НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики
Кафедра системного програмування
і спеціалізованих комп'ютерних систем

«На правах рукопису»
УДК 004.9

До захисту допущено:
Завідувач кафедри
_____Віталій РОМАНКЕВИЧ
«___»_____20__р.

Магістерська дисертація

на здобуття ступеня магістра
за освітньо-професійною програмою «Системне програмування та спеціалізовані комп'ютерні системи»
зі спеціальності 123 «Комп'ютерна інженерія»
на тему: «Децентралізовані веборієнтовані засоби для контролю та виконання транзакцій із криптовалютою з використанням технології Blockchain»

Виконав:
студент ІІ курсу, групи КВ-12мп
Шевченко Дмитро Сергійович _____

Науковий керівник:
к.т.н., доц. каф. СПСКС
Клятченко Ярослав Михайлович _____

Рецензент:
_____
_____ _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань. Студент _____

Київ – 2022 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра системного програмування

і спеціалізованих комп‘ютерних систем

Рівень вищої освіти – другий (магістерський)

Спеціальність – 123 «Комп‘ютерна інженерія»

Освітньо-професіна програма «Системне програмування та спеціалізовані комп‘ютерні системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____Віталій РОМАНКЕВИЧ

«___»_____20___р.

ЗАВДАННЯ

на магістерську дисертацію студенту

Шевченку Дмитру Сергійовичу

1. Тема дисертації «**Децентралізовані веборієнтовані засоби для контролю та виконання транзакцій із криптовалютою з використанням технології Blockchain**», науковий керівник дисертації Клятченко Ярослав Михайлович, к.т.н., доц. каф. СПСКС, затверджені наказом по університету від 25.11.2022 р. № 4314-С

2. Термін подання студентом дисертації _____

3. Об’єкт дослідження
   - технологія Blockchain та її застосування в веборієнтованих засобах для контролю та виконання транзакцій з криптовалютою.

4. Вихідні дані
   - засоби для контролю та виконання транзакцій із криптовалютою з використанням технології Blockchain

5. Перелік завдань, які потрібно розробити
   - проаналізувати технологію Blockchain
   - реалізувати децентралізований веборієнтований засоб для контролю та виконання транзакцій із криптовалютою з використанням технології Blockchain

6. Орієнтовний перелік графічного (ілюстративного) матеріалу
   -презентація

7. Орієнтовний перелік публікацій
   - Прикладна математика та комп’ютинг. «XIV науково-практична конференція магістрантів та аспірантів ПМК-2022 факультету прикладної математики»
   - Міжнародна науково-технічна Internet-конференція у Національному університеті харчових технологій

8. Дата видачі завдання _01.10.2021

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Вивчення літератури за тематикою МД	01.10.2021	
2.	Розроблення та узгодження технічного завдання	05.10.2021	
3.	Аналіз існуючих рішень	14.12.2021	
4.	Підготовка матеріалів першого розділу МД	18.05.2022	
5.	Розробка програмного забезпечення	16.09.2022	
6.	Відлагодження програмного продукту	14.10.2022	
7.	Підготовка матеріалів другого розділу МД	19.10.2022	
8.	Розроблення інтерфейсу програмного забезпечення	26.10.2022	
9.	Оформлення документації МД	29.11.2022	
10.	Проходження попереднього захисту	01.12.2022	

Студент Дмитро ШЕВЧЕНКО

Науковий керівник Ярослав КЛЯТЧЕНКО

РЕФЕРАТ

Актуальність теми. В сучасному світі інтернет є невід'ємною частиною життя кожної людини. Сьогодні неможливо уявити будь-яку сферу бізнесу, навчання, надання послуг, або реклами без використання вебресурсів таких, як вебдодатки або вебсторінки, соціальні мережі, месенджери тощо. Web 3.0 представляє собою нову ітерацію або фазу еволюції Інтернету. Розвиток Web 3.0 розпочався в 2010 році і триває зараз. Web 3.0 побудовано на основних концепціях децентралізації, відкритості та більшої зручності для користувачів. Завдяки децентралізації можна вирішити проблеми непрозорості вебсервісів, цензури у мережі та конфіденційності особистих даних, що є надважливою проблемою в поточному Web 2.0. Так як Blockchain є фундаментальною концепцією для нового покоління вебресурсів, та передача даних в блокчейн-мережі проходить миттєво, його головна сфера його використання це фінанси. Криптовалюта вже зараз являється повноцінним платіжним засобом, тому створення зручного децентралізованого веборієнтованого засобу для проведення операцій над криптовалютою, використовуючи технологію Blockchain, дозволить вирішити проблему конфіденційності, безпечної авторизації та приватності, при проведенні фінансових транзакцій.

