TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Artyom Strelkov 206164IADB

Detsentraliseeritud krüptorahavahetuse veebirakenduse arendamine

Bakalaureusetöö

Juhendaja: Lembit Viilup
Doktorikraad

Tallinn 2025

Autorideklaratsioon

Kinnitan, et olen koostanud antud lõputöö iseseisvalt ning seda ei ole kellegi teise poolt varem kaitsmisele esitatud. Kõik töö koostamisel kasutatud teiste autorite tööd, olulised seisukohad, kirjandusallikatest ja mujalt pärinevad andmed on töös viidatud.

Autor: Artyom Strelkov

01.09.2024

Annotatsioon

Käesoleva bakalaureusetöö eesmärgiks o luua detsentraliseeritud krüptorahavahetuse veebirakendus, et pakkuda kasutajatele turvalisemat ja mugavamat kauplemiskeskkonda.

Töö raames viiakse läbi olemasolevate detsentraliseeritud krüptorahavahetuste analüüs, uuritakse nende eeliseid ja puudusi ning selgitatakse välja põhinõuded arendatavale platvormile. Selle analüüsi põhjal töötatakse välja platvormi arhitektuurne disain, sealhulgas tehnoloogiate, turvamehhanismide ja plokiahela valik.

Rakenduse arendamine koosnes kahest osast. Esimene osa hõlmas nutilepingute loomist, et hallata vahetuse funktsionaalsust, ja nende juurutamist plokiahelas. Nutikalepingud võimaldavad automatiseerida, turvata ja detsentraliseerida krüptorahade kauplemist, tagades usaldusväärsed, läbipaistvad ja vahendajateta tehingud otse plokiahelas. Teine osa oli pühendatud kasutajaliidese loomisele, mis võimaldab kasutajatel ühendada oma krüptorahakoti ja vahetada krüptovaluutasid.

Lõputöö on kirjutatud eesti keeles ning sisaldab teksti 29 leheküljel, 7 peatükki, 7 joonist, 1 tabel.

Abstract

Development of a Decentralized Cryptocurrency Exchange Platform

The goal of this bachelor's thesis is to create a decentralized cryptocurrency exchange web application to provide users with a safer and more convenient trading environment. The thesis includes an analysis of existing decentralized cryptocurrency exchanges, eploring their advantages and disadvantages, and identifying the key requirements for the platform to be developed. Based on this analysis, an architectural design for the platform is developed, including the selection of technologies, security mechanisms, and blockchain.

The development of the application consisted of two parts. The first part involved choosing smart contracts to manage the exchange functionality and deploying them on the blockchain. Smart contracts enable the automation, security, and decentralization of cryptocurrency trading, ensuring reliable, transparent, and intermediary-free transactions directly on the blockchain. The second part focused on creating the user interface, which allows users to connect their cryptocurrency wallet and exchange cryptocurrencies.

The thesis is in Estonian and contains 29 pages of text, 7 chapters, 7 figures, 1 table.

Lühendite ja mõistete sõnastik

AMM	*Automated Market Maker* - automatiseeritud turutegija
API	*Application Programming Interface* - rakendusliides
CEX	*Centralized Exchange* - tsentraliseeritud krüptorahavahetuse rakendus
CSS	*Cascading Style Sheet* - veebilehtede kujunduse ja paigutamise märgistuskeel
DApp	*Decentralized Application* - detsentraliseeritud rakendus
DeFi	*Decentralized Finance* - detsentraliseeritud rahandus
DEX	*Decentralized Exchange* - detsentraliseeritud krüptorahavahetuse rakendus
ERC	*Ethereum Request for Comment* - juhiste kogumine andmete vormindamiseks ja edastamiseks
EVM	*Ethereum Virtual Machine* - virtuaalmasin, mis käitab baitkoodi, mis on koostatud kõrgetasemelistest nutikatest lepingukeeltest
LP	*Liquidity Provider* - likviidsuse pakkuja
P2P	*Peer-to-Peer* - detsentraliseeritud platvorm, kus kaks inimest suhtlevad üksteisega otse
QR	*Quick Response* - kiire vastus

Sisukord

Jooniste loetelu

Tabelite loetelu

1 Sissejuhatus

Krüptorahade kasutajate arv maailmas kasvab pidevalt ja selline ekstensiivne areng nõuab uusi tehnoloogilisi lahendusi kasutajate varade turvalisuse tõstmiseks, samuti kaasaegseid seadusandlikke regulatsioone võimalike pettuste ennetamiseks. Krüptorahade kasutamisega kaasnev anonüümsus pakub kahtlemata paljudele investoritele suurt huvi, kuid kahjuks pole neil alati oskusi ja kogemusi krüptorahadega opereerimiseks. Operaatoreid, kes pakuvad abi krüptomaailmas tegutsemisel on küll suhteliselt palju, kuid nende tehnoloogiline ja juriidiline usaldusväärsus on erinev, samas on nende tausta kontrollimine väga keeruline kui mitte võimatu. Suurem äririsk on tegutsemine tsentraliseeritud krüptorahabörsil, kus häkkimise, volitamata juurdepääsu ja manipuleerimise riskid on kõrged, kuna kontroll kogu börsi tegevuse üle on koondunud ühe isiku või firma kätte. Seetõttu, haldamaks turvalisemalt oma krüptoraha turul tekkivaid riske, on mõistlik arendada alternatiivina detsentraliseeritud krüptorahabörse, mis ekspertide hinnangul tõstavad kauplemise turvalisust, läbipaistvust ja vähendavad võimalusi rahapesuks ja kriminaalse taustaga raha liikumiseks

Käesoleva lõputöö eesmärk on uurida ja arendada detsentraliseeritud krüptovaluutavahetusplatvormi, et pakkuda kasutajatele turvalisemat ja mugavamat kauplemiskeskkonda.

Selle töö raames viiakse läbi olemasolevate detsentraliseeritud krüptovaluutavahetuste analüüs, uuritakse nende eeliseid ja puudusi ning selgitatakse välja põhinõuded arendatavale platvormile. Selle analüüsi põhjal töötatakse välja platvormi arhitektuurne disain, sealhulgas tehnoloogiate, turvamehhanismide ja plokiahela valik.

Töö oluliseks aspektiks on ka detsentraliseeritud krüptovaluutavahetusplatvormi testimine valitud tehnoloogiaid kasutades. See etapp võimaldab hinnata arendatud platvormi tõhusust ja usaldusväärsust reaalsetes tingimustes.

Lõpuks tulemuste põhjal vaadatakse läbi saadud tulemused ja tehakse järeldused püstitatud eesmärkide saavutamise ning töö raames loodud detsentraliseeritud soojusvaheti edasiarendamise väljavaadete kohta.

1.1 Taust ja aktuaalsus

Igal aastal kasvab huvi krüptovaluutade ja plokiahela tehnoloogia vastu jätkuvalt. See väljendub nii kasutajate ja investorite arvu kasvus kui ka suurkorporatsioonide ja valitsuste osalemises selle valdkonna teadus- ja arendustegevuses. Krüptovaluuta algne idee oli peamiselt anonüümne ja detsentraliseerimine. Bitcoini anonüümne looja Satoshi Nakamoto avaldas 2008. aastal valge raamatu [1]. Raamat tutvustas detsentraliseeritud krüptoraha mõisteid. Bitcoin loodi elektrooniliste maksete alternatiivse meetodina, mis tagab tehingute detsentraliseerimise, anonüümsuse ja turvalisuse ning annab kasutajatele ka kontrolli oma rahaasjade üle.

Seetõttu suureneb krüptovaluutade populaarsuse kasvuga ka vajadus nende turvalise vahetamise infrastruktuuri arendamiseks. Detsentraliseeritud krüptovaluutavahetusplatvormid mängivad selles protsessis võtmerolli, pakkudes kasutajatele uuenduslikke ja turvalisi lahendusi.

1.2 Probleemi olemus

Tsentraliseeritud krüptovaluutavahetuse platvormid pakuvad mugavat ja kiiret kauplemise viisi, kuid nagu iga süsteem, kaasnevad ka nendega mitmeid riske.

Üks peamisi riske on raha kaotamine häkkimise, varguse, börsi või kolmandate isikute pettuse korral. Tsentraliseeritud börsid salvestavad oma serveritesse suures koguses krüptovaluutasid, muutes need häkkerite jaoks atraktiivseks sihtmärgiks. Varem on olnud mitmeid juhtumeid, kus börse on rünnatud ja kaotatud miljoneid dollareid. Näiteks 2014. aastal kaotas Mt.Goxi börs umbes 647 tuhat bitcoini, mis moodustas peaaegu 7% kogu olemasolevast pakkumisest. [2] Üks hiljutistest häkkidest, mis leidis aset juulis 2024, oli suurim India börs WazirX. Kahjum ulatus enam kui 230 miljoni dollarini. Kas investorid oma raha tagasi saavad, jääb teadmata. [3]

Peamine risk on ka kontrolli puudumine omavahendite üle. Kui kasutajad deponeerivad oma krüptovaluutat tsentraliseeritud börsil, annavad nad neile oma privaatvõtmed, mis on

ainus viis omandiõiguse ja raha liikumise kontrollimiseks. See tähendab, et kasutajad loodavad börsi usaldusele ja usaldusväärsusele, kes võib igal ajal oma reegleid muuta, kontosid külmutada või konfiskeerida, määrata kõrgeid tasusid või väljamaksete piiranguid. Lisaks on tsentraliseeritud börsidel regulatiivne surve ning need võivad olla sunnitud järgima erinevaid seadusi ja regulatsioone, nagu kliendi tuvastamise nõuded, kahtlaste tehingute aruandluse nõuded, sanktsioonid ja keelud teatud krüptovaluutade või riikide suhtes.

Kolmas risk on läbipaistvuse puudumine. Tsentraliseeritud börsid tegutsevad suletud ja tsentraliseeritud põhimõttel, mistõttu on nende tegevuse ja finantsseisundi kontrollimine keeruline. Kasutajad ei saa olla kindlad, et börsil on tegelikult piisavad reservid kõigi oma kohustuste katmiseks või et ta ei tegele turuga manipuleerimisega, vale pakkumise või nõudlusega, siseringi kauplemisega või kokkumänguga teiste osalejatega. Tsentraliseeritud börsidel võib esineda ka tehnilisi tõrkeid, vigu, viivitusi või ummikuid, mis võivad põhjustada kauplemisvõimaluste kaotamist, korralduste ebaõiget täitmist või soovimatuid hinnamuutusi. Näiteks 2022. aastal läks igapäevase kauplemismahu järgi 4. tipus olnud FTX börs pankrotti pärast seda, kui avastati, et tal pole piisavalt reaalvara, et katta kohustusi klientide ja investorite ees. FTX-i kokkuvarisemine tõi kaasa umbes 2 miljardi euro suuruse võla investorite ees, krüptovaluutade hindade languse ja usalduse kaotuse krüptotööstuse vastu [4].

