%
Ardor	2	>1%
Bitshares	18	3%
Counterparty	7	1%
Ethereum	642	86%
Ethereum Classic	2	>1%
NEM	3	>1%
NEO	16	2%
Nxt	4	1%
Omni	9	1%
Qtum	12	2%
Stellar	3	0%
Ubiq	5	1%
Waves	25	3%
Kopā	**749**	



2.1. att. **2.1. tabulā esošo datu vizualizācija**

Pēc augstāk esošā apkopojuma var uzskatīt, ka Ethereum ir nepārspējams līderis jaunu kriptovalūtas marķieru veidošanā, jo no 749 uzrādītajiem marķieriem 642 ir izveidoti izmantojot Ethreum platformu, kas kopā ir 86%, otrajā vietā ierindojas Waves platforma ar 25 marķieriem jeb 3%. Jāņem vērā, ka dati ir no visveiksmīgāko projektu izlases, kas ir piesaistījuši jau ievērojamu skaitu lietotāju uzmanības un ar kuriem notiek aktīva tirgošanās. Vadoties pēc etherscan.io lapā esošās informācijas Ethereum tīklā pavisam ir izveidoti jau gandrīz 85000 ERC20 standarta kriptovalūtas marķieru. Izveidotie marķieri var tikt izmantoti kāda konkrēta projekta ietvaros, kā veids ar kuru iespējams norēķināties par kāda servisa piedāvātajiem pakalpojumiem vai tikt sākotnēji izmantoti investīciju vākšanas pasākumos jeb ICO, kur interesentiem ir iespējams iegādāties kriptovalūtas marķierus izmantojot populārākās kriptovalūtas kā Bitcoin vai Ethereum tādējādi finansiāli atbalstot kāda projekta realizāciju, kur tā realizācijas beigās tiks izveidota decentralizēta lietojumprogramma jeb DAPP, kurā būs iespējams izmantot pieejamos pakalpojumus izmantojot jau iegādāto kriptovalūtas marķierus.

Izmantojot viedos līgumus ir iespējams veidot dažāda veida lietojumprogrammas, kas var veikt darbības ar pašu Ether valūtu vai jau izveidotiem marķieriem, tā var darboties arī kā maza datubāze un glabāt sevi dažāda veida datus.

Lai gan Bitcoin nav uzrādīts augstāk minētajā tabulā, šobrīd šai blokķēdei ir pieejama nepieciešamā funkcionalitāte, lai izveidot primitīvus viedo līgumus, kuriem gan nav tik plašas funkcionālās iespējas, kā uz Ethereum veidotiem viedajiem līgumiem.

2.2.3 Decentralizētas lietojumprogrammas

Decentralizētas lietojumprogrammas jeb DAPPS tiek uzskatītas par blokķēdi 3.0, kas patiesībā pēc uzbūves ir viedais līgums, kas darbojas kopā ar lietotājam draudzīgu saskarni, kas visbiežāk tiek realizēta, kā tīmekļa vietne.

DAPPs ir iespējams iedalīt trīs veidos [14] :

1) Veic darbības tikai ar kriptovalūtām, to uzglabāšanu, pārskaitīšanu, maiņu vai pārdošanu.
2) Veic darbības ar kriptovalūtu balstoties uz no ārējiem resursiem saņemto informāciju, balstoties pēc kuras tiek pieņemti lēmumi, kā rīkoties ar pieejamajiem līdzekļiem
3) Decentralizētas lietojumprogrammas veidā realizēta decentralizēta autonoma organizācija, kurā pastāv pilnīga demokrātija un tajā esošo līdzekļu kontrole un izmantošana tiek veikta izmantojot sistēmā esošo lietotāju balsojuma rezultātus, kur nepieciešams iegūt noteiktu pārsvaru balsu, lai konkrētās darbības tiktu izpildītas

Zemāk izveidots blokķēžu platformu un uz to pamata radīto DAPP skaita apkopojums. Visām no jau iepriekš minētajām platformām, nebija iespējams uzrādīt aptuveno veiksmīgo DAAP lietojumprogrammu skaitu, jo netika atrasti nepieciešamie dati.

2.2. tabula

Blokķēžu platformu, kas piedāvā viedo līgumu izveidi, salīdzinājums balstoties uz pieejamo veiksmīgo decentralizēto lietojumprogrammu skaitu.

Nosaukums	Skaits	Avots
Ethereum	1511	https://www.stateofthedapps.com/
NEM	19	https://nem.io/community/projects/
NEO	42	http://ndapp.org/

Lai gan šie dati nav tik uzskatāmi, jo šāda veida datu apkopojumi internetā nav plaši atrodami, arī šeit ir novērojams Ethereum platformas izmantošanas popularitātes pārākums, kas stipri pārsniedz citās blokķēdēs veidoto decentralizētu lietojumprogramu skaitu.

Pēc definīcijas decentralizēta lietojumprogramma sastāv no viedā līguma un lietotāja saskarnes, kur tā vietā, lai lietotājs veiktu manuālu viedajā līgumā esošo funkciju izsaukšanu un parametru padošanu, tiek realizēta ērta lietotāja saskarne, kas ir spējīga komunicēt ar blokķēdi, līdzīgi kā citos alternatīvos gadījumos tas tiktu veikts ar datubāzi.

2.3 Mūsdienu blokķēdes tehnoloģijā esošās problēmas

Kā pirmo eksistējošo problēmu, kas arī mūsdienās ir vislielākais ierobežojums straujai blokķēdes tehnoloģijas masu adaptācijai, varētu minēt mērogojamības ierobežojumus, kas rodas no nepietīkami ātrā transakciju apstrādes ātruma. Bitcoin šobrīd ir spējīgs apstrādāt līdz 7 transakcijas sekundē, savukārt Ethereum apstrādā vidēji aptuveni 20 transakcijas sekundē, kas salīdzinoši ar Visa, kas spēj apstrādāt līdz par 25,000 transakciju sekundē, ir niecīgs skaitlis. Eksistē tādas jauni veidotas kriptovalūtas kā Ripple, kuras jau šobrīd spēj vienlīdzīgi konkurēt ar Visa transakciju apstrādes laiku, kā arī tiek strādāts pie tādiem risinājumiem kā Lightening tīkls [15], kas paredzēts Bitcoin transakciju apstrādes ātruma vairākkārtējai uzlabošanai un Raiden tīkls [16], kas paredzēts Ethereum transakciju apstrādes uzlabošanai līdz pat 100,000 transakcijām sekundē.

Tā kā mūsdienās blokķēdes tehnoloģija ir salīdzinoši jauna tad, tikai nesen valdības un atbildīgās institūcijas ir uzsākušas izskatīt šīs tehnoloģijas regulācijas iespējas un to ieviešanas procesa sākšanu. Šobrīd šī tehnoloģija netiek regulēta un uz tās tiek realizētas krāpnieciskas

darbības, kuru rezultātā no cilvēkiem tiek izkrāpta nauda, kā arī tiek veikta datu zagšana un kibernoziegumi ar mērķi iegūt savā īpašumā pēc iespējas vairāk kriptovalūtas līdzekļu

3. ETHEREUM BLOKĶĒDE

Ethereum šī darba rakstīšanas laikā ir otrā lielākā kriptovalūta vadoties pēc tirgus izmēra salīdzinājumiem. Ethereum izveidoja Vitaļiks Buterins.un tajā ir iebūvēta Tjuringa pilnīga programmēšanas valoda ar nosaukumu Solidity, kas dod katram izstrādātājam iespēju veidot viedos līgumus.

Darba rakstīšanas laikā pēc izpētīto reālo lietojuma gadījumu skaita Ethereum ir vispopulārākais un biežāk izmantotais blokķēdes protokols viedo līgumu izstrādei. Tāpēc darba ietvaros veikta padziļināta Ethereum platformas izpēte ar mērķi iepazīties ar doto platformu un izmantojot to veikt decentralizētas lietojumprogrammas prototipa un tā pamatā esošā viedā līguma izveidi

3.1 Ethereum darbības pamatprincipi

Ethereum tīklā eksistē trīs kontu veidi:

- Kriptovalūtas racēji jeb tā saucamie mezgli – veic jauno datu ierakstu apkopošanu, pārbaudīšanu un to ierakstīšanu datu blokos. Lai veiktu minētas darbības lietotājam ir nepieciešams kļūt par blokķēdes tīkla mezglu un glabāt lokālo blokķēdes kopiju
- Lietotāja konts – lietotājam ir iespējams saņemt un veikt kriptovalūtas pārskaitījumus. Glabāt Ether un citas ERC20 marķieru kriptvalūtas.
- Viedā līguma konts – kontu kontrolē tam piesaistīts viedā līguma kods. Lietotājam ir iespējams sazināties ar viedā līguma kontu izsaucot tajā esošās funkcijas.