Об'єктом дослідження є технологія Blockchain та її застосування в веборієнтованих засобах для контролю та виконання транзакцій з криптовалютою.

Предметом дослідження є веборієнтоване програмне забезпечення для контролю та виконання транзакцій з криптовалютою з використанням технології Blockchain.

Мета роботи: розробити децентралізований веборієнтований засіб для контролю та виконання транзакцій з криптовалютою, який би був швидким у роботі у порівнянні з централізованими засобами проведення фінансових транзакцій, прозорим завдяки використанню децентралізованої структури, надійним та конфіденційним для користувачів через авторизацію за допомогою криптогаманця.

Наукова новизна полягає в наступному: запропоновано децентралізований веборієнтований засіб для контролю та виконання транзакцій з криптовалютою. Перевагою такого засобу є те, що завдяки технології Blockchain, він є швидким та прозорим, надійним та конфіденційним. Крім того децентралізований додаток ніколи не припиняє роботу завдяки відсутності центрального сервера, який міг би зупинитися. Дані в ньому розподілені між вузлами, при цьому всі вузли діють незалежно, і якщо один з них зупиниться, інші продовжать роботу в мережі.

Практична цінність отриманих в роботі результатів полягає в тому, що розроблений вебдодаток дозволяє авторизуватися за допомогою свого криптогаманця, що є покращує конфіденційність, безпечно та швидко виконувати транзакцій по переводу криптовалют із одного гаманця на інший, зручно контролювати виконання транзакцій.

Апробація роботи. Основні положення були представлені та обговорювались на XIV науковій конференції магістрантів та аспірантів «Прикладна математика та комп'ютинг» ПМК-2022 (Київ, листопад 2022 р.) та Міжнародній науково-технічній Internet-конференція у Національному університеті харчових технологій.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, чотирьох розділів та висновків.

У вступі подано загальну характеристику роботи, зроблено оцінку сучасного стану проблеми, обґрунтовано актуальність напрямку досліджень, сформульовано мету і задачі досліджень, показано наукову новизну отриманих результатів і практичну цінність роботи, наведено відомості про апробацію результатів і їхнє впровадження.

У першому розділі розглянуто існуючі програмні засоби для контролю та виконання транзакцій з криптовалютою.

У другому розділі описано принципи роботи технології Blockchain та аналіз способів для розробки.

У третьому розділі та четвертому розділах наведено структуру та опис роботи програмного забезпечення, а також проведено тестування та проаналізовано результати виконаного дослідження.

У висновках представлені результати проведеної роботи.

Робота представлена на 86 аркушах, містить посилання на список використаних літературних джерел.

Ключові слова: Web 3.0, Blockchain, Ethereum, Solidity, Bitcoin, криптовалюта, транзакції.

ABSTRACT

Actuality of theme. In today's world, the Internet is an integral part of every person's life. Today, it is impossible to imagine any sphere of business, education, provision of services, or advertising without the use of web resources such as web applications or web pages, social networks, messengers, etc. Web 3.0 represents a new iteration or phase of the Internet's evolution. The development of Web 3.0 began in 2010 and continues now. Web 3.0 is built on the core concepts of decentralization, openness, and greater user-friendliness. Decentralization can solve the problems of web service opacity, network censorship, and personal data privacy, which are pressing issues in the current Web 2.0. Since Blockchain is a fundamental concept for the new generation of web resources, and the transfer of data in the blockchain network is instantaneous, its main field of use is finance. Cryptocurrency is already a full-fledged means of payment, so the creation of a convenient decentralized web-oriented tool for carrying out operations on cryptocurrency using Blockchain technology will allow solving the problem of confidentiality, secure authorization and privacy when conducting financial transactions.

The object of the study is the analysis of Blockchain technology and its application in web-based tools for controlling and executing cryptocurrency transactions.

The subject of the research is web-based software for monitoring and executing cryptocurrency transactions using Blockchain technology.