Nende probleemide lahendamiseks on tekkinud nõudlus detsentraliseeritud krüptorahavahetuse järele, kus tehingud toimuvad otse kasutajate vahel ilma vahendajaid kaasamata.

1.3 Eesmärk

Käesoleva lõputöö eesmärk on uurida ja arendada detsentraliseeritud krüptovaluutavahetusplatvormi, et pakkuda kasutajatele turvalisemat ja läbipaistvamat kauplemiskeskkonda. Selle töö raames viiakse analüüsi põhjal läbi olemasolevate detsentraliseeritud krüptovaluutavahetuste analüüs, selgitatakse välja põhinõuded arendatavale platvormile. Selle analüüsi põhjal töötatakse välja platvormi arhitektuurne disain, sealhulgas turvalisuse ja jõudluse tagamise tehnoloogiate, protokollide ja mehhanismide valik. Loodud rakendus peaks pakkuma kasutajatele turvalisemat ja läbipaistvamat krüptovaluutavahetust. Loodud liides võimaldab kasutajatel intuitiivselt

tehinguid teha. Isegi inimestel, kellel pole detsentraliseeritud vahetuste kasutamise kogemust, ei teki kiire vahetamisega raskusi.

2 Plokiahela tehnoloogiad ja detsentraliseeritud rakendused

Plokiahela tehnoloogia on täiustatud andmebaasimehhanism, mis võimaldab läbipaistvat teabe jagamist ärivõrgus. Plokiahela andmebaas salvestab andmed plokkidena, mis on omavahel ahelas ühendatud. Andmed on kronoloogiliselt järjepidevad, ahelat ei ole võimalik võrgust konsensuseta kustutada ega muuta. Selle tulemusena kasutavad inimesed plokiahela tehnoloogiat, et luua muutumatu või muutumatu pearaamat tellimuste, maksete, kontode ja muude tehingute jälgimiseks. Süsteemil on sisseehitatud mehhanismid, mis takistavad volitamata tehingute sisestamist ja loovad järjepidevuse nende tehingute jagatud vaates. [5]

2.1 Plokiahela tehnoloogia põhiprintsiibid

Plokiahela tehnoloogial on järgmised põhifunktsioonid:

Detsentraliseerimine plokiahelas viitab kontrolli ja otsuste tegemise üleandmisele tsentraliseeritud üksuselt (indiviidilt, organisatsioonilt või rühmalt) hajutatud võrku. Detsentraliseeritud plokiahelavõrgud kasutavad läbipaistvust, et vähendada osalejate vahelise usalduse vajadust. Need võrgud takistavad ka osalejaid üksteise üle võimu või kontrolli avaldamast viisil, mis halvendab võrgu funktsionaalsust.

Muutumatus tähendab, et midagi ei saa muuta ega muuta. Ükski osaleja ei saa tehingut rikkuda, kui keegi on selle jagatud pearaamatusse registreerinud. Kui tehingukirje sisaldab viga, peate vea tühistamiseks lisama uue tehingu ja mõlemad tehingud on võrgus nähtavad.

Plokiahela süsteem kehtestab reeglid osalejate nõusoleku kohta tehingute salvestamiseks. Uusi tehinguid saate salvestada ainult siis, kui enamik võrgus osalejaid on selleks nõusoleku andnud.

Plokiahela tehnoloogia transformatiivne potentsiaal läbipaistvate ja jõustatavate omandiõiguste loomisel, kaitstes samas isikuvabadusi. Plokiahela pakub oma muutumatuse, läbipaistvuse ja töötõendite kasutamisega lahendust, võimaldades turvalisi ja detsentraliseeritud tehinguid, mida ei saa tagasi pöörata ega rikkuda. Nutikad lepingud

on selle uuenduse kesksel kohal, võimaldades täpseid ja isetäituvaid kokkuleppeid, mis suudavad hallata keerulisi õigusi, nagu intellektuaalomand või ettevõtte juhtimine. Need lepingud välistavad sõltuvuse kolmandatest osapooltest, vähendavad pettusi ning tagavad juriidiliste ja ühiskondlike lepingute järgimise. Plokiahela detsentraliseeritud olemus annab inimestele võimaluse kontrollida oma andmeid, osaleda jagatud ressurssides ja ammutada isiklikust teabest väärtust ilma nõusolekut või privaatsust kaotamata. See loob uue paradigma varade, lepingute ja individuaalsete õiguste haldamiseks digitaalses maailmas. [6]

2.2 Detsentraliseeritud rakendused plokiahelas

Detsentraliseerimine hõlmab võimu, kontrolli ja otsuste tegemise jaotamist võrgustiku või süsteemi vahel, selle asemel, et koondada ühte organisatsiooni või üksikisikut. Selline volituste jaotus mitme osaleja vahel välistab vajaduse keskse üksuse järele teha kõik otsused. Sellised tehnoloogiad nagu plokiahel võimaldavad seda detsentraliseerimist, kus mitu arvutit (sõlmed) haldavad jagatud andmebaasi ja kontrollivad tehinguid. See tagab, et ühelgi üksusel pole süsteemi üle täielikku kontrolli.

DAppide toiteallikaks on nutikad lepingud, mille taustakood töötab hajutatud võrdõigusvõrkudes. Nutikas leping on eelnevalt määratletud reeglite kogum, mida jõustab arvutikood. Kui teatud tingimused on täidetud, täidavad kõik võrgusõlmed lepingus määratud ülesandeid. Kui nutikas leping on plokiahelas kasutusele võetud, on koodi raske muuta või hävitada, mis võib säilitada DApp-i funktsionaalsuse isegi siis, kui selle taga olev meeskond laiali läheb. [7]

Tsentraliseeritud ja detsentraliseeritud rakenduste (DApps) võrdlemisel on olulisi erinevusi arhitektuuri, kontrolli, turvalisuse ja kasutuskogemuse osas. (Tabel 1)

Tabel 1. Tsentraliseeritud ja detsentraliseeritud rakenduste võrdlemine

Tsentraliseeritud rakendused	**Detsentraliseeritud rakendused**
Veebirakendusi, nagu Twitter ja Google, haldab ja kontrollib keskne ettevõte.	DApp-e juhitakse hajutatud võrgus, kus puudub keskne osapool, mis võimaldab mitmel osalejal otsuseid langetada.
Traditsiooniline klient-server arhitektuur: kasutaja suhtleb kesksserveriga, mis töötleb päringuid ja vastab neile.	Peer-to-peer (P2P) arhitektuur: kasutaja saab otse suhelda, suhelda ja teha tehinguid teiste kasutajatega, ilma et oleks vaja vahendajaid.

Tsentraliseeritud rakendused	Detsentraliseeritud rakendused
Tsentraliseeritud andmesalvestus: rakenduste andmeid salvestatakse tavaliselt tsentraliseeritud serveritesse, mis muudab need andmetega seotud rikkumiste või volitamata juurdepääsu suhtes haavatavaks.	Detsentraliseeritud andmesalvestus: DApp-i andmed salvestatakse detsentraliseeritud võrku, näiteks plokiahelasse, tagades andmete terviklikkuse, turvalisuse ja vastupidavuse üksikutele tõrkepunktidele.
Rakendused traditsioonilises Interneti-infrastruktuuris, tsentraliseeritud serveritele ja sideprotokollidele.	DApps võimendab oma tööks plokiahelat, võimaldades läbipaistvat ja detsentraliseeritud tehingute töötlemist ja konsensusmehhanisme.
Traditsiooniliste rakenduste kasutajad peavad usaldama rakenduse pakkujale oma andmed, tehingud ja rakenduse nõuetekohase toimimise.	DApp-id töötavad usaldusväärsetes süsteemides, võimaldades kasutajatel suhelda otse plokiahela võrguga ilma keskasutusele või vahendajatele lootmata.

2.3 Detsentraliseeritud rakenduste eelised ja piirangud

Detsentraliseeritud rakendus (DApp) on rakendus, mis töötab plokiahela võrgus ja on enamasti või täielikult detsentraliseeritud. DApp-i äriloogikat juhib nutikas leping, DApp-i taustaprogramm jaotatakse ja hallatakse täielikult plokiahela platvormil. Erinevalt tsentraliseeritud serverisse juurutatud rakendusest ei ole DApp-l seisakuid ja see on saadaval seni, kuni platvorm veel töötab.

DApp-rakenduse ahelas olemus võimaldab kõigil koodi kontrollida ja selle toimimises kindlam olla. Igasugune suhtlus DApp-iga salvestatakse igaveseks plokiahelasse. Niikaua kui kasutajal on juurdepääs plokiaela sõlmele, kasutajal on alati võimalik DApp-iga suhelda ilma tsentraliseeritud juhtseadmete sekkumiseta. Ükski teenusepakkuja ega isegi nutika lepingu omanik ei saa koodi pärast selle võrku juurutamist muuta. [8]

DApp-id annavad kasutajatele parema kontrolli oma andmete üle ja kaotavad vajaduse vahendajate järele. Need võimaldavad isikutel oma andmete kasutamise ja jagamise suhtes rohkem kaasa rääkida, vähendades sõltuvust tsentraliseeritud üksustest, mis sageli kasutajate andmete pealt tulu teenivad. Kasutajad saavad DApp-e kasutama hakata, luues nendega ühenduse oma krüptorahakoti abil, ning alustada kauplemist ja muude funktsioonide täitmist ilma pikki registreerimisprotsesse läbi tegemata või isikuandmeid jagamata. [9]

Detsentraliseeritud rakendused pakuvad:

- Vea taluvus. Kui töötab üks võrk, võib detsentraliseeritud platvorm jääda kättesaadavaks, kuigi jõudlus võib olla tõsiselt häiritud. DApp-l ei ole seisakuid ja see on jätkuvalt saadaval seni, kuni plokiahel veel töötab. Kasutaja saab lepinguga otse suhelda, ilma et oleks vaja kasutajaliidest kasutada.

- Andmete terviklikkus. Plokiahelasse salvestatud andmed on muutumatud ja turvalised, kuna plokiahela konsensusalgoritmid tagavad, et plokiahelasse salvestatud andmed on muutustele vastupidavad.