Abi kontu veidi ir spējīgi izpildīt transakcijas Ethereum tīklā, kuras ļauj netikai veikt kriptovalūtas pārskaitījumus, bet arī pārsūtīt datus un izsaukt konkrētam viedajam līgumam esošās funkcijas. Transakciju aptiprināšanu un izpildi veic jau iepriekš aprakstītie racēj lietotāji, kuri par veiksmīgu transakciju pievienošanu blokķēdei saņem komisijas maksu, ko Ethereum vidē sauc par Gas. Pēc būtības Gas ir maza Ether kriptovalūtas vienība. Lietotājam veicot transakciju ir iespējams norādīt sākontējo un maksimāli izmantojamo Gas daudzumu. Transakcijas kurām ir lielāks uzstādītais Gas daudzums tiek prioritizētas Ethereum tīklā.

3.2 Ether kriptovalūta

Uz Ethereum blokķēdes eksistē Ether kriptovalūta, kura pamatā kalpo kā norēķina veids par veikto darījumu nodevām, kas tiek samaksātas mezgliem par darījumu apstiprināšanu. Konkrētās nodevas Ethereum sistēmā tiek dēvētas par GAS vienībām. Ether ari tiek izmantots

kā kriptovalūta ar kuru ir iespējams tirgoties pret citām kritovalūtam vai iekš blokķēdes radītajiem kriptovalūtu marķieriem.

Ether iedalās sekojošajās apakšvienībās [17], kur katra no tām apzīmē mazākas Ether vienības.

3.1. tabula

Ether kriptovalūtas sadalījums oficiāli pieņemtajos mazākos nominālos un to nosaukumu sadalījums

Nosaukums	Wei Vērtība	Wei daudzums
wei	1 wei	1
Kwei (babbage)	1e3 wei	1,000
Mwei (lovelace)	1e6 wei	1,000,000
Gwei (shannon)	1e9 wei	1,000,000,000
microether (szabo)	1e12 wei	1,000,000,000,000
milliether (finney)	1e15 wei	1,000,000,000,000,000
ether	1e18 wei	1,000,000,000,000,000,000

Kā mazākā apzīmējamā Ether vienība ir Wei, kas ir 1 / 1,000,000,000,000,000,000 no Ether.

3.3 Transakcijas

Ethereum blokķēdē transakcijas veido gan tīkla lietotāji, gan viedie līgumi. Izmantojot transakcijas ir iespējams netikai veikt kriptovalūtu pārskaitījumu, bet arī pārsūtīt datus. Ja viedais līgums saņem transakciju, tas ir spējīgs atbildēt un atgriezt vērtību uz šo izsaukumu. Viedā līguma pēc tehniskā risinājuma būtu iespējams salīdzināt ar parastu funkciju izpildi, kas saņemot ieejā datus izpilda noteiktās darbības un atgriež rezultātu.

Transakcija pēc uzbūves satur datus par sūtītāju un saņēmēju, nosūtīto Ether daudzumu un citus datus. Papildus ir iespējams konfigurēt izmantojamā GAS daudzuma norādīšanu, ko izmantot, lai apstiprinātu transakciju un samaksātu racēju lietotājiem.

Tjuringa-pilnīgā programmēšanas valodā pastāv iespēja nonākt situācijā kad viedā līguma iekšienē izveidojas mūžīgs cikls, kura izpildes mēģinājuma laikā tiks izmantots viss atvēlētais GAS daudzums, tādā gadījuma tiktu atsauktas visas veiktās izmaiņas, bet GAS vērtība, kas tika samaksāta racēja lietotājam par veikto darbu netiek atgriezta atpakaļ. Ja

transakcijas izpilde ir veiksmīgi pabeigta un netika izmantots viss atvēlētais GAS daudzums, tad pārpalikums tiek atgriezts sūtītājam.

Šāda veida sistēma arī nodrošina labu aizsardzību pret DDOS uzbrukumiem, jo par katru nepieciešamo transakciju ir nepieciešams maksāt GAS.

Šobrīd Ethereum blokķēde spēj veikt 20 darījumu sekundē, bet tas nav pietiekoši ātri, lai šo tehnoloģijai varētu realizēt masu adoptāciju, tāpēc tiek plānots realizēt Raiden tīklu, kas ļaus Ethereum veikt līdz pat 100000 darījumu sekundē.

3.4 Viedie līgumi

Ethereum blokķēdē, lai veiktu viedo līgumu uzstādīšanu uz blokķēdes par darījumu nepieciešams maksāt nodevu, ko apzīmē par GAS. Šī nodeva savā būtībā ir mazāka Ethereum vienība. Lietotājam ir iespējams pirms darījuma veikšanas norādīt maksimāli izmantojamā GAS daudzumu un cenu par to. Jo augstāka tiek norādīta cena par GAS, jo vairāk šis darījums tiks prioritizēts ātrākai tā pārbaudei un apstiprināšanai [18]. GAS nodeva ir jāmaksā par katru darījumu, kas veic izmaiņas blokķēdes stāvoklī. Viedajā līgumā esošo funkciju izpildei, kas neveic izmaiņas blokķēdē, bet vienkārši atgriež datus, nav jāmaksā GAS nodeva.

Veidojot viedo līgumu jāatceras, ka par katru tajā esošās funkcijas izpildi, kas veic datu rakstīšanu blokķēdē, būs jāveic GAS nodeva maksa. Tieši šī iemesla dēļ viedajā līgumā ir finansiāli dārgi veikt lielu datu daudzuma ielādi, ja par katru ielādes reizi būs nepieciešams maksāt GAS nodevu [19].

3.2. tabula

GAS patēriņa tabula balstoties uz veicamo operāciju

Operācija	GAS daudzums	Apraksts
ADD/SUB	3	Aritmētiska operācija
MUL/DIV	5	Aritmētiska operācija
ADDMOD/MULMOD	8	Aritmētiska operācija
AND/OR/XOR	3	Loģiskā operācija
LT/GT/SLT/SGT/EQ	3	Salīdzinājuma operācija
POP	2	Steka operācija
PUSH/DUP/SWAP	3	Steka operācija
MLOAD/MSTORE	3	Atmiņas operācija
JUMP	8	Beznosacījuma leciens
JUMPI	10	Nosacījuma leciens
SLOAD	200	Glabātuves operācija
SSTORE	5000/20000	Glabātuves operācija
BALANCE	400	Lietotāja bilances atgriešana
CREATE	32000	Jauna lietotāja izveidošana
CALL	25000	Jauna lietotāja izveidošana
VIEDĀ LĪGUMA IZVEIDE	53000	Jauna viedā līguma uzstādīšana uz blokķēdes

Veidojot viedos līgumus ir jākoncentrējas uz GAS efektīvu programmu izstrādi, kurā netiek veikta lieku aprēķinu veikšana, jo par katru no veiktajiem aprēķiniem ir jāmaksā. Izmantojot Remix [20] izstrādes vidi ir iespējams aprēķināt aptuvenās GAS izmaksas pirms darījuma vai viedā līguma uzstādīšanas veikšanas veicot koda analīzi.

3.5 Jaunu kriptovalūtu tokenu izveide

Šobrīd viens no vispopulārākajiem viedo līgumu izmantošanas veidiem ir jaunu kriptovalūtu marķieru izveide. Uz Ethereum balstītu marķieru izstrādei šobrīd izmanto ERC20 standartu, kas pēc būtības apraksta nepieciešamās funkcijas, lai izveidotu uz Ethereum balstītu kriptovalūtu marķierus.

Izveidotās kriptovalūtas visbiežāk tiek izmantotas tā sauktajai pūļa investīciju vākšanai jeb ICO, kur izstrādātāji piedāvā publiskajiem lietotājiem investēt projektos izmantojot tādas populārākās kriptovalūtas kā Bitcoin, Ethereum un citas, pretī saņemot attiecīgo kriptovalūtu, ko varētu uzskatīt pēc būtības kā konkrētās kompānijas daļu un, kas vēlāk var tikt izmantota, lai norēķinātos par kompānijas piedāvātajiem pakalpojumiem vai vienkārši tikt pārdota, ja tā veiksmīgi sasniedz augstāku vērtību nekā par ko tā tika sākotnēji iegādāta, kas var notikt, ja konkrētās kompānijas izveidotā sistēma kļūst veiksmīga un to pakalpojumus sāk izmantot liels apjoms lietotāju, kuriem attiecīgi, lai izmantotu konkrēto sistēmu nepieciešams iegadāties konkrēto kriptovalutu tādējādi ceļot tā vērtību.

3.5.1 ERC standarti

Visbiežāk izmantotais standarts uz Ethereum balstītu tokenu veidošanai šobrīd ir ERC20, kurā ir definēta nepieciešamā loģika tokenu veidošanai, kas eksistētu uz Ethereum blokķēdes.