The purpose of the work: to develop a decentralized web-based software for controlling and executing cryptocurrency transactions, which would be fast in operation compared to centralized means of conducting financial transactions, transparent due to the use of a decentralized structure, reliable and confidential for users due to authorization using a crypto wallet.

The scientific novelty is as follows: a decentralized web-oriented tool for controlling and executing cryptocurrency transactions is proposed. The advantage of such a tool is that, thanks to Blockchain technology, it is fast, transparent and reliable. In addition, a decentralized application never stops working due to the absence of a central server that could stop. The data in it is distributed among nodes, while all nodes operate independently, and if one of them stops, the others will continue to work in the network.

The practical value of the results obtained in the work is that the developed web application allows you to authenticate using your crypto wallet, which improves privacy, safely and quickly performs transactions by transferring cryptocurrencies from one wallet to another, conveniently monitoring the execution of transactions.

Approbation of work. The main provisions were presented and discussed at the XIV Scientific Conference of Master's and Postgraduate Students "Applied Mathematics and Computing" PMK-2022 (Kyiv, November 2022).

Structure and scope of work. The master's thesis consists of an introduction, four chapters and conclusions.

The introduction provides a general description of the work, assesses the current state of the problem, substantiates the relevance of the research direction, formulates the purpose and tasks of the research, shows the scientific novelty of the obtained results and the practical value of the work, provides information on the approbation of the results and their implementation.

The first section examines the existing software tools for monitoring and executing cryptocurrency transactions.

The second chapter describes the working principles of Blockchain technology and the analysis of methods for development.

The third chapter and the fourth chapter give the structure and description of the software, as well as testing and analyzing the results of the research.

The results of the work are presented in the conclusions.

The work is presented on 86 sheets, containing links to the list of used literary sources.

Keywords: Web 3.0, Blockchain, Ethereum, Solidity, Bitcoin, cryptocurrency, transactions.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ, УМОВНИХ ПОЗНАЧЕНЬ, ТЕРМІНІВ

ІКТ - інформаційно-комунікаційні технології.

WWW (англ. World Wide Web) - всесвітня павутина Інтернет.

ІТ - інформаційні технології.

HTML (англ. Hyper Text Markup Language) - стандартна мова розмітки для вебсторінок.

RSS (англ. Really Simple Syndication) - загальна назва для усіх вебстрічок, які використовують формат XML.

IoT (англ. Internet of Things) - інтернет речей.

RDF (англ. Resource Description Framework) - концепція семантичної павутини, що включає середовище опису ресурсів.

IPFS (англ. InterPlanetary File System) - однорангова розподілена файлова система, що з’єднує всі обчислювальні пристрої в єдину систему файлів.

ШІ - штучний інтелект.

NFT (англ. Non-fungible token) - особливий тип криптовалюти, що є взаємозамінним. Такі активи існують лише у власних криптосистемах.

BTC - криптовалюта Bitcoin.

ETH - криптовалюта Ether.

LTC - криптовалюта Litecoin.

DOGE - криптовалюта Dogecoin.

BNB - криптовалюта Binance Coin.

ICO (англ. Initial Coin Offering) - первинне розміщення монет або токенів, форма залучення інвестицій в проєкти та стартапи.

KYC (англ. Know your customer) - термін у банківській сфері, що означає ідентифікацію та встановлення клієнта перед проведенням фінансової операції.

P2P (англ. Peer-to-peer) - архітектура системи, що складається із мережі рівноправних вузлів

EVM (англ. Ethereum Virtual Machine) - віртуальна машина Ethereum, обчислювальне середовище, розподілений комп’ютер, що відповідає за виконання алгоритмів смарт-контрактів в мережі Ethereum.

BAAS (англ. Banking as a service) - банк, як послуга. Здача банками в оренду своєї інфраструктури.

UTXO (англ. Unspent Transaction Output) - невитрачений вихід транзакції, представляє деяку суму цифрової валюти, яка була авторизована одним обліковим записом для використання іншим.

API (англ. Application programming interface) - прикладний програмний інтерфейс.

ABI (англ. Application binary interface) - прикладний бінарний інтерфейс, що використовується для двома бінарними програмними модулями.

JSON (англ. JavaScript Object Notation) - відкритий стандартний формат файлів і формат обміну даними, який використовує зрозумілий для людини текст для зберігання та передачі об’єктів даних, що складаються з пар атрибут–значення та масивів, створений на основі мови програмування JavaScript.