- Kasutaja privaatsus. Rakendusespetsiifiliste funktsioonide kasutamiseks ei pea kasutajad DAppsile isikuandmeid esitama. See võimaldab kasutajal jääda anonüümseks.

Detsentraliseeritud rakendustel on järgmised puudused:

- Hooldus. Parandused nõuavad konsensusmehhanismi kasutamist, et tagada plokiahelapõhise võrgu kõigi partnerite vaheline kokkulepe, mis raskendab DApp-i hooldust, silumist ja värskendusi.

- Skaleeritavus. Detsentraliseeritud võrke on raskem skaleerida kui tsentraliseeritud võrke.

- Kasutajakogemus. Arendajatel võib olla raskusi DApp-i lõppkasutajatele kasutajasõbraliku kogemuse loomisega. DApp-i puhul vajavad kasutajad sisselogimiseks avalikku ja privaatvõtit, võrreldes traditsioonilise rakenduse kasutajanime ja parooliga.

2.4 Detsentraliseeritud rakenduste võimalikud ohud

Dappi kasutamisega kaasnevad ka teatud riskid, mida kasutajad peaksid selle kasutamisel arvestama. Kuna rakendus Dapp töötab plokiahelas, võib see kasutajatele teatud probleeme tekitada, kuna plokiahela suur koormus toob kaasa kalleid komisjonitasusid ja pikki tehinguaegu. Suureks probleemiks on ka turvanõrkused, eriti nutikate lepingute puhul, kus koodi vead võivad kaasa tuua rahakaotuse. On olnud ka juhtumeid, kus projektid rakendasid nutikateks lepinguteks “tagauksed”. See võib olla mõeldud

lepingutingimuste muutmiseks teatud osapoole kasuks või volitamata juurdepääsu võimaldamiseks rahalistele vahenditele, mis võib viia pettuse. Kasutajatel, kes pole varem Web3 maailmaga seotud olnud, võib detsentraliseeritud rakenduste kasutamisel tekkida raskusi, kuna need rakendused erinevad sageli tavalistest, mistõttu võivad kogenematud kasutajad kergesti petta saada ja raha kaotada.

3 Olemasolevate detsentraliseeritud krüptorahade vahetamise platvormide analüüs

Tehnoloogia arenguga suureneb ka DEX-ide arv. Iga kuu lastakse välja uusi plokiahelaid, mis omakorda eeldavad DEX-i loomist vahetuse kiiruse ja turvalisuse tagamiseks. DEX-i on 4 peamist tüüpi, millest igaühel on oma ainulaadsed omadused ja kasutusjuhud, mis vastavad krüptovaluutaga kauplejate ja investorite erinevatele vajadustele.

DEX-i on 4 peamist tüüpi:

1. Automatiseeritud turutegijad (AMM) võimaldavad kaubelda krüptovaluutadega otse müntide kogumist, mille pakuvad kasutajad, kes teenivad oma panuse eest tasu. Ostu- ja müügitellimuste sobitamise asemel kasutavad AMM-id algoritme, et määrata märgihinnad kogumi pakkumise põhjal. See süsteem tagab kõigile pideva kauplemise ja lihtsa ligipääsu likviidsusel

2. DEX-agregaatorid on platvormid, mis leiavad automaatselt krüptoga kauplemiseks parimad hinnad, otsides korraga mitut DEX-id. Need ühendavad erinevate DEX-ide likviidsuse, et pakkuda soodsaimat vahetust ja madalaimaid kauplemiskulusid. See muudab kauplemise tõhusamaks, säästes aega ja raha.

3. Cross-chain DEX-id on loodud selleks, et võimaldada kasutajatel vahetada krüproraha erinevate plokiahelate vahel. Enamik DEX on traditsiooniliselt üheahelalised, mis tähendab, et varasid saate vahetada ainult samas plokiahelavõrgus. Kui vahetate krüproraha, lukustab platvorm krüproraha ühte plokiahelasse ja vabastab teises plokiahelas olevate krüprorahade samaväärse väärtuse.

4. Tuletisinstrumendid DEX-id on detsentraliseeritud börsid, kus võimalik kaubelda finantstoodetega, nagu futuurid, mille väärtus tuleneb alusvarast. Need võimaldavad panustada krüptorahade tulevase hinnaliikumise peale, kasutades sageli võimendust potentsiaalse kasumi suurendamiseks. See annab võimaluse

investeeringute maandamiseks või hinnamuutustega spekuleerimiseks ilma tegelikku vara omamata.

3.1 Populaarsete detsentraliseeritud rakenduste ülevaade

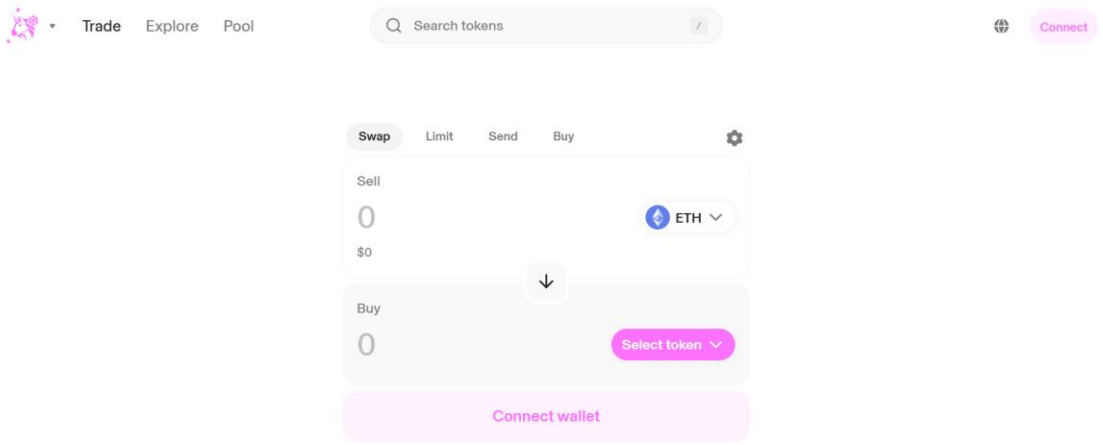
DEX-e on neli peamist tüüpi, millest igaühel on oma ainulaadsed omadused ja kasutusjuhud, mis vastavad krüptorahaga kauplejate ja investorite erinevatele vajadustele. Detsentraliseeritud börsi arendamisel otsustati kasutada olemasolevat arhitektuuri ja nutilepinguid, et minimeerida häkkimise ja rahakaotuse võimalust nii platvormi kui ka kasutajate jaoks. Nutilepingute valikul oli esmatähtis tagada turvalisus. Arendatav rakendus on AMM (automatiseeritud turuformeerija) detsentraliseeritud börs, mille nutilepingud juurutatakse Ethereum Virtual Machine (EVM) toetusega plokiahelates. Nutilepingute arhitektuuri analüüsi ja valiku etapis kaaluti ainult neid platvorme, mis samuti töötavad EVM-i baasil. See valik põhines soovil arvestada EVM-i ökosüsteemi eripäradega, sealhulgas lepingute ühilduvus, populaarsed tokenite standardid (ERC-20, ERC-721 jt), samuti tehnilised ja majanduslikud mudelid, mis on oma tõhusust juba tõestanud.

DefiLlama andmete põhjal valiti välja kaks kõige suuremat platvormi kauplemismahtude poolest: Uniswap ja Velodrome [10]. DefiLlama platvormi kasutatakse usaldusväärse teabe hankimiseks kõikide DEX-ide kauplemismahtude ja statistika kohta.

a) Uniswap

   Uniswap oli üks esimesi detsentraliseeritud finantsrakendusi, mis saavutas Ethereumis märkimisväärse veojõu. Uniswap käivitati 2018 (Joonis 1). aasta novembris. Sellest ajast alates on käivitatud palju muid detsentraliseeritud börse, kuid Uniswap on praegu kõige populaarsem. Uniswap tutvustas AMM-i mudelit 2018. aastal ja on endiselt suurim detsentraliseeritud börs. See põhineb Ethereumil ja on hiljem kasutusele võetud 11 erinevas EVM-võrgus. [10] Uniswap-protokollil on 4 versiooni. Viimane versioon avaldati augustis 2024. Uniswap V4 on automatiseeritud turutegija (AMM), mis hõlbustab tõhusat väärtusevahetust Ethereumi virtuaalmasinas (EVM). Nagu ka Uniswap-protokolli eelmiste versioonide puhul, on see mittekaitsev, seda ei saa täiendada ja lubadeta. Uniswap v4 fookuses on täiendav kohandamine arendajate jaoks ja arhitektuursed

muudatused gaasitõhususe parandamiseks, tuginedes Uniswap V1 ja V2 loodud AMM-mudelile ja Uniswap V3-s kasutusele võetud kontsentreeritud likviidsusmudelile. Selle protokolli Uniswap V3 kasutab kontsentreeritud likviidsuskogumeid, mis võimaldavad likviidsuse pakkujatel valida oma panuste jaoks konkreetsed hinnavahemikud. See süsteem suurendab likviidsust, vähendab libisemist ja parandab kõigi asjaosaliste kauplemiskogemust.