ERC20 standartā ietilpst sekojošās definējošās funkcijas:

- totalSupply() – atgriež kopējo tokenu daudzumu
- balanceOf(address _owner) – atgriež lietotājam piederošo tokenu daudzumu, kā argumentu saņemot lietotāja publisko adresi mainīgajā ‘ ‘_owner’.
- transfer(address _to, uint256 _value) – funkcija kā argumentus saņem skaitli mainīgajā ‘_value’, nosūta attiecīgo daudzumu tokenu uz publisko adresi, kas eksistē mainīgajā ‘_to’ un izsauc Transfer notikumu
- transferFrom(address _from, address _to, uint256 _value) – funkcija kā argumentus saņem sūtītāja ‘_from’ un saņēmēja ‘_to’ adreses, kā arī skaitli ‘_value’ un veic attiecīgā daudzuma tokenu pārskaitīšanu starp abām adresēm un izsauc Transfer notikumu.
- approve(address _spender, uint256 _value) – funkcija, kas ļauj ‘_spender’ lietotājam izņemt jebkādu tokenu daudzumu līdz pieļaujamajam ‘_value’ daudzumam.
- allowance(address _owner, address _spender) – atgriež tokenu skaitlisko daudzumu, ko ‘_spemder’ lietotājam ir atļauts izņemt no ‘_owner’ lietotāja konta.

Iekļautie notikumu veidi:

- Transfer(address indexed _from, address indexed _to, uint256 _value) – tiek izsaukts, kad tiek veikta tokenu pārskaitīšana.
- Approval(address indexed _owner, address indexed _spender, uint256 _value) – tiek izsaukts, kad tiek izsaukta ‘approve’ funkcija

ERC20 standartā ir pieņemti divi veidi, kas tiek izmantoti, lai veiktu tokenu pārskaitījumus:

1) transfer() – funkcija, kas tiek izmantota, lai pārskaitītu tokenus uz citu lietotāju adresēm.
2) approve() + transferFrom() – lai iemaksātu tokenus viedajā līgumā.

Tā kā viedā līguma un parasta lietotāja konti pēc uzbūves ir ļoti līdzīgi un to publiskās adreses tiek ģenerētas pēc viena principa, var rasties situācija, kad tiek mēģināts veikt tokenu iemaksu viedajā līgumā izmantojot transfer() funkciju. Šādā gadījumā pārskaitījums būs veiksmīgs, bet šāda veida pārskaitījums netiks atpazīts viedajā līgumā un, ja tajā nav implementēta neadekvātu pārskaitījumu atgriešanai nepieciešamā funkcionalitāte, tad pārsūtītie tokeni tiks savā veidā mūžīgi iesaldēti attiecīgajā viedajā līgumā un tos nav iespējams atgūt atpakaļ.

Šīs ERC20 tokenu standartā esošās kļūdainās darbības dēļ Ethereum tīklā ir zaudēti vairāku miljonu vērti tokeni.

Pēc internetā pieejamas informācijas [21]:

- $730,000 vērti EOS tokeni ir iesaldēti viedajā līgumā
- $520,000 vērti Qtum tokeni ir iesaldēti viedajā līgumā
- $123,000 vērti STORJ tokeni ir iesaldēti viedajā līgumā
- 310,067 GNT tokeni ir iesaldēti Golem viedajā līgumā
- 14,506 1ST tokeni ir iesaldēti FirstBlood viedajā līgumā

Vadoties pēc uzrādītajiem zaudēto līdzekļu apjomiem rodas jautājums, kāpēc vispopulārākajā viedo līgumu izstrādes platformā joprojām visplašāk izmantotais tokenu veidošanas standarts ir ERC20, neskatoties uz tajā iespējami izraisāmo kļūdaino darbību, jo bez ERC20 eksistē arī citi standarti, kas izveidoti, lai novērstu minēto un citu kļūdainu darbību notikšanu.

Kā iespējamos iemeslus kāpēc šis standarts joprojām tiek izmantots ir:

- Ethereum joprojām reklamē ERC20 kā izmantojamo standartu, lai gan apzinās tajā esošās kļūdainās darbības iespējamību. Šobrīd Ethereum apzinās viņu platformā esošās mērogojamības problēmas un jauna standarta ieviešana varētu izraisīt nevēlamus efektus uz kopējā tīkla ātrdarbību.
- Tokenu izstrādātāji neuzņemas atbildību par iespējamajiem lietotāju zaudējumiem, ko tie var izraisīt sūtot tokenus uz viedajiem līgumiem to neatbalstītā veidā.
- Daudzi kriptovaūtu maki un apmaiņas vietas atbalsta tikai ERC20 standartu un, lai būtu iespējams pievienot citu ERC protokola tokenu, kas nav atpakaļ saderīgs ar ERC20, nepieciešams veikt papildus programmēšanas darbus, lai nodrošinātu atbalstu. Eksistē ERC protokoli, kas ir atpakaļ saderīgi ar ERC20 standartu un kurus iespējams izmantot turpat kur iespējams ERC20. Citi eksistējošie standarti aprakstīti tekstā zemāk.

Citas ERC20 standarta alternatīvas varētu iedalīt divās kategorijās, kur pirmā kategorija ir ar standartiem, kas ir atpakaļ saderīgi ar ERC20 standartu:

- ERC223
- ERC621
- ERC777
- ERC827

Un kategorija, kas nav atpakaļ saderīga ar ERC20 standartu:

- ERC721

3.6 Decentralizēta automatizēta organizācija

Izmantojot viedos līgumus ir arī iespējams izveidot decentralizēta autonoma organizāciju jeb DAO [22], kurā pastāv pilnīga demokrātija un tajā esošo līdzekļu kontrole un izmantošana tiek veikta izmantojot sistēmā esošo lietotāju balsojuma rezultātus, kur nepieciešams iegūt noteiktu pārsvaru balsu, lai konkrētās darbības tiktu izpildītas

3.7 Decentralizētas lietojumprogrammas

Decentralizētas lietojumprogrammas pēc uzbūves sastāv no viena vai vairākiem savstarpēji saistītiem viedajiem līgumiem un lietotāja saskarnes, kas parasti tiek realizēta kā tīmekļa vietne.

Lai nodrošinātu iespēju tīmekļa pārlūkprogrammai komunicēt ar Ethereum blokķēdi un tajā esošajiem viedajiem līgumiem nepieciešams izmantot web3.js javascript [23] vai kādu citu līdzīgu funkcionalitāti nodrošinošu bibliotēku. Šo bibliotēku ir iespējams izmantot divos veidos:

1) Veikt web3.js integrāciju tīmekļa vietnē un tam nepieciešamā lokālā Ethereum node uzstādīšanu uz servera. Šajā gadījumā web3 komunicē ar lokālo Ethereum node, kas ir sinhronizēta ar Ethereum tīklu.
2) Izmantojot Google Chrome eksistējošo Metamask paplašinājumu, kas dod iespēju komunicēt ar Ethereum blokķēdi pa tiešo bez vajadzības veidot savu lokālo node.

3.1. att. **Lietotāja saskarnes un viedā līguma komunikācijas arhitekrūra**

Šobrīd eksistē daudzas populāras uz Ethereum balsīttas decentralizētas lietojumprogrammas, kas ir realizētas dažādiem mērķiem. Kā interesantu piemēru iespējams minēt pašu populārāko uz Ethereum balstītu spēli Crypto Kitties, kas tās lietotājiem ļauj iegādāties, kolekcionēt un savstarpēji digitāli pārot virtuālos kaķus, kur katram no tiem piemīt unikālas īpašības. Šī spēle ir populāra blokķēdes tehnoloģijas interesentu vidu, jo tā 2017. gada decembra beigās guva lielu uzmanību un kļuva strauji populāra, īsa laikā sasniedzot 13% no kopējā Ethereum tīkla darījumu skaita, kas rezultējas kopējā Ethereum tīkla transakciju apstiprināšanas darbības ātruma samazināšanā un darījumu apstiprināšanas rindā esošo transakciju seškārtējā skaita palielināšanā, kas atklāja pirmās reālās Ethereum mērogojamības problēmas izpausmes. Eksistē daudzas citas populāras un interesantas DAPP realizācijas kā piemēram:

- Augur, kas piedāvā lietotājiem veikt likmes prognozēšanas tirgū uz nākotnes notikumiem
- EtherDelta, kas piedāvā decentralizētu Ethereum kriptovalūtas tokenu tirdzniecību

- BasicAttentionToken, kas piedāvā digitālās reklamēšanās iespējas
- Decentralland, kas eksistē kā virtuālā realitāte, kurā tās lietotājiem ir iespējams iegādāties virtuālos zemes gabalus uz kura tās lietotājiem ir iespējams būvēt pašiem savu virtuālo realitāti un objektus. Kā pats projekts citē: "Your imagination is the limit", kas tulkojot idejiski nozīmē: "Vienīgais ierobežojums ir tava iztēle".

3.8 Viedo līgumu tehniskās realizācijas ierobežojumi

Lai veiksmīgi izveidotu decentralizētas lietojumprogrammas prototipu vispirms nepieciešams veikt viedo līgumu tehnisko iespēju un ierobežojumu [25] izpēti un vadoties pēc iegūtās informācijas konceptuāli aprakstīt kāda veida lietojumprogrammas šobrīd ir iespējams izveidot.