ООП - об’єктно-орієнтоване програмування.

CLI (англ. Command-line interface) - інтерфейс командного рядка.

DApp (англ. Decentralized Application) - децентралізований додаток. Нове покоління вебдодатків, що використовує хоча б одну Web 3.0 технологію.

GUI (англ. Graphical user interface) - графічний інтерфейс користувача.

JSON-RPC (англ. JavaScript Object Notation Remote Procedure Call) - протокол віддаленого виклику процедур, що використовує формат JSON для кодування повідомлень.

UI (англ. User interface) - інтерфейс користувача.

DOM (англ. Document Object Model) - об’єктна модель документа, яка розглядає документ XML або HTML як структуру дерева, де кожен вузол є об’єктом, що представляє частину документа.

JSX (англ. JavaScript syntax extension) - розширення синтаксису мови програмування JavaScript. Використовується бібліотекою React для створення графічних інтерфейсів користувача.

CSS (англ. Cascading Style Sheets) - каскадні таблиці стилів, що використовуються для опису зовнішнього виду HTML-сторінки.

XML (англ. Extensible Markup Language) - розширювана мова розмітки документів в мережі Інтернет.

IDE (англ. integrated development environment) - інтегроване середовище розробки. Програмне забезпечення, яке надає програмістам комплексні можливості для розробки програмного забезпечення.

OS (англ. Operating system) - операційна система.

ВСТУП

В сучасному світі інтернет є невід’ємною частиною життя кожної людини. Сьогодні неможливо уявити будь-яку сферу бізнесу, навчання, надання послуг, або реклами без використання вебресурсів таких, як вебдодатки або вебсторінки, соціальні мережі, месенджери тощо. Але інтернет не був таким завжди, і значно змінився з часів створення першої вебсторінки. З того часу минуло тридцять два роки, 20 грудня 1990 року Тім Бернерс-Лі, британський вчений з Європейської організації з ядерних досліджень (CERN), запустив перший у світі вебсайт. Він був присвячений опису нової технології глобального обміну інформацією - “World Wide Web” або “Всесвітня павутина”.

В наш час найпрогресивнішим підходом для створення вебдодатків є Web 3.0, який представляє собою нову ітерацію еволюції Інтернету. Він побудований на основних концепціях децентралізації, відкритості та більшої зручності для користувачів. Фундаментальним елементом в Інтернеті нового покоління є технологія Blockchain. Завдяки їй та криптовалютам, що стрімко розвиваються і вже стали повноцінним платіжним засобом, актуальним завданням є створення вебдодатків нового покоління для контролю операцій над криптовалютою.

Метою роботи є створення децентралізованого вебдодатку, який працює на Blockchain мережі Ethereum та дозволяє авторизуватися за допомогою свого криптогаманця MetaMask, проводити транзакції над криптовалютою Ether з одного гаманця на інший, та виконувати моніторинг проведених транзакцій в реальному часі. При цьому основною його перевагою є простота, прозорість, конфіденційність особистих даних, надійність та детермінованість виконання операцій. Головною технологією такого додатку є Blockchain.

1. АНАЛІЗ ІСНУЮЧИХ РІШЕНЬ ТА ОБҐРУНТУВАННЯ ТЕМИ МАГІСТЕРСЬКОЇ ДИСЕРТАЦІЇ

1.1. Аналіз версій World Wide Web

Світові зміни, зокрема, інформаційно-комунікаційні технології (ІКТ) сильно вплинули на функціонування сучасного суспільства. Розвиток ІКТ по-різному змінив погляди користувачів на сучасне суспільство. ІКТ є інструментом, що дозволяє надавати ефективні комунікаційні та інформаційні послуги людям. Всесвітня павутина (WWW) значною мірою змінила спосіб доступу людей до інформації та спілкування з іншими людьми. Соціальні медіа - це комунікація через Інтернет, яка сприяє налагодженню соціальних зв'язків. Останні досягнення в галузі інформаційних технологій (ІТ) не тільки покращили можливості доступу, зберігання та обробки інформації в усьому світі, але й призвели до значних змін у концепції, розширенні послуг, функціонуванні та управлінні мережею. Всесвітня мережа Інтернет розвивалася в геометричній прогресії протягом останніх кількох років.