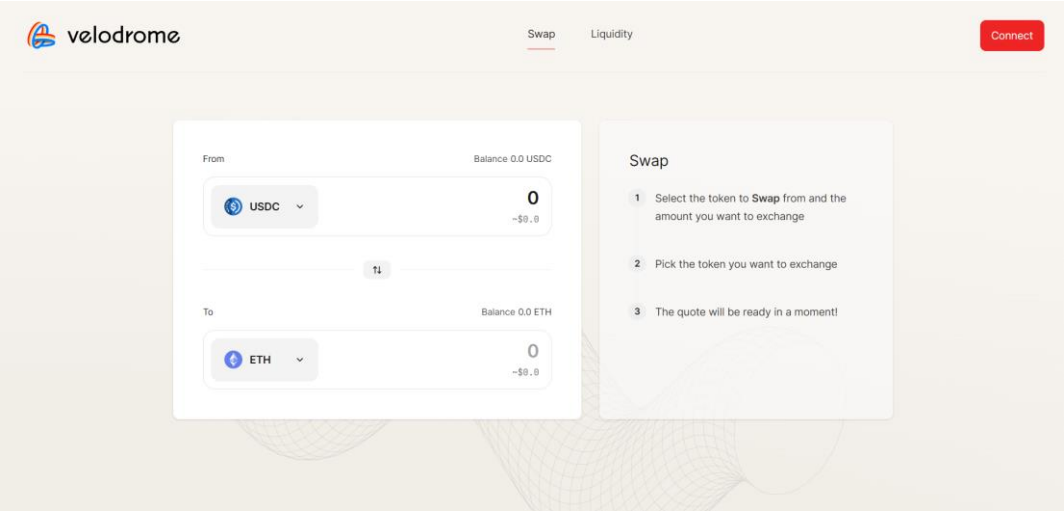
Joonis 1. Uniswap rakenduse kasutajaliidese vaade. (autori joonis)

a) Velodrome Velodrome Finance on järgmise põlvkonna AMM (automaatne turu tegija), mis ühendab Curve'i, Convexi ja Uniswapi parimad omadused ning on loodud toimima Superchaini likviidsuskeskusena (Joonis 2). Velodrome võimaldab protokollidel ehitada sügavat likviidsust kapitalitõhusal viisil, suunates $VELO emissioonid oma likviidsusfondidesse. Velodrome käivitati esmakordselt 2. juunil 2022. [11]

   Detsembris 2022 teatas Velodrome V2 versiooni avaldamisest, mis tähistas olulist sammu edasi detsentraliseeritud vahetuse tehnoloogias. V2 versiooni märkimisväärsed funktsioonid hõlmavad kontsentreeritud likviidsusfonde (clAMM), kohandatavaid tasusid ja dünaamilisi emissioonimäärasid VELO FED-i kaudu. Need funktsioonid annavad nii kauplejatele kui ka likviidsuse pakkujatele võimaluse oma strateegiaid optimeerida. Kohandatavad tasud pakuvad kauplejatele enneolematut paindlikkust. Olenemata sellest, kas eesmärk on minimaalne libisemine või garanteeritud tehingu täitmine, saavad kasutajad kohandada tasustruktuure vastavalt oma konkreetsetele vajadustele. See

dünaamiline lähenemine tasude määramisele loob konkurentsivõimelisema ja tõhusama kauplemiskeskkonna. [12]

Turvalisuse tagamiseks on lepinguid auditeerinud sellised meeskonnad nagu Spearbit, Chainsecurity ja Sherlock. Kõigi lepingute avatud lähtekood asub Github-is, mis muudab rakenduse läbipaistvamaks ning võimaldab teistel arendajatel avastada probleeme ja parandada süsteemi turvalisust.

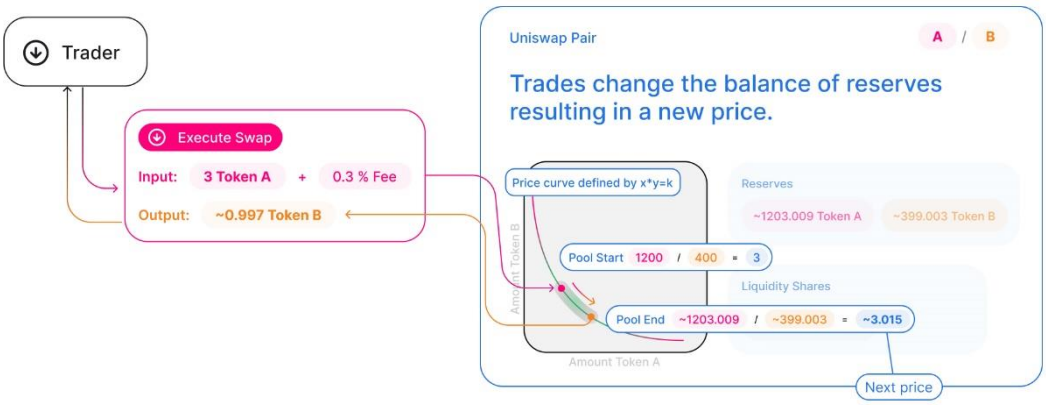


Joonis 2. Velodrome rakenduse kasutajaliidese vaade. (autori joonis)

4 Loodud detsentraliseeritud rakenduse arhitektuur

Oma rakenduse loomiseks valiti Uniswap DEX arhitektuuri, kuna nende Uniswap V2 lepingud on kasutajate jaoks kõige turvalisemad ja neid on lihtsam arendada kui Uniswap V3. Kui rääkida Uniswap V2 turvalisusest, siis kuuest insenerist koosnev meeskond vaatas üle ja kinnitas ametlikult nutikate lepingute olulised komponendid. Uniswapi meeskond viib läbi ka veapreemia, mis võimaldab turvaspetsialistidel probleeme teha. Uniswap v2 käivitati 2020. aasta mais ja sellest ajast peale pole olnud ühtegi häkkimist. Paljud AMM-id on Uniswap V2 fork [13]. Uniswap V2 on palju lihtsam kui V3, seetõttu on seda lihtsam hooldada.

Iga Uniswap nutileping või paar haldab likviidsuskogumit, mis koosneb kahe ERC-20 krüprovaluuta reservidest. Igaüks võib saada kogumi likviidsuse pakkujaks (LP), kui deponeerib iga alusmärgi samaväärse väärtuse vastutasuks kogumi valuutade eest. Need valuutad jälgivad proportsionaalseid LP-osakuid kogureservidest ja neid saab igal ajal alusvara vastu lunastada. (Joonis 3)



Joonis 3. Krüptoraha vahetuse mehhanism Uniswap v2 abil.

Selles osas analüüsiti Uniswap V2 põhikomponente (Joonis 4), et mõista, kuidas süsteem töötab. Edaspidi vaatan lähemalt iga kasutuselevõetava lepingu funktsioone.

Põhilepingute funktsioonid:

- Uniswap Core

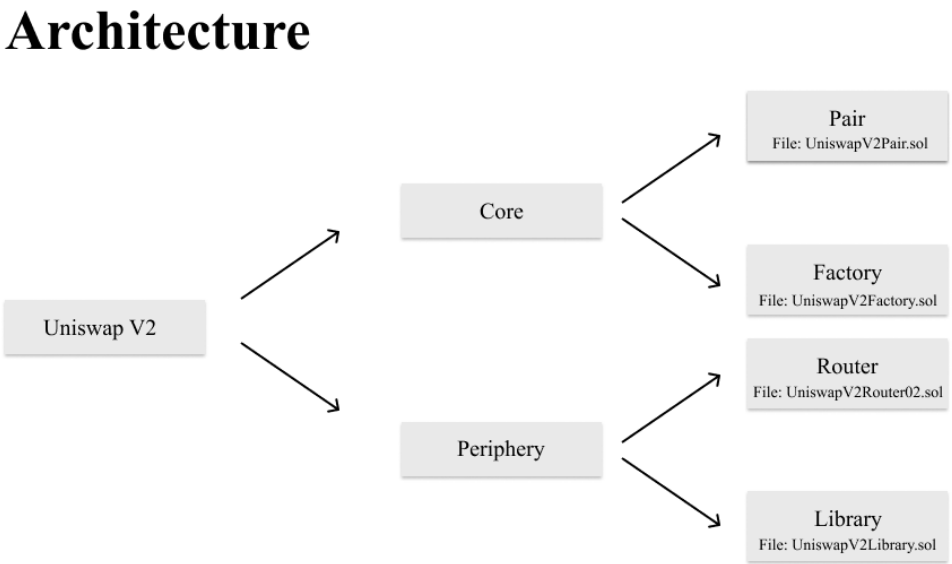
  Factory: Tehaseleping paarilepingute loomiseks ja protokollitasu saaja aadresside määramiseks

  Pair: Paarileping, mis määratleb mitmed kauplemisega seotud põhimeetodid, nagu vahetus/münt/põletamine, hinnaoraakel jne, olles ise ERC20 leping, mis pärib Uniswap V2 ERC20

- Uniswap Periphery

  Router02: Ruuteri lepingu uusim versioon, võrreldes Router01-ga, lisab see *FeeOnTransfer* raha toe; rakendab Uniswap v2 kõige sagedamini kasutatavaid liideseid, nagu likviidsuse lisamine/eemaldamine, valuutade A vahetamine B vastu, ETH vahetamine valuutade vastu

  Library: Raamatukogu leping aitab arvutada vahetustehingute ja likviidsuse juhtimise põhiväärtusi. See ei salvesta rahalisi vahendeid ega käsitle tehinguid otse, kuid pakub olulisi arvutusi, mida kasutavad nii ruuteri kui ka partnerlepingud [14]



Joonis 4. Uniswap v2 lepingu arhitektuur. (autori joonis)

4.1 Plokiahela platvormi valimine

DEX-i arendamiseks ja testimiseks valiti Sepolia võrgu. Sepolia on Ethereumi testvõrk, mis pakub arendajatele stabiilset keskkonda oma rakenduste testimiseks ja juurutamiseks enne EVM-i põhivõrgus käivitamist. Sepolia kasutab tasuta testraha, mida saab segistite kaudu hõlpsasti taotleda. See võimaldab testida nutikaid lepinguid ilma tõelise raha kaotamise riskita.

Lõplik DEX-versioon käivitati Arbitrumi võrgus. Arbitrum on Layer 2 lahendus Ethereumi jaoks, mis tagab suure skaleeritavuse ja läbilaskevõime. Kasutades "optimistic rollups" tehnoloogiat, võimaldab Arbitrum töödelda hulgaliselt tehinguid väljaspool Ethereumi põhivõrku, säilitades samal ajal kõrge detsentraliseerituse ja turvalisuse taseme. Arbitrum on näidanud kasutajaskonna märkimisväärset kasvu. 2024. aasta juunis ületas aktiivsete kuukasutajate arv 8 miljonit, mis võimaldas Arbitrumil ületada Ethereumi selles näitajas [15]. Lisaks ületas 2024. aasta oktoobris võrgus tehtud tehingute koguarv 1 miljardi piiri, mis viitab kasutajate kõrgele aktiivsusele. Need saavutused rõhutavad Arbitrumi kasvavat populaarsust ja aktsepteerimist krüptovaluutade kogukonnas.

4.2 Valmis detsentraliseeritud lepingu funktsionaalsuse ülevaade

Uniswap Core vastutab põhitoimingute haldamise eest, nagu vahendite (likviidsus), likviidsuse pakkuja (LP) žetoonide vermimine ja põletamine ning rahade vahetustehingute hõlbustamine. Siin asub põhiline äriloogika ja see on jagatud kaheks peamiseks nutikaks lepinguks: Pair ja Factory.

a) Pair Contract

   Paarileping on Uniswap v2 üks olulisemaid komponente. Igal kauplemispaaril (nt ETH/USDT) on oma spetsiaalne paarileping. See leping hoiab raha selle konkreetse kauplemispaari jaoks, hõlbustab märgivahetustehinguid ja haldab likviidsusfondi.