Zemāk aprakstītie ierobežojumi neattiecas tikai uz Ethereum platformu, bet arī uz citām publiskajām blokķēdēm, kas piedāvā viedo līgumu izstrādi, jo šie ierobežojumi rodas no bloķēdes un viedo līgumu arhitektūras uzbūves īpašībām .

1) Viedā līguma saziņa ar ārējiem servisiem

Kā pirmo ierobežojumu iespējams minēt viedā līguma spēju sazināties ar arējiem servisiem. Šāda veida saziņu tiešā veidā nav iespējams realizēt, jo blokķēdes risinājuma pamatā eksistē uz vienprātību balstīta sistēma, kuras ietvaros katram blokķēdes mezglam, kas veic veikto darījumu pārbaudi ir jānonāk pie identiska rezultāta, tādēļ visām uz blokķēdes veiktajām darbībām ir jābūt paredzamām un mezglam veicot vienas transakcijas pārbaudi nedrīkst rasties situācija, kad var iegūt dažādus rezultātus. Dažādu rezultātu situācija viedajam līgumam sazinoties ar ārējiem resursiem var rasties, jo katrs blokķēdes mezgls pārbauda veikto transakciju neatkarīgi no citiem mezgliem, tādējādi datu atgūšana no ārējiem servisiem tiek veikta vairākas reizes, jo tos izsauc katrs mezgls. Šādā situācijā nav iespējams garantēt, ka katrs mezgls no ārējā servisa saņems vienu un to pašu atbildi, jo katrs mezgls neveic transakcijas pārbaudi vienlaicīgi, tāpēc laika atšķirības dēļ ārējais serviss var atgriezt dažādus datus, vai arī neatgriezt datus vispār, jo uz laiku kļūst nepieejams.

Lai tehniski realizētu datu saņemšanu no ārējiem servisiem, tā vietā, lai viedais līgums veiktu pieprasījumu uz arējiem servisiem, nepieciešams izmantot uzticamas trešās personas pakalpojumus, kas veiktu nepieciešamo datu ievietošanu blokķēdē, tādējādi katram mezglam

būs pieejami vienādi dati, kas tiks izmantoti viedā līguma loģikas izpildei un visos mezglos tiks iegūts vienāds rezultāts.

Kā konkrētu aprakstītās situācijas piemēru ir iespējams minēt velmi realizēt labības apdrošināšanas servisu, kas atkarībā esošajiem laikapstākļiem veic automātisku apdrošināšanas summu izmaksu, ja laikapstākļi ir nelabvēlīgi un nav iespējams veikt veiksmīgu lauksaimniecību.

Līdzīgi ierobežojumi rodas situācijās, kad tā vietā, lai viedais līgums iegūtu datus no ārējiem servisiem, tam ir nepieciešams izsaukt kādu ārējo servisu, lai ierosinātu notikumu izpildi ārpus blokķēdes ietvariem. Šādā situācijā arī katrs mezgls veic transakcijas pārbaudi un viedajā līgumā esošā koda izpildi, tādēļ katram no tiem būtu nepieciešams izsaukt ārējo servisu. Šādā realizācijā ārējais serviss vienas transakcijas dēļ tiku izsaukt simtiem reižu, kas nav labs programmatūras realizācijas veids.

Arī šo situāciju būtu iespējams atrisināt izmantojot uzticamu trešo personu pakalpojumus, kas veiktu ārpus blokķēdes esošu notikumu izpildes ierosināšanu, balstoties uz novērotajām blokķēdē veiktajām datu izmaiņām un jaunajiem ierakstiem. Šādā situācijā blokķēde veic datu validāciju un glabāšanu, savukārt trešā persona veic attiecīgi nepieciešamo darbību izpildi, balstoties uz blokķēdē esošajiem datiem.

Var secināt, ka, lai veiksmīgi realizētu datu apmaiņu starp viedo līgumu un tā saucamo ārējo pasauli ir nepieciešama uzticama trešā persona, kas veiktu iepriekš minētās darbības un šī iemesla dēļ šādā veidā realizēti viedie līgumi daļēji zaudē decentralizētas sistēmas īpašības, jo ir nepieciešams uzticēties kādai trešajai personai un tās sniegtajiem pakalpojumiem.

Viens publiski izmantojams šādas trešās personas piemērs ir Oraclize [24], kas tā lietotājiem kā pieejamos datu avotus piedāvā:

- URL – dod piekļuves iespējas jebkurai mājaslapai vai pieejamam HTTP API galapunktam.
- Random – atgriež nemodificētus gadījuma veidā ģenerētus baitus, kas tiek ģenerēti drošā lietotnē.
- WolframAlpha – dod piekļuvi sistēmas skaitļošanas līdzekļiem
- IPFS – dod iespēju atgriezt faila saturu, kas tiek glabāts IPFS decentralizētajā tīklā

Pēc dotajiem realizācijas piemēriem var redzēt, ka blokķēde un tajos esošie viedie līgumi ir savā ziņā ir ierobežoti tādu darbību izpildei, kas līdzinās parastai datubāzei, kurā ir iespējams

rakstīt un no kuras var nolasīt datus, tāpēc kā jau iepriekš minēts blokķēdi var uzskatīt par decentralizētu datubāzi.

2) Viedo līgumu realizēta regulāro maksājumu automatizēta izpilde

Kā otro ierobežojumu ir iespējams minēt viedajā līgumā realizētu regulāro maksājumu izpildes automatizāciju, kas reālajā dzīvē nav realizējama praktiski derīgā veidā. Šī piemēra pamatā ir ideja, ka viedo līgumu varētu izmantot, lai automatizētu regulāro maksājumu veikšanu konkrētā laikā, kas atkārtotos ik pēc noteikta laika perioda. Reālajā dzīvē tie varētu būt regulāra aizdevuma apmaksa, maksa par īri vai līzingā iegādātu auto. Lai realizētu šāda veida viedo līgumu nepieciešamajiem naudas līdzekļiem būtu jābūt konkrētā viedā līguma kontrolē, pretējā gadījumā viedais līgums nevarētu garantēt maksājuma izpildi. Šādā gadījumā līdzekļus, kas atrodas viedajā līgumā nevar kontrolēt to reālais īpašnieks, jo, ja tie būtu īpašnieka kontrolē, tad tas jebkurā brīdī tos varētu pārvietot no viedā līguma uz citu kontu un maksājumi vairs nebūtu iespējami. Pēc minētā var secināt, ka šādā veidā realizēts viedais līgums būs bezjēdzīgs saņēmējam, ja maksātājam būs pieejama kontrole pār līgumā esošajiem līdzekļiem un tas var jebkurā brīdī tos pārvietot uz citu kontu, vai arī bezjēdzīgs maksātājam, ja nepieciešamie līdzekļi būs viedā līguma kontrolē un maksātājs tos nevarēs izmantot savu mērķu realizācijai, jo šādā situācijā viedajā līgumā būtu nepieciešams ievietot jau visu pilnai apmaksai nepieciešamo līdzekļu apjomu, lai garantētu, ka tiks veikti visi nepieciešamie maksājumi.

3) Konfidenciālu datu slēpšana

Izmantojot viedos līgumus nav iespējams veikt konfidenciālu datu slēpšanu no nevēlamiem lietotājiem.

Viedajos līgumos ir iespējams veikt loģiskus aprēķinus un datu glabāšanu maza izmēra datubāzē pār kuru tam ir pilnīga kontrole. Lasīt vai rakstīt šajā viedajam līgumam piederošajā datubāzē glabātos datus ir iespējams izmantojot tikai konkrētajā viedajā līgumā esošās funkcijas, tādējādi aizliedzot vienam viedajam līgumam lasīt cita viedā līguma datubāzē esošo informāciju tiešā veidā. Viedajā līgumā esošajām funkcijām ir iespējams realizēt piekļuves tiesības tādējādi atļaujot tikai konkrētiem lietotājiem veikt šo funkciju izsaukšanu.

Balstoties uz augstāk minētā viedo līgumu realizācijas apraksta varētu šķist, ka tajos ir iespējams glabāt konfidenciālu informāciju izmantojot piekļuves tiesību ierobežošanu konkrētiem lietotājiem, tādējādi neļaujot nevēlamiem lietotājiem piekļūt tajos esošajai informācijai. Šādā veidā tik tiešām ir iespējams slēpt informāciju no parastajiem lietotājiem, kas veic viedā līguma funkciju izsaukšanas mēģinājumus, bet šo informāciju nav iespējams

slēpt no blokķēdes mezgliem, jo tajos atrodas pilna blokķēdes datu kopija un konkrēta mezgla lietotājam ir iespējams veikt šīs informācijas lasīšanu tiešā veidā no lokālas bloķēdes kopijas.