У 1989 році Тім Бернерс-Лі запропонував проект міжнародної системи протоколів. Це розподілений гіпермедійний сервер, який дозволяє користувачам мережі впорядковувати електронні документи, які є композитами або вказівниками на багато різних файлів різних типів, розкиданих по всьому світу. Він назвав її Всесвітньою павутиною (WWW). Крім цього він розробив багато комунікаційних протоколів, які складають основу Всесвітньої павутини [1].

Web 1.0 - це перше покоління Інтернету, активний розвиток якого тривав з 1990 по 2000 рік [1]. У Web 1.0 обмежена кількість авторів формувала вебсторінки для величезної кількості читачів. Web 1.0 був в основному лише для читання. Його ідея полягала у створенні спільного інформаційного простору для користувачів Інтернету та обміну інформацією між ними. Така мережа надавала доступ до контактної інформації, електронної пошти, адрес, номерів телефону, факсу, а також до брошур рекламних оголошень в газетах і журналах. вебсайти являли

собою статичні сторінки на мові розмітки гіпертексту сторінки HTML, які оновлювалися не часто.

Web 2.0 - це друге покоління Інтернету. Термін Web 2.0 був офіційно виокремлений у жовтні 2004 року і був запропонований Дейлом Догерті (O'Reilly Media) та Крейгом Клайном (Media Live). Ідея нової мережі виникла під час мозкового штурму на конференції між O'Reilly та Media Live International [2].

Друге покоління розглядається як мережа для читання та запису, яка дозволяє керувати та збирати велику кількість людей по всьому світу зі схожими інтересами в соціальних відносинах. Термін Web 2.0 зазвичай пов'язують з вебдодатками, які полегшують інтерактивний обмін інформацією, дизайн, адаптований до потреб користувача, та співпрацю у Всесвітній павутині. Основна концепція соціальних мереж - це канали для обміну інформацією та спілкування в режимі реального часу. Соціальні мережі заохочують соціальну взаємодію через профільні облікові записи користувачів та зосереджуються на нових шляхах комунікації та взаємовідносин з користувачами. Web 2.0 також визначають як мережу знань, людино-орієнтовану мережу, мережу для читання-запису та мережу участі. Це наступна великий прорив у Всесвітній павутині, яка пропонує використання останніх технологій і концепцій для того, щоб створити користувачеві знайомий, більш інтерактивний, корисний і взаємопов'язаний простір. Web 2.0 - це не тільки нова версія Web 1.0, вона включає в себе гнучкий вебдизайн, творче повторне використання, оновлення, спільне створення та модифікацію контенту. Вона включає в себе технології та послуги і складається з блогів, аудіо, чатів, закладок, календарів, електронної комерції, електронної пошти, ігор, електронного навчання, спілкування, форумів, картографії, мультимедіа, порталів, дуже простої синдикації (RSS), тегів і т.д. Це поточна версія Інтернету, з якою знайомі сучасні користувачі. Web 2.0 є стандартом за замовчуванням. Активний розвиток цієї версії тривав з 2000 по 2010 рік. Якщо перша версія являє собою статичну мережу з лінійною інформацію, то друга версія - це динамічна мережа, що складається із нелінійної інформації, в якій не має обмежень, і крім цього її можна редагувати. Основними проблемами даної

мережі є компромісна безпека та конфіденційність, цензурування даних та централізація контенту [1].

Web 3.0 - словосполучення, запропоноване Джоном Маркоффом з New York Times у 2006 році - третє покоління Всесвітньої павутини, яке зазвичай включає в себе семантичне маркування контенту. Web 3.0 також відомий як Семантична павутина, основою якої є інтеграція даних. За допомогою метаданих, дані "тільки для відображення" перетворюються на змістовну інформацію, яка може бути знайдена, оцінена і доставлена програмними агентами. Тім Бернерс Лі, винахідник Всесвітньої павутини, був тим, хто прийшов до думки про семантичну павутину, тобто про Web 3.0. На внутрішньому рівні Web 3.0 надає можливості перегляду та пошуку контенту (семантики) та контексту наступного покоління, що складається із великих обсягів даних. Мережа третього покоління підтримує всесвітню базу даних і веборієнтовану архітектуру, яка на більш ранній стадії була описана як павутина документів, та складається в основному зі статичних документів HTML, але динамічно відображувані сторінки і альтернативні формати повинні слідувати тим же концептуальним стандартам верстки, коли це можливо [3]. Розвиток Web 3.0 розпочався в 2010 році і триває зараз [1]. Web 3.0 побудовано на основних концепціях децентралізації, відкритості та більшої зручності для користувачів.