   Paarilepingu põhiülesanded:

- Vahetamine: hõlbustab konstantsel tootevalemil põhinevate žetoonide vahetamist. Kui kasutajad vahetavad valuuta, kohandab leping reserve selle suhte säilitamiseks.

- LP-raha vermimine: kui likviidsuse pakkujad (LP-d) lisavad kogumile likviidsust (st deponeerivad paaris mõlemad raha), antakse neile LP-raha, mis esindavad nende osa likviidsuskogumis. Selle protsessi eest vastutab *mint()* funktsioon.

- LP-raha põletamine: kui LP-d soovivad oma likviidsust eemaldada, tagastavad nad oma LP-raha, mis seejärel "põletatakse" (hävitatakse) ja vastav osa kogumi varadest kantakse funktsiooni *burn()* kaudu tagasi likviidsuse pakkujale.

- Jälgimisreservid: lepingus säilitatakse kaks võtmesaldot, üks iga paari märgi jaoks (nt ETH ja USDT). Reserve uuendatakse vahetustehingute, likviidsuse lisamise ja eemaldamise käigus.

  Paarilepingu põhifunktsioonid:

- *swap()*: teostab tegeliku märgivahetuse paaris oleva kahe märgi vahel.

- *mint():* väljastab likviidsuse pakkujatele LP-märke vastutasuks likviidsuse eest.

- *burn()*: põletab LP märgid ja kannab vastava osa likviidsusest tagasi LP-le.

b) Factory Contract

Tehase leping (factory contract) toimib kõigi üksikute paarilepingute registri- ja loomise mehhanismina. See vastutab kauplemispaaride elutsükli haldamise eest ja jälgib kõiki likviidsuskogumeid.

Tehase lepingu põhiülesanded:

- Paari loomine: iga kord, kui on vaja kaubelda kahe mündiga (nt ETH ja USDT), kasutatakse uue paarilepingu loomiseks tehaselepingut. Igal paaril on oma aadress.

- Jälgimispaarid: Tehase leping säilitab kõigi loodud paaride kirje, mis muudab olemasolevate paaride otsimise loa aadresside abil lihtsaks.

Tehase lepingu põhifunktsioonid:

- *createPair(tokenA, tokenB)*: Loob antud müntide jaoks uue likviidsuskogumi (paarileping).
- *getPair(tokenA, tokenB)*: Otsib kahe antud mündi paarilepingu aadressi, kui see on olemas.
- *allPairs()*: Tagastab kõigi tegurite loodud paariaadresside massiivi.

Periphery

Periphery koosneb nutikatest lepingutest, mis pakuvad kasutajasõbralikku liidest põhilepingutega suhtlemiseks. Kui Core tegeleb põhiloogika ja likviidsuse juhtimisega, siis Periphery lihtsustab ja abstrakteerib kasutajate suhtlust. Perifeeria kaks peamist komponenti on Router ja Library.

a) Router Contract:

Ruuteri (router) leping on liides, mis suhtleb põhilepingutega (paar ja tehas) ning pakub kasutajatele ja arendajatele hõlpsasti kasutatavaid funktsioone. Selle asemel, et sidumislepingu madala taseme funktsioonidega käsitsi suhelda, suhtlevad kasutajad tavaliselt ruuteriga.

Ruuteri lepingu põhiülesanded:

- Vahetustehingute lihtsustamine: see pakub müntide vahetamiseks lihtsustatud funktsioone (nt ETH USDT vastu), ilma et oleks vaja paarilepinguga otse suhelda. See abstraheerib vajaduse korral mitme paariga suhtlemise keerukuse.
- ETH konversiooni töötlemine: ruuter võimaldab kasutajatel vahetada ETH ja ERC-20 märke. Kuna ETH ei ole ERC-20 luba, tegeleb ruuter ETH teisendamisega ERC-20-ga ühilduvaks WETH-märgiks (Wrapped ETH).

- Likviidsuse lisamine ja eemaldamine: pakub funktsioone kogumi likviidsuse lisamiseks või eemaldamiseks (st mõlema müntide deponeerimiseks või paarilepingust väljavõtmiseks).

Ruuteri lepingu põhifunktsioonid:

- *swapExactETHForTokens()*: Võimaldab kasutajatel vahetada ETH konkreetse mündi vastu, vahetades etteantud koguse ETH nii paljude müntide vastu, kui praegune hind lubab.
- *swapTokensForExactETH()*: Võimaldab kasutajatel vahetada teatud arvu mündi fikseeritud koguse ETH vastu.
- *addLiquidity()*: Võimaldab likviidsuse pakkujatel hoiustada mõlemad mündid paaris likviidsuskogumisse.
- *removeLiquidity()*: Võimaldab likviidsuse pakkujatel oma mündid kogumist välja võtta, põletades oma LP- mündid.

b) Library Contract:

Raamatukogu (Library) leping sisaldab kasulikke funktsioone, mis aitavad arvutada vahetustehingute ja likviidsuse juhtimise põhiväärtusi. See ei salvesta rahalisi vahendeid ega käsitle tehinguid otse, kuid pakub olulisi arvutusi, mida kasutavad nii ruuteri kui ka paarislepingud.

Raamatukogu lepingu põhiülesanded:

- Reservide arvutamine: Otsib likviidsuskogumi jooksvaid reserve, mida on vaja vahetuslepingute ajal hinna arvutamiseks.
- Hinnaarvutused: Sisaldab funktsioone, mis arvutavad, kui palju ühest mündist on vaja, et vahetada teatud summa teise mündi vastu, võttes aluseks praeguse likviidsusreservi.
- Tasude korrigeerimised: arvutab vahetuslepingute tasudega korrigeeritud summad.

Raamatukogu lepingu põhifunktsioonid:

- *getReserves(pair)*: Tagastab antud paari mõlema märgi praegused reservid.
- *getAmountOut(amountIn, reserveIn, reserveOut)*: Arvestades sisendmüntide hulka ja praeguseid reserve, arvutab see funktsioon välja, kui palju väljundmünte pärast vahetust vastu võetakse.
- *getAmountIn(amountOut, reserveIn, reserveOut)*: Arvestades soovitud väljundmünte ja praeguseid reserve, arvutab see funktsioon välja, kui palju sisendmärke on vahetuse jaoks vaja.

5 Detsentraliseeritud rakenduse arendamine ja testimine

Krüptorahavahetuse veebirakenduse (DEX) arendamine Uniswap V2 lepingu arhitektuuri ja React.js kasutajaliidese abil hõlmab mitmeid olulisi samme. Vajalik nii esiprogrammi (React.js) kui ka nutika lepingu taustaprogrammi (Solidity, Uniswap V2) haldamiseks.

For developing application I will use:

- React.js (liidese arendamiseks)
- MetaMask (nutilepingute ja rakendustega suhtlemiseks)
- Hardhat (nutikate lepingute koostamiseks ja juurutamiseks)
- Solidity (nutikate lepingute kirjutamiseks)

5.1 Rakenduse arendamine

Oma rakenduse loomiseks valiti Uniswap DEX arhitektuuri, kuna nende Uniswap V2 lepingud on kasutajatele kõige turvalisemad ja arendajatele lihtsamad kui Uniswap V3. DefiLlama andmete põhjal paistab Uniswap V2 silma 670 "forks", mis on kõigist protokollidest suurim arv [13]. See näitab selle olulist mõju ja laialdast kasutuselevõttu DeFi ökosüsteemis.

Rakenduse edukaks tööks salvestatakse plokiahelas 4 lepingut - *UniswapV2ERC20, UniswapV2Factory, UniswapV2Pair, UniswapV2Router02*.

Rakenduse funktsionaalsuse edasiseks testimiseks luuakse ka standardne ERC20 token, mida saab vahetustes kasutada. Oletame, et *ETH⇔MyToken(ERC20)*.

UniswapV2ERC20 leping näeb ette paaride lepingu. Paarileping on Uniswap v2 üks olulisemaid komponente. Igal kauplemispaaril (nt ETH/USDT) on oma spetsiaalne paarileping. See leping hoiab raha selle konkreetse kauplemispaari jaoks, hõlbustab märgivahetustehinguid ja haldab likviidsusfondi.

Järgmisena luuakse UniswapV2Factory leping. Tehase leping toimib kõigi üksikute paarilepingute registri- ja loomise mehhanismina. See vastutab kauplemispaaride elutsükli haldamise eest ja jälgib kõiki likviidsuskogumeid. Leping aktsepteerib ainult ühte sisendargumenti, see on krüptorahakoti aadress, kuhu müntide vahetamisel kogutakse komisjonitasu.

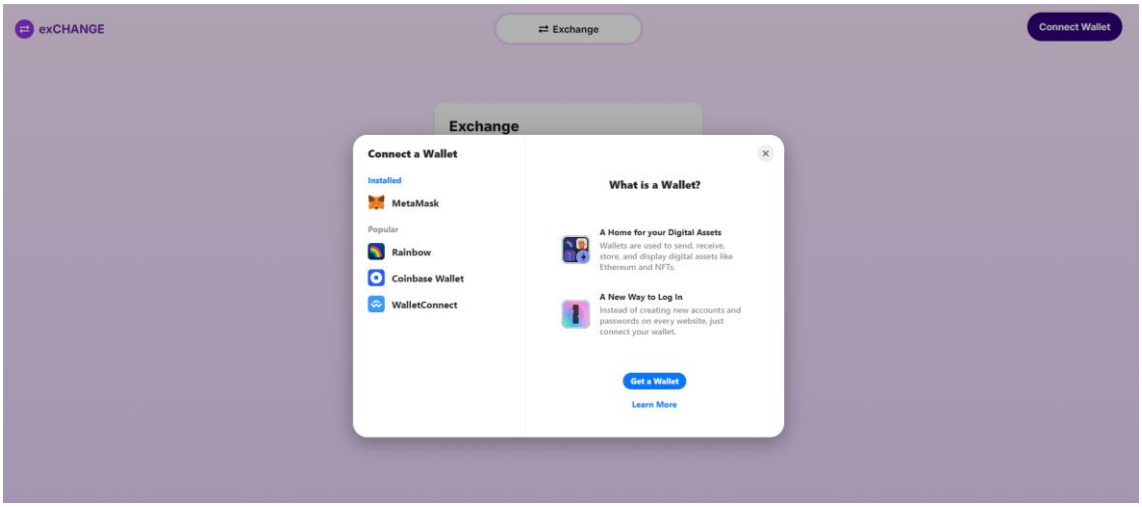
Järgmisena luuakse leping UniswapV2Router02. Lepingul on kaks argumenti: loodud UniswapV2Factory lepingu aadress ja võrgus oleva natiivse loa aadress. See aadress erineb olenevalt võrgust. Kui võtame EVM, on see aadress WETH-märgi aadress.