Konfidenciālu datu uzglabāšanu būtu iespējams realizēt privātās blokķēdēs, kas pēc savas uzbūves ir centralizētas, jo katrs no tīklā eksistējošajiem mezgliem savā veidā pieder šīs privātās bloķēdes īpašniekiem, tādā veidā neļaujot mezglu veidošanu neuzticamām trešajām personām, bet šis gadījums neattiecas uz darba mērķi realizēt decentralizētu lietojumprogrammu, jo šādā situācijā joprojām nebūtu iespējama veiksmīga konfidenciālu datu slēpšana.

4) Sākotnējo datu ielāde viedajā līgumā

Kā jau minēts augstāk esošajā trešajā punktā, vienīgais veids kā lasīt vai rakstīt informāciju viedajā līgumā ir izmantojot tam esošās funkcijas. Funkcijas, kas lasa datus no viedā līguma ir iespējams izpildīt par brīvu un par to izpildi nav nepieciešams maksāt transakcijas nodevu, kas Ethereum gadījumā tiek dēvēts par GAS, vairāk info iespējams skatīt 3.4.Savukārt funkcijām, kas veic izmaiņas viedajā līgumā esošajos datos šī nodeva ir jāmaksā un tā tiek aprēķināta atkarībā no darbībām, kas ir jāveic viedā līguma koda izpildes laikā. Šī iemesla dēļ sākotnēja datu ielāde situācijās, kad ielādei nepieciešamo datu daudzums ir liela apjoma var būt salīdzinoši dārgs process un izmaksāt pat vairākus Ether.

4. DECENTRALIZĒTAS AUTO STĀVVIETAS NOMAS LIETOJUMPROGRAMMAS PROTOTIPS

Vadoties pēc darbā aprakstītās informācijas un viedo līgumu ierobežojumu izpētes, darba ietvaros tika izveidots decentralizētas lietojumprogrammatūras prototips auto stāvvietas nomai, lai noskaidrotu un aprakstītu pastāvošās iespējas un veidus kā blokķēdes tehnoloģiju būtu iespējams sasaistīt ar reālās pasaules notikumiem un piedāvātajiem pakalpojumiem, kā arī uzskaitīt un aprakstīt eksistējošos ierobežojumus šādu sistēmu reālā ieviešanā. Dotās lietojumprogrammas prototips atbilst 2.2.3 punktā minētajam otrajam decentralizētas lietotnes veidam, kas veic darbības ar kriptovalūtu balstoties uz no ārējiem resursiem saņemtas informācijas, pēc kuras tiek pieņemti lēmumi, kā rīkoties ar pieejamajiem līdzekļiem

4.1 Sistēmas prototipa darbības apraksts

Šīs sistēmas potenciālajiem lietotājiem ir nepieciešams Ethereum konts ar tajā esošu Ether kriptovalūtu, kas tiks izmantota samaksai par saņemtajiem stāvvietas pakalpojumiem. Viedajā līgumā tika ietverta nepieciešamā funkcionalitāte, lai par katru sistēmā esošo stāvvietu būtu pieejama gan vispārīgā to aprakstošā informācija, gan informācija par tajā pieejamajām brīvajām un aizņemtajām vietām, kā arī informācija par reģistrētajiem klientiem un tiem piederošā auto reģistrācijas numuru, kas tiks izmantots, lai noteiktu, vai konkrētais auto ir reģistrēts sistēmā un tam ir pieejami autostāvvietas pakalpojumi, izmantojot pie stāvvietām esošās auto numuru atpazīšanu veicošās kameras. Ja auto tiek atpazīts tad tiek veikta automātisko vārtu pacelšana un uzsākta laika atskaite par stāvvietas pakalpojuma izmantošanu.

Lai lietotājs būtu spējīgs veikt stāvvietas nomu, tam ir nepieciešams viedajā līgumā iemaksāt kriptovalūtas depozītu, kas tiks izmantots, lai veiktu samaksu par izmantotajiem pakalpojumiem un kura atlikumu lietotājs var atgriezt savā kontā, ja vairs nevēlas saņemt stāvvietas nomas pakalpojumus. Šāda depozīta ieskaitīšana ir nepieciešama, jo viedais līgums var garantēt samaksu par izmantotajiem pakalpojumiem tikai situācija, kad tam ir kontrole pār līdzekļiem, kas tiks izmantoti samaksas veikšanai, tāpēc dotajiem līdzekļiem nepieciešams atrasties viedajā līgumā.

Šāda decentralizēta lietojumprogramma strādātu kā publiski pieejama distributīva datubāze, kuru izmantojot būtu iespējams veikt automatizētu maksājumu veikšanu par izmantotajiem pakalpojumiem ar precizitāti līdz pat sekundei, tādējādi nodrošinot, ka lietotājs maksā tikai par stāvvietas pakalpojuma izmantoto laiku.

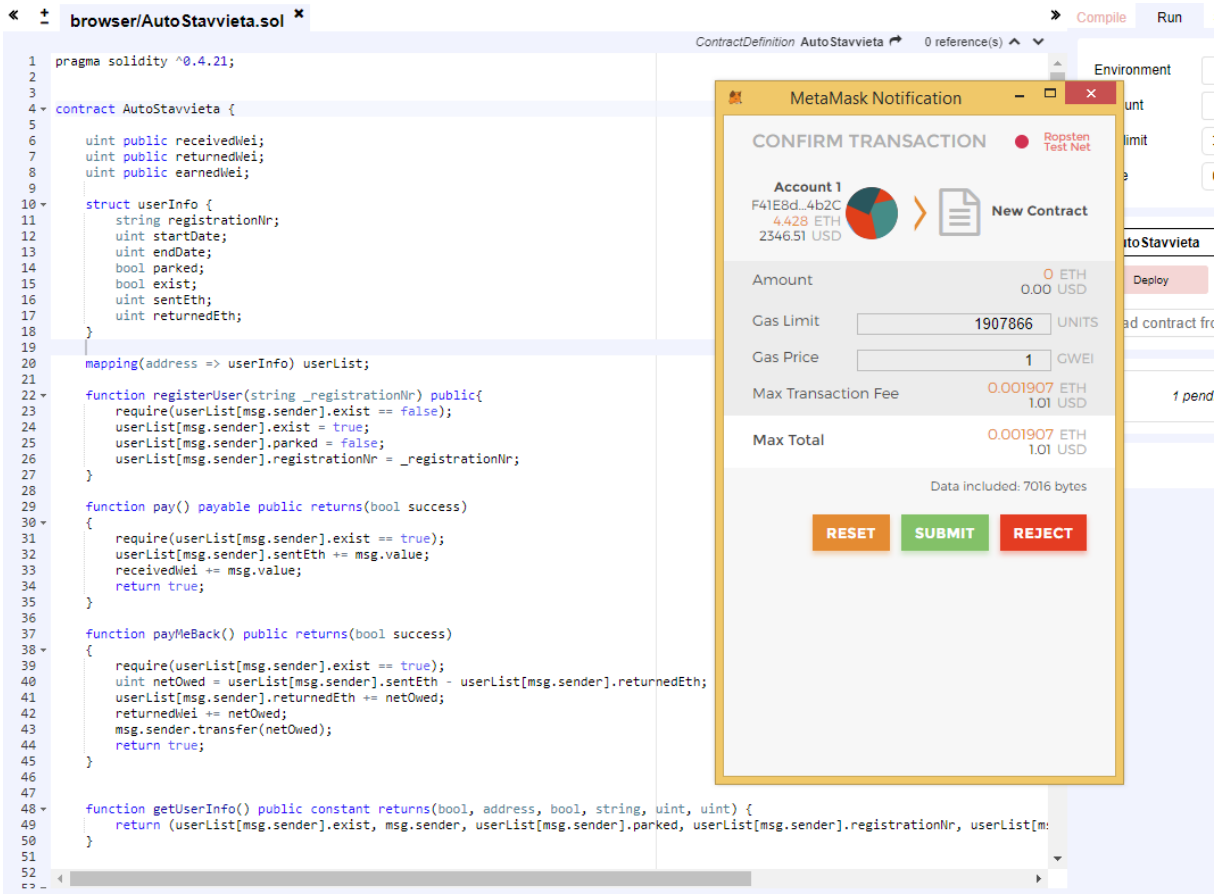
Šajā sistēmā visiem tās lietotājiem ir datu lasīšanas iespējas un katram lietotāja veidam atsevišķas rakstīšanas iespējas. Šāda veida lietotāju implementācija ļauj papildus apskatīt un izpētīt vienas sistēmas ietvaros dažādu piekļuves tiesību līmeņu veidošanas iespējas.

4.2 Tehniskās realizācijas apraksts

Prototipa tehniskā realizācija tika veikta izmantojot Ethereum Ropsten Testnet [26], kas tiek izmantots kā testa tīkls uz kura izstrādātājiem ir iespējams veikt viedo līgumu izstrādi, uzstādīšanu un testēšanu, pirms tie tiek uzstādīti uz reālā Ethereum tīkla.