Якщо уважно подивитись на всі три версії, то можна помітити, що вони мають лише кілька фундаментальних спільних рис [4]:

- Всі вони мають справу з відносинами між кінцевими користувачами та інформацією
- Всі вони надають користувачам функцію "читання"
- Всі вони покладаються на Інтернет для прискорення виконання своїх завдань

Нижче наведено розбивку основних особливостей кожної версії [5]:

1. Web 1.0

- Відсутність зв'язку між користувачем і сервером
- Статичні вебсайти

- Лише перегляд вмісту
- Гіперпосилання та закладки на сторінках
- вебсторінки тільки для читання

2. Web 2.0

- Покращена взаємодія з користувачем порівняно із Web 1.0
- Впровадження вебдодатків
- Онлайн-документи, потокове відео тощо
- Інформація додатків зберігаються на серверах
- Інтерактивна реклама
- Операції з хмарними обчисленнями
- Централізовані дані
- Читання і запис

3. Web 3.0

- Інтелектуальні вебфункції та додатки
- Децентралізовані процеси
- Поєднання вебтехнологій та представлення знань
- Поведінкова реклама
- Машинне навчання
- Відео в реальному часі
- Інтернет речей (IoT)
- Семантичний пошук
- Читання, запис та керування Інтернетом

Порівнюючи всі 3 версії, можна сказати, що Web 1.0 - мережа лише для читання, Web 2.0 - для читання та запису, а Web 3.0 - для читання, запису та виконання (Рисунок 1) .