Esiotsa osa jaoks otsustati kasutada React.js raamistikku. Enamik Web3 rahakoti ühendamiseks mõeldud teeke ja plokiahelaga suhtlemist lihtsustavaid teeke kasutavad React.js-i.

Rahakoti ühendamiseks valiti Rainbow raamatukogu. Rainbow teek pakub konsooli käsku rakenduse kiireks juurutamiseks, mis võimaldab luua React.js rakendusi, ühendades samaaegselt Web3 rahakottide ühendamise teegi ja plokiahelaga suhtlemise teegi (wagmi teek).

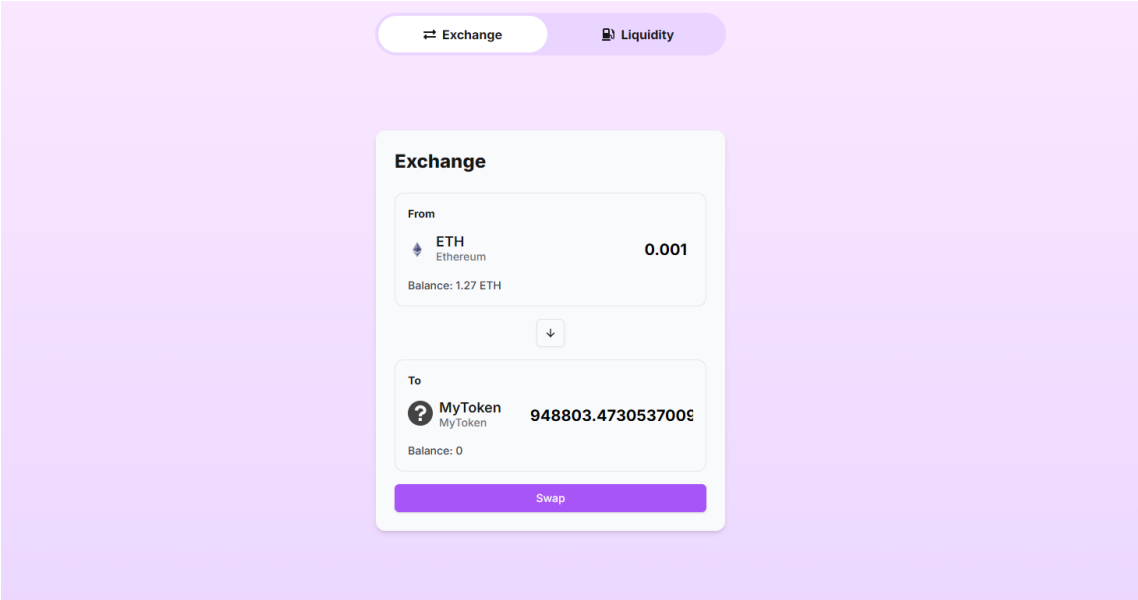
5.2 Tehtud rakenduse ulevaade

Vahetus detsentraliseeritud krüptorahavahetuse veebirakenduses (DEX) läbib mitu etappi, mis hõlmavad kasutaja suhtlemist DEX-liidese, nutikate lepingute ja plokiahelaga Vahetuse tegemiseks peab kasutajal olema paigaldatud krüptorahakott, millega on võimalik vahetust sooritada (Joonis 5). RainbowKiti kasutatav teek võimaldab valida, millised rahakotid saavad kasutajad rakendusega ühenduse luua, seega valiti kõige populaarsemad rahakotid (Metamask, Rabbit Wallet, Coinbase Wallet). Toetatakse ka WalletConnecti kaudu telefoni kaudu ühenduse loomist. WalletConnect on protokoll, mida toetavad paljud krüptovaluuta rahakotid, kuna see võimaldab hõlpsasti luua ühenduse erinevate detsentraliseeritud rahanduse (DeFi) DAppidega. Selleks kasutaja peab lihtsalt valima DA-rakenduse ja looma ühenduse QR-koodi või lingi abil. [16]



Joonis 5. Kasutajaliides krüptorahakoti ühendamiseks loodud rakendusega. (autori joonis)

Kasutaja valib krüptoraha, millega ta soovib kaubelda (ETH) ja krüptoraha, mida ta soovib saada (MyToken) (Joonis 6). Näitab krüptorahade arvu, mida ta soovib vahetada. Platvorm arvutab automaatselt välja hinnangulise krüptorahade arvu, mille kasutaja saab.



Joonis 6. Krüptorahavahetuse kasutajaliides loodud rakenduses. (autori joonis)

DEX küsib likviidsuskogumitest vahetuskursside ja likviidsuse andmeid. Selleks kutsuge UniswapV2Router02 lepingus funktsioon "*getAmountsOut()*". Kuna kõik lepingute helistamisel saadud arvväärtused on Wei-vormingus. Väärtus teisendatakse kasutajale tuttavaks väärtuseks ja kuvatakse. Pärast seda saab kasutaja vahetuse kinnitada, klõpsates nuppu „Swap”. Kasutaja kontrollib ja kinnitab rahakoti liideses vahetusandmeid.

Vahetuse lõpuleviimiseks kutsutakse UniswapV2Router02 lepingus välja funktsioon "swapExactETHForTokens" (Joonis 7).

```
export const swapExactETHForTokens = async (
  amountOutMin, // Minimum amount of tokens you expect
  path, // Array of token addresses (e.g., [WETH_ADDRESS, TOKEN_ADDRESS])
  to, // Recipient address (your wallet address)
  deadline, // Unix timestamp (transaction deadline)
  payableAmount, // ETH to send (as a string, in ether)
) => {
  try {
    const provider = new ethers.providers.Web3Provider(window.ethereum)
    const signer = provider.getSigner()

    const routerContract = new ethers.Contract(
      ROUTER_ADDRESS,
      ROUTER_ABI,
      signer,
    )

    const tx = await routerContract.swapExactETHForTokens(
      amountOutMin,
      path,
      to,
      deadline,
      {
        value: ethers.utils.parseEther(payableAmount),
      },
    )

    // Wait for the transaction to be mined
    const receipt = await tx.wait()

    // Return transaction receipt after it's mined
    return receipt
  } catch (error) {
    console.error('Error swapping tokens:', error)
    throw error
  }
}
```

Joonis 7. Krüptoraha vahetamise programmikood. (autori joonis)

DEX-i nutikas leping suhtleb likviidsuskogumiga, et määrata kindlaks kõige kasumlikum vahetustee (kaasa arvatud marsruudid läbi teiste žetoonide, kui see vähendab kulusid). Tokenid debiteeritakse kasutaja rahakotist ja saadetakse basseini. Tokenid, mida kasutaja soovib saada, saadetakse talle likviidsusest. Pärast seda salvestatakse tehing plokiahelasse ja plokiahel kinnitab tehingu. Pärast plokki kaasamist saab kasutaja kinnituse. Vahetus on lõppenud ja märgid saabuvad rahakotti.

5.3 Funktsionaalsuse ja turvalisuse testimine

Ehitatud detsentraliseeritud krüptorahavahetuse testimine on kriitiline protsess, mis hõlmab funktsionaalsuse, turvalisuse, jõudluse ja kasutajakogemuse kontrollimist. Selleks, et tagada rakenduse ja lepingute funktsionaalsus. Detsentraliseeritud börsi testimiseks valiti Sepolia võrgu. Sepolia kasutab tasuta testmärke, mida saab segistite kaudu hõlpsasti taotleda. See võimaldab testida nutikaid lepinguid ilma tõelise raha kaotamise riskita.

5.3.1 Unit testimine

Testimine on nutilepingute arendamise protsessi oluline osa, tagades nende töökindluse, turvalisuse ja prognoositavuse. Üks arendatava rakenduse kohustuslikest nõuetest oli turvalisus, mistõttu mängib testimine võtmerolli. Kuna lepingud on pärast nende juurutamist plokiahelas muutumatud, vajavad need põhjalikku kontrollimist, sest igasugune koodiviga võib kaasa tuua tõsiseid tagajärgi, sealhulgas kasutajate vahendite kaotust või oluliste funktsioonide täitmise võimatust.

Loodud lepingute testimiseks loodi automaatsete testide süsteem, mis kasutab kaasaegseid tööriistu nutilepingute kontrollimiseks. Testid kirjutati JavaScriptis, kasutades Mocha raamistikku, Chai ja Chai-as-promised teegid. Testide täitmiseks kasutatakse Hardhati keskkonda, mis pakub plokiahela turvaliseks täitmiseks ja silumiseks. [17] Lisaks sellele kasutatakse ethers.js teek lepingutega suhtlemiseks ja kontode haldamiseks. [18] (vt. lisa 2).

Skriptis kirjeldati peamised stsenaariumid loodud lepingutega suhtlemiseks tokenite vahetamiseks. Kuna põhifunktsionaalsus toimub Router-lepingu kaudu, kirjutati testid sellele lepingule. Esimeses testis kontrollitakse edukat vahetustehingut. Esiteks antakse WETHi tehingule heakskiit Router-lepingu jaoks. Seejärel kutsutakse funktsioon swapExactETHForTokens, et vahetada ETH (Ethereum) TestTokeni vastu. Veendudes, et ETH saldo väheneb ja TestTokeni saldo suureneb, kontrollitakse tehingu õigsust.

Teises testis toimub sarnane vahetus, kuid vastupidises suunas. Enne vahetuse algust antakse TestTokeni vahetamiseks Router-lepingule heakskiit. Seejärel kutsutakse funktsioon swapExactTokensForETH, et vahetada TestToken ETH (Ethereum) vastu. TestTokeni saldo vähenemist ja ETH saldo suurenemist kontrollides veendutakse vahetuse korrektsuses.

Kolmandas testis simuleeritakse olukorda, kus vahetuskatse tehakse ilma Router-lepingule eelneva heakskiiduta. See peab viima veani, kuna Router-leping ei saa hallata kasutaja vahendeid ilma vastava loata.

Neljas test demonstreerib olukorda, kus vahetuse maht ületab kasutaja rahakotis olevate tokenite saldo. Sellise vahetuskatse tegemine peab lõppema ebaõnnestumisega, kuna pole võimalik edastada rohkem tokeneid, kui rahakotis on.