Dotajā scenārijā viedā līguma izstrāde un uzstādīšanai uz Ethereum Ropsten testa tīkla tika veikta izmantota http://remix.ethereum.org izstrādes vides mājaslapa un Google Chrome pārlūkprogrammas "Metamask" [27] paplašinājums.

http://remix.ethereum.org ir iespējams veikt Solidity [28] koda programmēšanu un uzstādīšanu gan uz reālā, gan testa Ethereum tīkla. Metamask paplašinājums nodrošina nepieciešamo web3 atbalstu, kas dod iespēju komunicēt ar blokķēdes tīklu, un iztikt bez pilna Ethereum mezgla uzstādīšanas uz savas lokālās darbstacijas.



4.1. att **Viedā līguma uzstādīšana uz Ropsten testa vides**

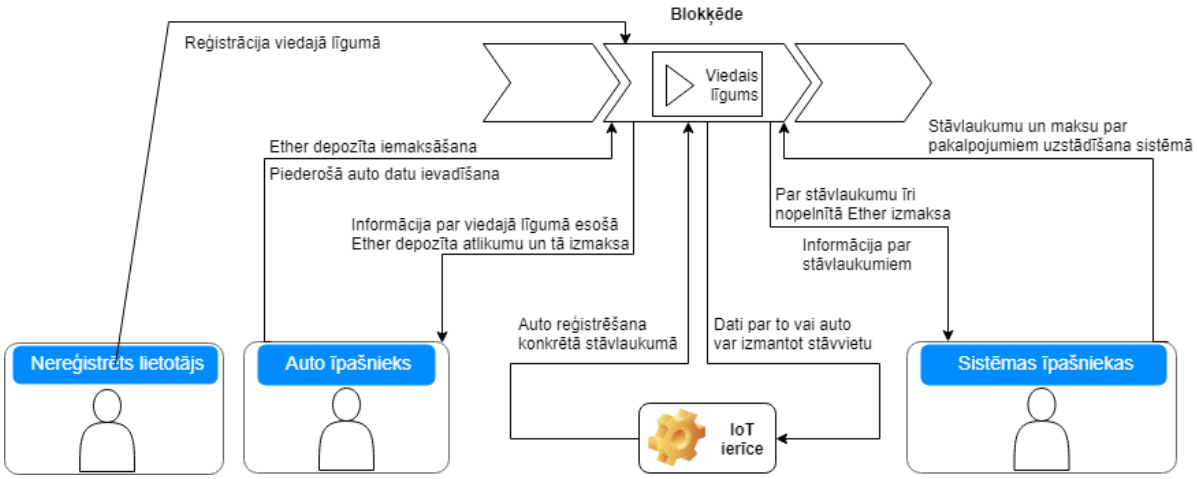
Lai veiktu viedā līguma uzstādīšanu uz blokķēdes ir nepieciešams minimāls daudzums Ether valūtas, kas GAS veidā tiks samaksāta kriptovalūtas racējiem, kas veiks konkrētā ieraksta

pārbaudīšanu un tā ierakstīšanu datu blokā, tāpēc pirms tiek veikta viedā līguma uzstādīšana ir nepieciešams iegūt minimālu daudzumu testa Ether valūtas, ko iespējams izdarīt izsaucot Ropsten tīklā esošu viedā līguma funkciju, kas automātiski pārskaita mazu daudzumu testa Ether valūtas uz izsaucēja adresi. Dotajā situācijā testa Ether ir nepieciešams Metamask esošajā makā.

4.2.1 Sistēmas lietotāji

Sistēmā lietotājus pēc tiem pieejamajām iespējām var iedalīt četros veidos:

- Auto īpašnieks, kas veic stāvvietu nomas pakalpojumu izmantošanu.
- Sistēmas īpašnieks, kas saņem maksu par iznomātajām stāvvietām.
- IoT ierīce auto numuru atpazīšanai, kas izmantojot starpniekserveri var rakstīt un lasīt datus no blokķēdē esošā viedā līguma, kā arī elektroniskā barjera, kas tiek automātiski pacelta, ja auto ir atļauts izmantot stāvlaukumu.
- Nereģistrēts lietotājs, kas var veikt reģistrēšanos viedajā līgumā.



4.2. att **Sistēmas lietotājiem un viedajam līgumam pieejamā funkcionalitāte**

Attēlā 4.2 vizuāli parādītas katram lietotājam viedajā līgumā pieejamās funkcionalitātes. Katram lietotāja veidam eksistē atsevišķas funkcionālās iespējas. Šāda veida lietotāju implementācija tika realizēta izveidojot dažādu piekļuves tiesību līmeņus viedajā līgumā esošajām funkcijām.

```
128     // MODIFIERS
129
130     // Ļauj funkciju izsaukt tikai viedā līguma īpašniekam
131     modifier onlyOwner() {
132         require(msg.sender == owner);
133         _;
134     }
135
136     // Ļauj funkciju izsaukt tikai viedajā līgumā reģistrētam lietotājam
137     modifier onlyUser() {
138         require(userList[msg.sender].exist == true);
139         _;
140     }
141
142     // Ļauj funkciju izsaukt tikai viedajā līgumā vēl nereģistrētam lietotājam
143     modifier onlyUnknown() {
144         require(userList[msg.sender].exist == false);
145         _;
146     }
147
148     // Ļauj funkciju izsaukt tikai IoT sistēmas atvēlētajam adresēm
149     modifier isIoT() {
150         require(iotWhitelist[msg.sender] == true);
151         _;
152     }
153
154     function rent(uint param) public payable isIoT(){
```

4.3. att **Ar Solidity funkcijas modifier realizēta viedajā līgumā esošo funkciju piekļuves tiesību ierobežošana un izmantošanas piemērs.**

Attēlā 4.3 redzams kā viedā līguma prototipā tika realizēta tajā esošo funkciju piekļuves tiesību ierobežošana izmantojot modifier tipa struktūru, kas ļauj realizēt konkrētu nosacījuma pārbaudi un ko atkārtoti iespējams izmantot pie visām vajadzīgajām funkcijām ievietojot tā nosaukumu funkcijas definīcijā, tādējādi katru reizi, kad tiks veikta konkrētās funkcijas izsaukšana, tiks pārbaudīts, vai konkrētam lietotājam ir ļaut izsaukt doto funkciju. Viedajā līgumā tika realizēti četri lietotāju piekļuves ierobežojoši modifier:

- onlyOwner() – ļauj funkciju izsaukt tikai viedā līguma īpašniekam.
- onlyUser() – ļauj funkciju izsaukt tika viedajā līgumā reģistrētam lietotājam
- onlyUnknown() – ļauj funkciju izsaukt tikai vēl viedajā līgumā nereģistrētam lietotājam
- isIoT() – ļauj funkciju izsaukt ārējam servisam, kas izmanto saņemtos datus no auto numura atpazīšanas sistēmas

4.2.2 Sistēmas funkcijas

Lai nodrošinātu decentralizētajai lietojumprogrammai nepieciešamo funkcionalitāti, viedajā līgumā tika realizētas sekojošās funkcijas

- registerUser() – lietotāja izveides funkcija viedajā līgumā, kurā kā ieejas parametrs tiek saņemts lietotājam piederošās mašīnas reģistrācijas numurs, pārējās lietotāju aprakstošās vērtības tiek uzstādītas automātiski.

- pay() – ļauj viedajā līgumā reģistrētam lietotājam veikt Ether valūtas iemaksu, kas tiks izmantota stāvvietas pakalpojumu apmaksai. Lai stāvvieta būtu lietotājam pieejama ir nepieciešams, lai tam būtu vismaz vienas pilnas stavlaukuma dienas īres izmaksu apjomā esoši pieejami līdzekļi viedajā līgumā.
- payMeBack() – ļauj viedajā līgumā reģistrētam lietotājam saņemt atpakaļ viedajā līgumā iemaksāto Ether atlikumu, ja viss vēl nav iztērēts par stāvvietas nomas pakalpojumiem un lietotājs vairs neplāno vispār vai īslaicīgi izmantot stāvvietas pakalpojumus.
- getUserInfo() – atgriež informāciju par lietotāju, kas izsauc šo funkciju. Funkciju iespējams izsaukt tikai viedajā līgumā reģistrētam lietotājam un IoT ierīces servisam, lai noskaidrotu vai lietotājs drīkstēs izmantot stāvvietas pakalpojumus.
- registerParkingSpace() – ļauj viedā līguma īpašniekam reģistrēt jaunu stāvlaukumu.
- getParkingSpaceInfo() – ļauj atgriezt konkrēta stāvlaukuma aprakstošo informāciju, kurā ietilpst dati, par laukumā brīvajām un aizņemtajām vietām.
- AutoStavvieta() – viedā līguma konstruktors, kas viedā līguma uzstādīšanas laikā uz blokķēdes veic līguma īpašnieka adreses datu saglabāšanu sistēmā.
- rentingSetup() – ļauj viedā līguma īpašniekam konfigurēt maksu par stāvlaukumu īri. Lai novērstu ļaunprātīgu rīcību, šo varētu realizēt viedā līguma konstruktorā un tādā gadījumā maksa par īri būtu konstanta un to vairs nevarētu izmainīt.
- setRentigPricePerSecond() – funkcija, kas tiek izsaukta no rentingSetup(), lai uzstādītu katras sekundes maksu par stāvlaukuma īri. Šī vērtība tiek uzstādīta izmantojot Ether Wei nomināli
- rent() – funkciju kuru izsauc ārējais IoT serviss, lai uzsāktu stāvlaukuma īri. Kā argumentu funkcija saņem kameras noskenēto auto reģistrācijas numuru.
- endRent() - funkciju kuru izsauc ārējais IoT serviss, lai beigtu stāvlaukuma īres pakalpojumu.
- rentingElapsedTime() – funkcija, kas tiek izsaukta no endRent(), lai noskaidrotu cik ilgi auto īrēja stāvlaukumu
- rentingAccumulatedPrice() - funkcija, kas tiek izsaukta no endRent(), lai noskaidrotu iekasējamo samaksu par stāvvietas īres pakalpojumiem
- whitelistIoT() – funkcija, kas ļauj reģistrēt ārējiem servisiem piederošās addreses, kas varēs viedajā līgumā izsaukt lietotāja datu atgriešanas, īres uzsākšanas un pārtraukšanas funckijas.

4.2.3 Sistēmas datu struktūras

Sistēmā tika izveidotas divas datu glabāšanas struktūras, lai nodrošinātu nepieciešamās informācijas glabāšanu gan par sistēmas lietotājiem, gan stāvlaukumiem.

Stāvlaukuma datu struktūrā eksistēja tikai informācija par stāvlaukuma aizņemtajām un brīvajām vietām, kas nepieciešamas, lai noteiktu vai ir iespējama jaunu auto ielaišana stāvlaukumā.

Par katru lietotāju tiek glabāti septiņi datu lauki.

```
struct userInfo {
    string registrationNr;
    uint startDate;
    uint endDate;
    bool parked;
    bool exist;
    uint sentEth;
    uint returnedEth;
}

//address katram lietotājam ir unikāla
mapping(address => userInfo) userList;
```

4.4. att. **Par lietotāju glabāto datu struktūra un kartēšana izmantojot lietotāja maka adresi**

Izmantojot lietotāja userInfo datu struktūras piesaistīšanu katra lietotāja identificējošajai unikālajai maka adresei tiek izveidots sistēmā esošo lietotāju saraksts userList;

Par lietotāju glabātās informācijas apraksts:

- registrationNr – lietotājam piederošās mašīnas reģistrācijas numurs
- startDate – lauks, kas tiek aizpildīts, kad tiek uzsākta stāvlaukuma īre, ar datumu un laiku kad sākta stāvlaukuma īre
- endDate - lauks, kas tiek aizpildīts, kad tiek beigta stāvlaukuma īre, ar datumu un laiku kad beigta stāvlaukuma īre.
- parked – boolean tipa pazīme par to vai lietotājam piederošai auto dotajā mirklī izmanto auto stāvvietas pakalpojumus un ja dota vērtība ir patiesa, tad tā aizliedz lietotājam veikt tam atlikuša Ether valūtas izņemšanu no viedā līguma.
- exist – pazīme par lietotāja eksistenci sistēmā, kas ir patiesa, ja lietotājs ir reģistrējies, un nepatiesa, ja lietotāju nav iespējams atrast userList sarakstā
- sentEth – lietotāja kopējais iesūtītais Ether daudzums viedajā līgumā, kas tiek glabāts Wei formātā
- returnedEth – lietotājam atgrieztā Ether daudzums no viedā līguma, kas tiek glabāts Wei formātā

4.3 Stāvlaukuma īres sistēmas iespējamie uzlabojumi

Pēc veiktās prototipa izstrādes tika izvirzīti potenciālie uzlabojumi, ko varētu realizēt šādas reālas sistēmas izstrādē:

- Nodrošināt iespēju citiem stāvlaukumu īpašniekiem reģistrēties sistēmā un uzsākt automatizētu stāvlaukumu īrēšanu, ja tie ir spēj nodrošināt automatizēto auto reģistrācijas numuru nolasīšana ierīces un saziņu ar viedo līgumu.
- Kā jau iepriekš minēts, vislabākais veid kā realizēt tarifu noteikšanu par stāvlaukuma īri ir to uzstādīšanu veikt viedā līguma konstruktorā, kas tiek izsaukt viedā līguma uzstādīšanas laikā uz blokķēdes. Šādā situācijā cenas būtu fiksētas un tās vairs nebūtu iespējams mainīt, tādēļ tās būtu uzticamas.
- Ieviest paziņojumu sistēmu, kas uz lietotājam pieejamo kontaktinformāciju laicīgi paziņoju par nepieciešamību papildināt viedo līgumu ar Ether, lai varētu arī turpmāk veiksmīgi izmantot stāvlaukuma pakalpojumus.
- Realizēt drošu un uzticamu veidu kā auto reģistrācijas numura ierīce varētu komunicēt ar viedo līgumu. Izveidotajā prototipā netika apskatītas šāda droša savienojuma realizācijas iespējas.
- Realizēt lietotāja draudzīgu tīmekļa vietni, kuru izmantojot lietotājam būtu iespējams viegli papildināt viedajā līgumā esošos līdzekļus apskatīt citus pieejamos datus.

REZULTĀTI

Bakalaura darbā tika veikta blokķēdes tehnoloģijas un tās potenciālo pielietojumu izpēte. Padziļināti tika apskatītas viedo līgumu izstrādes iespējas un šajā sfērā eksistējošie tehnoloģiskie līderi. Tika apskatītas tādas mūsdienu tehnoloģijā esošās problēmas, kā ierobežoti transakciju ātrumi, kas pagaidām ierobežo šīs tehnoloģijas masu adaptācijas iespējas, kā arī eksistējošie viedo līgumu realizācijas ierobežojumi, kas neļauj viedajiem līgumiem tiešā veidā sazināties ar ārpus blokķēdes esošiem servisiem, glabāt konfidenciālus datus, pilnvērtīgi realizēt automatizētu atkārtoto maksājumu veikšanu un veikt apjomīga daudzuma datu ielādi viedajā līgumā. Mūsdienās minēto ierobežojumu dēļ ir ierobežota viedo līgumu izveides un pielietojuma iespējas atsevišķu mērķu sasniegšanai. Tika veikta padziļināta Ethereum blokķēdes platformas izpēte, jo pēc veiktās izpētes, tā tika izvirzīts par mūsdienu blokķēdes sfērā esošo tehnoloģisko līderi.

Balstoties uz darba pētnieciskajā daļā iegūtajiem rezultātiem uz Ethereum bloķēdes tika izstrādās decentralizētas auto stāvvietas nomas lietojumprogrammas pamatā esošā viedā līguma prototips, kas tehniski veic automatizētu auto stāvlaukumu nomas realizāciju. Prototipa izstrādes beigās tika izvirzīti sistēmas potenciālie uzlabojumu, ko būtu nepieciešams ieviest šādas reālas sistēmas realizācijas īstenošanai.

SECINĀJUMI

Darba mērķis bija izpēti un noskaidrot bloķēdes tehnoloģijas izmantošanas iespējas un potenciālās sfēras, kurās šī tehnoloģija varētu tik veiksmīgi izmantota. Pēc pētījuma iegūto rezultātu analīzes, tika secināts, ka bloķēdes tehnoloģijai piemīt liels potenciāls tikt izmantotai dažādās nozarēs, bet mūsdienās eksistē vēl neatrisinātas tehnoloģiskās problēmas un ierobežojumi, kuru dēļ vairākās no paredzētajām sfērām to vēl nav iespējams veiksmīgi un uzticami izmantot.

Tā kā bloķēde ir salīdzinoši jauna tehnoloģija, tad tās entuziastu starpā eksistē daudz neskaidrības un zināšanu trūkums par šo tehnoloģiju un tās izmantošanas iespējām ārpus kriptovalūtu sfēras.

Veicot uz Ethereum balstītu viedo līgumu tehnoloģijas izpēti un prototipa izstrādi tika secināts, ka šajā nozarē eksistē labu pieejamo standartu trūkums un joprojām tiek plaši izmantoti tādi standarti kā ERC20 protokols, kas lietotāju kļūdainas darbības pēc ir rezultējies jau vairāku miljonu ASV dolāru vērtu līdzekļu iesaldēšanu viedajos līgumos, kurus vairs nav iespējams atgūt. Kā arī tika secināts, ka viens no viedo līgumu trūkumiem priekš jauniem šīs tehnoloģijas izstrādātajiem ir nespēja veikt viedā līguma koda labošanu, jo tas pēc uzstādīšanas uz bloķēdes vairs nav izmaināms, kā arī neeksistē uzskatāmi materiāli par visbiežāk viedo līgumu veidošanā pieļautajām kļūdām un iespējamajiem drošības trūkumiem. Šo iemeslu dēļ daudziem uz Ethereum izveidotiem viedajiem līgumiem eksistē kļūdas un drošības caurumi, kurus šīs tehnoloģijas pārzinātāji var izmantot savā labā, kā labu piemēru var minēt hakera uzbrukumu DAO pūļa investīciju vākšanas projektam, kas tika balstīts uz Ethereum un no kura minētajai personai izdevās izņemt uz savu kontu 55 miljonu ASV dolāru vērtus līdzekļus.