Viies test uurib süsteemi käitumist, kui proovida vahetada null arvu tokeneid. See juhtum peab lõppema veaga, kuna tokeniteta vahetus pole mõttekas ja on reeglitega vastuolus.

Kõik testid viidi edukalt läbi, mis kinnitab, et vahetamise eest vastutavad lepingute põhifunktsioonid töötavad ootuspäraselt. Tehingud on turvalised, vead käsitletakse õigesti ja suhtlusele seatud piirangud täidetakse. Testide edukas läbimine kinnitab süsteemi valmisolekut reaalseks kasutamiseks ja edasiseks arendamiseks. Selline lähenemine tagab arendatava rakenduse kõrge kvaliteedi ja turvalisuse.

5.3.2 Funktsionaalne testimine

Funktsionaalne testimine on testimise tüüp, mille eesmärk on kindlaks teha, kas iga rakenduse funktsioon töötab vastavalt tarkvara nõuetele. Iga funktsiooni võrreldakse vastava nõudega, et veenduda, kas selle väljund vastab lõppkasutaja ootustele. Testimine toimub, pakkudes näidissisendeid, salvestades saadud väljundid ja kontrollides, kas tegelikud väljundid vastavad oodatud väljunditele. [19]

Funktsionaalse DEX-i testimise esimene samm on kasutajate suhtluse kontrollimine platvormiga, sealhulgas nende rahakottide, näiteks MetaMaski, ühendamine. Oluline on kontrollida, et kasutaja saab platvormiga edukalt ühenduse luua, näha oma vara ja suhelda vahetusliidesega. Seetõttu kontrolliti kõiki rahakotte ühendamise osas. Samuti oli väga oluline kontrollida võrku, millega rakendus ühendub, sest valesti ühendatud võrk põhjustab hiljem tõrkeid.

DEX-i põhifunktsioon seisneb krüptovaluutade vahetamises. Seetõttu oli oluline kontrollida eeldatava valuutakoguse arvutamise õigsust, mille kasutaja saab. Kuna nutilepingud tagastavad väärtusi kindlas vormingus, on oluline veenduda, et need andmed tõlgendatakse ja teisendatakse õigesti kasutajasõbralikeks väärtusteks. ERC-20 standardit kasutavate tokenitega töötamisel on oluline arvestada kümnendkohtade arvu, mis võib iga

tokeni puhul erineda. Näiteks võib ühel tokenil olla 18 kümnendkohta, teisel aga 6, mis mõjutab seda, kuidas väärtused teisendatakse ja kuvatakse.

Tokenitega õigeks suhtlemiseks on esmalt vaja hankida kümnendkohtade arv ja seejärel teisendada tokenite kogus kasutajasõbralikku vormingusse. Kui tokenil on 18 kümnendkohta, siis säilitatakse selle väärtused minimaalsetes ühikutes(wei) ja kasutajale kuvamiseks tuleb see arv jagada 10^{18}-ga. Kui tokenil on 6 kümnendkohta, siis on see 10^{6} ja arv tuleb jagada 1,000,000-ga. Decimal-väärtuse saamiseks tokenilt tuleb kutsuda tokeni lepingu funktsioon *decimals()*. Seejärel saab, kasutades saadud decimals väärtust, teisendada arvu loetavasse vormingusse, kasutades Ethers.js teek.

Tokenitega töötamisel on oluline kontrollida luba toiminguteks tokenitega. Selleks tuleb kontrollida, kas Router-nutilepingul on piisavalt luba ERC-20 tokenite kasutamiseks. Kui kasutaja soovib vahetada rohkem tokeneid, kui Routeril on luba kasutada, tuleb käivitada tehing tokenite kinnitamiseks

Lõppkokkuvõttes aitab DEX-i funktsionaalne testimine veenduda, et kogu süsteem töötab ühtse tervikuna, arvestades kõiki vahetusmehhanisme, likviidsust, libisemist, tasusid ja limiite. Kõigi testide edukas läbiviimine kinnitab, et platvorm on kasutusvalmis ja pakub kasutajatele turvalist ja tõhusat viisi krüptovaluutade vahetamiseks.

6 Tulemused

Rakendust, mis loodi lõputöö raames, saab hinnata vastavalt sellele, kui hästi see vastab kehtestatud nõudmistele ja saavutab lõputöö eesmärgid. Lisaks pakutakse käesolevas jaotises rakenduse edasise arendamise võimalusi, kuna autoril on laiem ülevaade sellest, milliseks see rakendus võib tulevikus kujuneda.

6.1 Rakenduse lõpptulemus

Lõputöö kirjutamise käigus töötati välja detsentraliseeritud börs krüptovaluutade vahetamiseks, mis põhineb plokiahelatehnoloogial. Arendatud rakendus on platvorm, mis võimaldab kasutajatel teostada krüptovaluutade vahetusi turvaliselt ja läbipaistvalt. Rakenduse peamine ülesanne oli pakkuda kasutajatele kiireid ja turvalisi krüptovaluutavahetusvõimalusi. Loodud rakendus võimaldab kasutajatel ühendada oma krüptorahakoti. Valikus on kõige populaarsemad rahakotid, sealhulgas mobiilsete rahakottide ühendamine WalletConnecti kaudu. Vahetus toimub läbi suhtluse nutilepinguga, mis on salvestatud plokiahelasse. Vahetusplatvormi funktsionaalsuse tagamiseks on juurutatud 4 nutilepingut, millest igaüks vastutab teatud ülesannete täitmise eest. Need lepingud tagavad vahetustehingute teostamise, likviidsuse haldamise, paaride loomise ning muud olulised funktsioonid rakenduse edukaks tööks.

Samuti on oluline aspekt, et rakendus on detsentraliseeritud, mis tähendab, et kasutajad saavad oma vahendeid hallata ja vahetusi teha otse lepingutega, isegi kui rakenduse esiotsa osa on välja lülitatud. Seda tagab nutilepingute kasutamine, mis on salvestatud plokiahelasse ja toimivad sõltumatult välistest liidestest.

Erilist tähelepanu on pööratud mugava ja intuitiivse kasutajaliidese loomisele, et minimeerida uute kasutajate sisene misläve. Vahetusprotsessi käigus valib kasutaja krüptovaluuta, mida ta soovib vahetada, ja krüptovaluuta, mida ta soovib saada. Loodud rakendus arvutab automaatselt eeldatava krüptovaluuta koguse, mida kasutaja saab. Rakendus küsib likviidsuspuljadelt andmed valuutakursi ja likviidsuse kohta. Seejärel kuvab süsteem praeguse vahetuskurssi, mis põhineb plokiahelast saadud andmetel, ja

arvutab lõppsumma, arvestades määratud tehingu mahtu. Autor usub, et projekti tulemus on hea alus edasiseks arendamiseks ja skaleerimiseks. Loodud rakendus on oluline samm teel suuremahulise detsentraliseeritud ja turvalise finantsvahetuse idee elluviimiseks.

6.2 Rakenduse edasiarendus

Lõputöö raames loodi detsentraliseeritud börs krüptovaluutade vahetamiseks. Kuigi rakendus täidab oma algsed ülesanded, on olemas piisavalt võimalusi rakenduse edasiseks arendamiseks ja täiustamiseks. Oluline samm on likviidsuse lisamise liidese elluviimine. Praegu saavad kasutajad lisada likviidsust otse lepingus, kuid likviidsuse haldamise mugavuse tagamiseks tuleks rakendusse lisada võimalus lisada ja eemaldada likviidsust otse rakenduse liidesest. See võimaldab algajatel kasutajatel lisada likviidsust müntide jaoks otse rakenduse liidese kaudu.

Teine võimalik funktsioon, mida võiks rakendada, on võimalus võimaldada kasutajatel teostada vahetusi erinevate plokiahelate vahel. Kross-chain vahetuste rakendamine, kus kasutaja saadab tokeneid ühes võrgus ja saab teised mündid teises võrgus, suurendab rakenduse atraktiivsust. See laiendab kasutajate võimalusi ja muudab börsi konkurentsivõimelisemaks detsentraliseeritud rahanduse turul.

Detsentraliseeritud börsi edasine arendamine ja konkurentsivõime suurendamine on oluline samm tokeniseerimise elluviimine. Oma tokeni loomine ja rakendamine avab laiad perspektiivid funktsionaalsuse täiustamiseks ja uute kasutajate kaasamiseks. Tokeniseerimine võimaldab luua premeerimissüsteemi börsi kasutajatele. Võimalus anda preemiaid likviidsuse pakkumise ja kauplemismahtude eest teeb börsi atraktiivsemaks likviidsuse pakkujate jaoks, mis omakorda suurendab kauplemismahtusid ja vähendab vahetusspredšide suurust.

7 Kokkuvõte

Käesoleva lõputöö eesmärk oli arendada detsentraliseeritud krüptoraha vahetusplatvormi, et pakkuda kasutajatele turvalisemat ja läbipaistvamat kauplemiskeskkonda. Selle eesmärgi saavutamiseks uuriti plokiahela tööpõhimõtteid. Edukaks plokiahelaga töötava rakenduse arendamiseks on oluline sügav arusaam plokiahelas toimuvate tehingute põhielementidest. Teoreetilises osas analüüsiti olemasolevaid detsentraliseeritud krüptorahavahetusi. Analüüsi käigus valiti kasutamiseks Uniswapi platvormi nutilepingud, kuna need lepingud on kõige sagedamini kasutatavad ja turvalisemad.

Arendusprotsessi käigus juurutatud valitud nutilepingud plokiahelas, mis tagavad platvormi põhifunktsionaalsuse, samuti töötati välja kasutajaliides, mis võimaldab kasutajatel lepingutega suhelda. Kasutajaliides on kujundatud nii, et see oleks intuitiivne ja ligipääsetav kõigile kasutajatele, sealhulgas neile, kellel puudub varasem kogemus krüptovaluuta vahetamisega.

Detsentraliseeritud krüptorahavahetuse rakendamiseks kasutati kaasaegseid tehnoloogiaid: nutilepingud kirjutati Solidity programmeerimiskeeles ja implementeeriti Hardhat tööriista abil, kasutajaliidese loomiseks kasutati React.js raamistiku, Ethers.js teeki plokiahelaga suhtlemiseks ning Rainbowkit'i krüptorahakoti rakendusega ühendamiseks. Arenduse ja testimise käigus kasutati Sepolia testvõrku, mis võimaldas teostada tehinguid testvaluutaga, millel ei ole reaalselt rahalist väärtust.

Töö käigus testiti rakenduse võtmekomponendid. Testitud lepingud implementeeriti Arbitrumi plokiahelas, mis pakub kasutajatele eeliseid madalate tehingutasude ja kiire tehingutöötluse näol. Kuigi rakendus täidab algsed ülesanded, on selle edasiarendamiseks ja täiustamiseks palju võimalusi. Tulevikus võib loodud rakendus kujuneda täisväärtuslikuks tooteks, mis suudab turul konkureerida ja pakkuda kasutajatele kvaliteetset vahendit krüptovaluutade vahetamiseks.

Kasutatud kirjandus

[1] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, 2008.

[2] „Russian Nationals Charged With Hacking One Cryptocurrency Exchange and Illicitly Operating Another,“ United States government, September 2023. [Võrgumaterjal]. Saadaval: https://www.justice.gov/opa/pr/russian-nationals-charged-hacking-one-cryptocurrency-exchange-and-illicitly-operating-another. [Kasutatud 15 Oktoober 2024].

[3] WazirX, „WazirX Bounty Program – Plan and Structure,“ November 2024. [Võrgumaterjal]. Saadaval: https://wazirx.com/blog/wazirx-bounty-program/. [Kasutatud 14 Oktoober 2024].

[4] U. S. government, „FTX Founder Indicted for Fraud, Money Laundering, and Campaign Finance Offenses,“ 13 Detsember 2022. [Võrgumaterjal]. Saadaval: https://www.justice.gov/opa/pr/ftx-founder-indicted-fraud-money-laundering-and-campaign-finance-offenses. [Kasutatud 4 Oktoober 2024].

[5] Amazon, „What is Blockchain Technology?,“ [Võrgumaterjal]. Saadaval: https://aws.amazon.com/what-is/blockchain. [Kasutatud 20 Oktoober 2024].

[6] D. Tapscott ja A. Tapscott, World, Blockchain Revolution: How the Technology Behind Bitcoin and Other Cryptocurrencies Is Changing the, Portfolio, 2018.

[7] Coinbase, „What are Decentralized Applications (DApps)?,“ [Võrgumaterjal]. Saadaval: https://www.coinbase.com/learn/crypto-basics/what-are-decentralized-applications-dapps. [Kasutatud 12 Oktoober 2024].

[8] A. Antonopoulos ja G. Wood, Mastering Ethereum: Building Smart Contracts and DApps, O'Reilly Media, 2019.

[9] B. Academy, „Detsentraliseeritud rakendus (DApp),“ [Võrgumaterjal]. Saadaval: https://academy.binance.com/et/glossary/decentralized-application. [Kasutatud 10 November 2024].

[10] C. C. F. Innovation, „What is Uniswap?,“ 13 Oktoober 2022. [Võrgumaterjal]. Saadaval: https://cryptoforinnovation.org/what-is-uniswap/. [Kasutatud 17 Oktoober 2024].

[11] Velodrome, „About Velodrome,“ [Võrgumaterjal]. Saadaval: https://velodrome.finance/docs. [Kasutatud 16 Detsember 2024].

[12] Okx, „What is Velodrome Finance (VELO): why it's a next-gen AMM,“ [Võrgumaterjal]. Saadaval: https://www.okx.com/learn/what-is-velodrome-finance. [Kasutatud 16 Detsember 2024].

[13] Defillama, „DEX Forks,“ [Võrgumaterjal]. Saadaval: https://defillama.com/forks. [Kasutatud 15 Detsember 2024].

[14] A. Shao, „Deep Dive into Uniswap v2 Smart Contracts,“ 25 Juuni 2024. [Võrgumaterjal]. Saadaval: https://github.com/adshao/publications/blob/master/uniswap/dive-into-uniswap-v2-contracts/README.md. [Kasutatud 24 Oktoober 2024].

[15] Binance, „Arbitrum Surpasses Ethereum In Monthly Active Users,“ [Võrgumaterjal]. Saadaval: https://www.binance.com/en/square/post/2024-06-03-arbitrum-surpasses-ethereum-in-monthly-active-users-8957458312602?utm_source=chatgpt.com. [Kasutatud 15 Detsember 2024].

[16] B. Academy, „How to Use WalletConnect,“ 14 Veebruar 2024. [Võrgumaterjal]. Saadaval: https://academy.binance.com/en/articles/how-to-use-walletconnect. [Kasutatud 10 November 2024].

[17] „Testing contracts,“ 12 Märts 2024. [Võrgumaterjal]. Saadaval: https://hardhat.org/tutorial/testing-contracts. [Kasutatud 10 Detsember 2024].

[18] „Ethers Documentation,“ 23 Aprill 2023. [Võrgumaterjal]. Saadaval: https://docs.ethers.org/v5/. [Kasutatud 12 Detsember 2024].

[19] Opentext, „What is Functional Testing?,“ [Võrgumaterjal]. Saadaval: https://www.opentext.com/what-is/functional-testing. [Kasutatud 11 Detsember 2024].

Lisa 1 – Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks[1]

Mina, Artyom Strelkov

1. Annan Tallinna Tehnikaülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Detsentraliseeritud krüptorahavahetuse veebirakenduse arendamine“, mille juhendaja on Lembit Viilup
   1.1. reprodutseerimiseks lõputöö säilitamise ja elektroonse avaldamise eesmärgil, sh Tallinna Tehnikaülikooli raamatukogu digikogusse lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
   1.2. üldsusele kättesaadavaks tegemiseks Tallinna Tehnikaülikooli veebikeskkonna kaudu, sealhulgas Tallinna Tehnikaülikooli raamatukogu digikogu kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. Olen teadlik, et käesoleva lihtlitsentsi punktis 1 nimetatud õigused jäävad alles ka autorile.
3. Kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest ning muudest õigusaktidest tulenevaid õigusi.

26.12.2024

1 Lihtlitsents ei kehti juurdepääsupiirangu kehtivuse ajal vastavalt üliõpilase taotlusele lõputööle juurdepääsupiirangu kehtestamiseks, mis on allkirjastatud teaduskonna dekaani poolt, välja arvatud ülikooli õigus lõputööd reprodutseerida üksnes säilitamise eesmärgil. Kui lõputöö on loonud kaks või enam isikut oma ühise loomingulise tegevusega ning lõputöö kaas- või ühisautor(id) ei ole andnud lõputööd kaitsvale üliõpilasele kindlaksmääratud tähtajaks nõusolekut lõputöö reprodutseerimiseks ja avalikustamiseks vastavalt lihtlitsentsi punktidele 1.1. ja 1.2, siis lihtlitsents nimetatud tähtaja jooksul ei kehti.

Lisa 2 – Testide programmikoodi näidis

```
describe('Uniswap Swap', function() {
  let signer;
  let provider;
  let contracts;

  before(async function() {
    signer = await getSigner();
    provider = hre.ethers.provider;

    contracts = {
      router: getContractInstance(CONTRACT_ADDRESSES.ROUTER, routerArtifact,
signer),
      testToken: getContractInstance(CONTRACT_ADDRESSES.TEST_ERC20,
erc20Artifact, signer),
      weth: getContractInstance(CONTRACT_ADDRESSES.WETH, wethArtifact,
signer),
    };

    // Send transaction to wrap 1 Ether into WETH
    const tx = await contracts.weth.deposit({ value: ethers.parseEther('1')
});
    await tx.wait();

    const wethBalance = await contracts.weth.balanceOf(signer.address);
    assert.equal(hre.ethers.formatEther(wethBalance), '1.0', 'Initial WETH
balance should be 1.0');
  });

  it('Should perform a successful swap', async function() {
    const amountIn = hre.ethers.parseEther('1');
    await executeSwap(provider, signer, contracts, amountIn);

    const wethBalance = await contracts.weth.balanceOf(signer.address);
    const testTokenBalance = await
contracts.testToken.balanceOf(signer.address);

    assert.isAbove(parseFloat(hre.ethers.formatEther(wethBalance)), 0.0,
'WETH balance should have decreased');
    assert.isAbove(parseFloat(hre.ethers.formatUnits(testTokenBalance, 18)),
0, 'TEST_ERC20 balance should have increased');
  });

  it('Should fail when trying to swap without approval', async function() {
    const amountIn = hre.ethers.parseEther('1');

    await assert.isRejected(
      contracts.router.swapExactETHForTokens(
        amountIn,
        0,
```

```
        [CONTRACT_ADDRESSES.WETH, CONTRACT_ADDRESSES.TEST_ERC20],
        signer.address,
        Math.floor(Date.now() / 1000) + (60 * 10)
      ),
      /revert/,
      'Swap should fail without approval'
    );
  });

  it('Should fail when trying to swap more than the balance', async
function() {
    const amountIn = hre.ethers.parseEther('2'); // Greater than the initial
balance of 1
    const tx1 = await contracts.weth.approve(CONTRACT_ADDRESSES.ROUTER,
amountIn);
    await tx1.wait();

    await assert.isRejected(
      contracts.router.swapExactETHForTokens(
        amountIn,
        0,
        [CONTRACT_ADDRESSES.WETH, CONTRACT_ADDRESSES.TEST_ERC20],
        signer.address,
        Math.floor(Date.now() / 1000) + (60 * 10)
      ),
      /revert/,
      'Swap should fail when trying to swap more than the balance'
    );
  });

  it('Should fail if the deadline has passed', async function() {
    const amountIn = hre.ethers.parseEther('1');

    const tx1 = await contracts.weth.approve(CONTRACT_ADDRESSES.ROUTER,
amountIn);
    await tx1.wait();

    try {
      await contracts.router.swapExactETHForTokens(
        amountIn,
        0,
        [CONTRACT_ADDRESSES.WETH, CONTRACT_ADDRESSES.TEST_ERC20],
        signer.address,
        Math.floor(Date.now() / 1000) - 60, // Setting the deadline to 1
minute in the past
      );
    } catch (error) {
      console.log(error);
      assert.fail('Swap should fail if the deadline has passed');
    }
  });
```

```
  it('Should fail when trying to swap 0 tokens', async function() {
    const amountIn = hre.ethers.parseEther('0'); // Trying to swap 0 WETH

    // First, approve the router to transfer WETH
    const tx1 = await contracts.weth.approve(CONTRACT_ADDRESSES.ROUTER,
amountIn);
    await tx1.wait();

    await assert.isRejected(
      contracts.router.swapExactETHForTokens(
        amountIn,
        0,
        [CONTRACT_ADDRESSES.WETH, CONTRACT_ADDRESSES.TEST_ERC20],
        signer.address,
        Math.floor(Date.now() / 1000) + 60
      ),
      /revert/,
      'Swap should fail when trying to swap 0 tokens'
    );
  });
});
```