Viedo līgumu izstrāde Ethereum platformā ir samērā viegli uzsākama un apgūstama, jo programmētājiem ir pieejama labi realizēta viedo līgumu testēšanas vide un visi nepieciešamie izstrādes līdzekļi, lai veidotu viedos līgumus, bez lielas sākotnējo konfigurāciju nepieciešamības.

Pagaidām saziņa ar viedo līgumu visbiežāk tiek realizēta tiešā veidā izsaucot viedajā līgumā esošās funkcijas un pagaidām vēl neeksistē daudzi parastam lietotājam publiski pieejami risinājumi, kas ļauti komunicēt ar viedo līgumu izmantojot ērtu un saprotamu lietotāja saskarni.

IZMANTOTĀ LITERATŪRA UN AVOTI

1. Bitcoin: A Peer-to-Peer Electronic Cash System. [tiešsaiste]. [atsauce 14.05.2018]. Pieejams internetā: https://bitcoin.org/bitcoin.pdf
2. Ethereum frontier guide. [tiešsaiste]. [atsauce 14.05.2018]. Pieejams internetā: https://ethereum.gitbooks.io/frontier-guide/content/ethereum.html
3. Block hashing algorithm. [tiešsaiste]. [atsauce 14.05.2018]. Pieejams internetā: https://en.bitcoin.it/wiki/Block_hashing_algorithm
4. What to mine [tiešsaiste]. [atsauce 15.05.2018]. Pieejams internetā: https://whattomine.com
5. How does blockchain use public key cryptography [tiešsaiste]. [atsauce 15.05.2018]. Pieejams internetā: https://www.blockchain-council.org/blockchain/how-does-blockchain-use-public-key-cryptography/
6. Why do i need a public and private key on the blockchain? [tiešsaiste]. [atsauce 15.05.2018]. Pieejams internetā: https://blog.wetrust.io/why-do-i-need-a-public-and-private-key-on-the-blockchain-c2ea74a69e76
7. Fork (blockchain) [tiešsaiste]. [atsauce 16.05.2018]. Pieejams internetā: https://en.wikipedia.org/wiki/Fork_(blockchain)
8. What is a Node? [tiešsaiste]. [atsauce 16.05.2018]. Pieejams internetā: https://lisk.io/academy/blockchain-basics/how-does-blockchain-work/nodes
9. Blokķēdes vienprātības protokoli [tiešsaiste]. [atsauce 16.05.2018]. Pieejams internetā: http://kriptovalutas.com/blokkedes-vienpratibas-protokoli/
10. Smart Contracts: 12 Use Cases for Business & Beyond [tiešsaiste]. [atsauce 16.05.2018]. Pieejams internetā: http://www.the-blockchain.com/docs/Smart%20Contracts%20-%2012%20Use%20Cases%20for%20Business%20and%20Beyond%20-%20Chamber%20of%20Digital%20Commerce.pdf
11. Blockchain: a secure, decentralized database that's impossible to forge [tiešsaiste]. [atsauce 16.05.2018]. Pieejams internetā: https://www.oodrive.com/blog/innovation/blockchain-a-secure-decentralized-database-thats-impossible-to-forge/
12. Coin market cap [tiešsaiste]. [atsauce 16.05.2018]. Pieejams internetā: https://coinmarketcap.com/
13. Ethereum White Paper [tiešsaiste]. [atsauce 16.05.2018]. Pieejams internetā: http://www.the-blockchain.com/docs/Ethereum_white_paper-

a_next_generation_smart_contract_and_decentralized_application_platform-vitalik-buterin.pdf

14. What is a Decentralized Application? [tiešsaiste]. [atsauce 16.05.2018]. Pieejams internetā: https://www.coindesk.com/information/what-is-a-decentralized-application-dapp/
15. Lightning Network. [tiešsaiste]. [atsauce 16.05.2018]. Pieejams internetā: https://lightning.network/
16. Raiden Network. [tiešsaiste]. [atsauce 16.05.2018]. Pieejams internetā: https://raiden.network/
17. What is ether? [tiešsaiste]. [atsauce 17.05.2018]. Pieejams internetā: http://ethdocs.org/en/latest/ether.html
18. Ethereum GAS Tracker. [tiešsaiste]. [atsauce 17.05.2018]. Pieejams internetā: https://etherscan.io/gastracker
19. Account Types, Gas, and Transactions. [tiešsaiste]. [atsauce 17.05.2018]. Pieejams internetā: https://github.com/ethereum/homestead-guide/blob/master/source/contracts-and-transactions/account-types-gas-and-transactions.rst#example-transaction-cost
20. Solidity IDE. [tiešsaiste]. [atsauce 17.05.2018]. Pieejams internetā: http://remix.ethereum.org
21. Comparing ERC20, ERC223, and ERC777 Ethereum Token Standards. [tiešsaiste]. [atsauce 17.05.2018]. Pieejams internetā: http://icocrowd.com/comparing-erc20-erc223-and-erc777-ethereum-token-standards/
22. DAOs, DACs, DAs and More: An Incomplete Terminology Guide. [tiešsaiste]. [atsauce 17.05.2018]. Pieejams internetā: https://blog.ethereum.org/2014/05/06/daos-dacs-das-and-more-an-incomplete-terminology-guide/
23. Web3.js. [tiešsaiste]. [atsauce 17.05.2018]. Pieejams internetā: https://github.com/ethereum/web3.js/
24. Oraclize [tiešsaiste]. [atsauce 18.05.2018]. Pieejams internetā: http://docs.oraclize.it
25. Smart contract misconceptions [tiešsaiste]. [atsauce 18.05.2018]. Pieejams internetā: https://www.coindesk.com/three-smart-contract-misconceptions/
26. Ropsten Ethereum testnet [tiešsaiste]. [atsauce 18.05.2018]. Pieejams internetā: https://ropsten.etherscan.io/
27. Metamask [tiešsaiste]. [atsauce 18.05.2018]. Pieejams internetā: https://metamask.io/
28. Solidity [tiešsaiste]. [atsauce 20.05.2018]. Pieejams internetā: http://solidity.readthedocs.io/en/v0.4.24/

29. ERC20 [tiešsaiste]. [atsauce 20.05.2018]. Pieejams internetā: https://theethereum.wiki/w/index.php/ERC20_Token_Standard
30. Deploying Metadata on Blockchain Technologies [tiešsaiste]. [atsauce 20.05.2018]. Pieejams internetā: https://www.researchgate.net/publication/321028658_Deploying_Metadata_on_Blockchain_Technologies
31. State of the dapps [tiešsaiste]. [atsauce 20.05.2018]. Pieejams internetā: https://www.stateofthedapps.com/
32. DECENT Whitepaper [tiešsaiste]. [atsauce 20.05.2018]. Pieejams internetā: https://decent.ch/media/documents/decent-whitepaper.pdf
33. State of the dapps [tiešsaiste]. [atsauce 20.05.2018]. Pieejams internetā: https://www.stateofthedapps.com/
34. Nem dapps [tiešsaiste]. [atsauce 20.05.2018]. Pieejams internetā: https://nem.io/community/projects/
35. Ndapp [tiešsaiste]. [atsauce 20.05.2018]. Pieejams internetā: http://ndapp.org/
36. Solidity Bug Info [tiešsaiste]. [atsauce 20.05.2018]. Pieejams internetā: https://etherscan.io/solcbuginfo
37. Block explorer [tiešsaiste]. [atsauce 20.05.2018]. Pieejams internetā: https://blockexplorer.com/

Bakalaura darbs „Blokķēdes tehnoloģija, tās pielietojums un potenciāls” izstrādāts LU Datorikas fakultātē.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: _____ Rainers Eduards Strautiņš 28.05.2018

Rekomendēju/nerekomendēju darbu aizstāvēšanai

Vadītāja: Dr. sc. administr. Imants Gorbāns _____ 28.05.2018.

Recenzents: profesors Jānis Bičevskis

Darbs iesniegts Datorikas fakultātē 28.05.2018.

Dekāna pilnvarotā persona: vecākā metodiķe Ārija Sproģe _____

Darbs aizstāvēts bakalaura gala pārbaudījuma komisijas sēdē

__.06.2018. prot. Nr. __.

Komisijas sekretārs(-e):_____