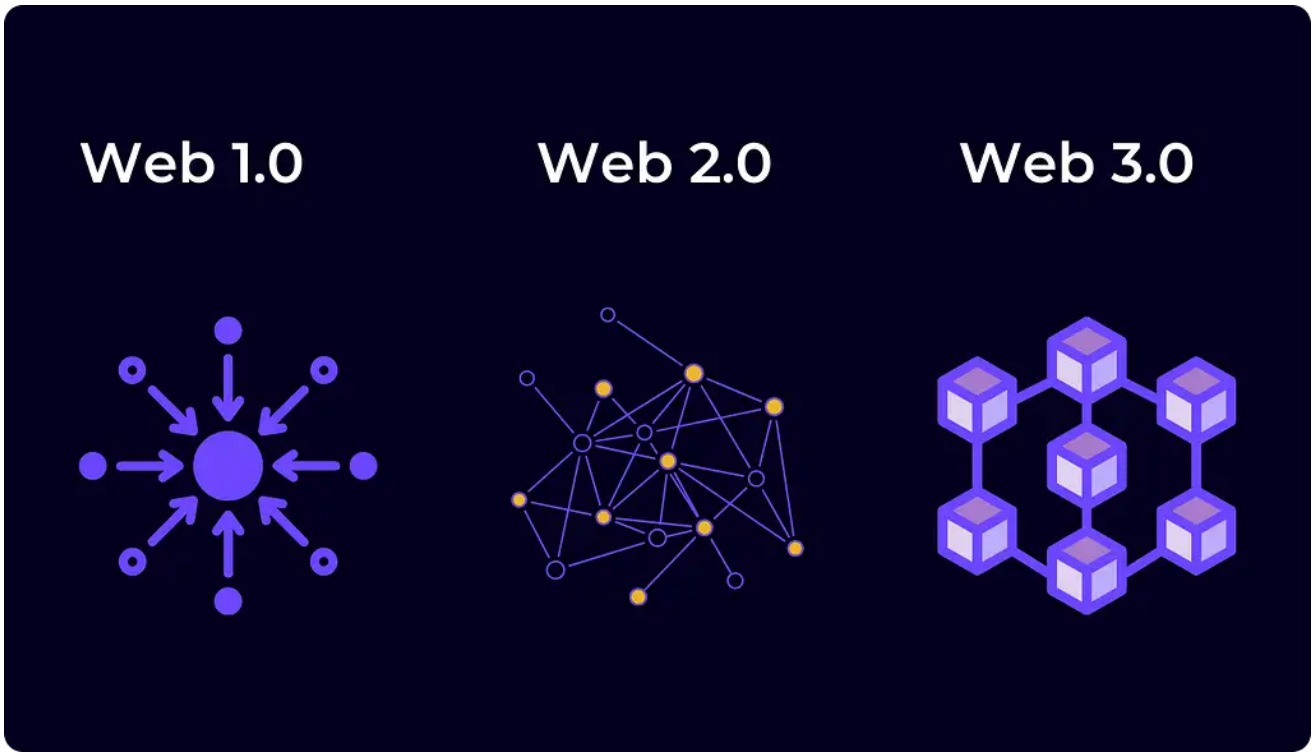

Рисунок 1 - Схематичне зображення трьох версій WWW

Для третьої версії всі дані є децентралізованими, тобто копії даних зберігаються на різних серверах, і ніхто не має контролю над ними. Децентралізовані додатки ніколи не припиняють роботу завдяки відсутності центрального сервера, який міг би зупинитися. Дані в них розподілені між вузлами, при цьому всі вузли діють незалежно, і якщо один з них зупиниться, інші продовжать роботу в мережі. Web 3.0 - являє собою інфраструктуру із декількох основних технологій: Blockchain, Semantic Web, Machine Learning та Internet of things. Кожна із цих технологій є невід'ємною частиною мережі зі своєю роллю в екосистемі, яка дозволяє створювати системи, додатки, сервери нового покоління [1].

Blockchain - це технологія розподіленої бази даних. Блокчейн-інформація записана в ланцюг блоків, які підключені в строго визначеному порядку. Системи, засновані на Blockchain, здатні працювати без централізованого управління та довіри між учасниками. Рішення в такій системі приймаються не центральними вузлами, а шляхом голосування всіх учасників системи. Для створення та підтримання блокчейну не потрібні потужні сервери, достатньо створити мережу

із звичайних комп’ютерів. Поведінка учасників регулюється консенсусним протоколом — набором правил для створення та запису блоків у реєстрі системи. Розподілений реєстр є основою децентралізованої вебінфраструктури, де вебпрограми можуть обмінюватися інформацією напряму та з усіма учасниками на рівних умовах. Завдяки децентралізації можна вирішити проблеми непрозорості вебсервісів, цензури у мережі та конфіденційності особистих даних, що є надважливою проблемою в поточному Web 2.0. Передача даних в блокчейн-мережі проходить миттєво, тому головна сфера його використання це фінанси. Криптовалюта, яка доступна вже зараз, керується децентралізованою платіжною системою повністю автоматизовано, тобто без внутрішнього або зовнішнього адміністрування. Підробити або скасувати запис в такій системі неможливо, або дуже витратно. Всі транзакції виконуються анонімно, відомі тільки номери рахунків, а не особисті дані [1].

Semantic Web або семантична павутина - наступна концепція, що вирішує проблему перетворення всієї інформації в мережі в зрозумілу для машин. Для реалізації семантичної павутини, всій інформації в мережі необхідно присвоїти метадані, тобто інформацію про інформацію. Ключовий елемент в реалізації семантичної павутини - RDF (Resource Description Framework). Це модель опису інформації через спеціальні зрозумілі машині твердження - триплети, що складається з трьох частин: «суб'єкт», «предикат» та «об'єкт». За допомогою цього вирішується проблема опису будь-якого об’єкта в мережі: вебдодатку, зображення, аудіозапису, людини [1].

Machine Learning або Машинне навчання — це система методів, які використовують комп’ютерні алгоритми для вирішення проблем без прямих інструкцій. Алгоритм вчиться виконувати певне завдання. Він аналізує набір даних і самостійно виявляє шаблони, які потім використовує для виконання завдань. Така концепція стає фундаментальною для Web 3.0, так як це вирішує проблему аналізу великої кількості даних, що присутня в мережі нового покоління [1].

Концепція Internet of things або Інтернет речей описує мережу багатьох пристроїв, які мають датчики та підключення до Інтернету, що взаємодіють один з одним і ними можна керувати віддалено. Інтернет речей поєднує світ вебдодатків зі світом запрограмованих розумних фізичних об'єктів: автомобілів, будинків, телевізорів. Для Web 3.0 такі пристрої є не тільки ще одним засобом доступу до мережі, а й постійними джерелами даних про фізичний світ. За допомогою Інтернету речей в мережі накопичуються детальні дані, що можна використовувати в потрібній для користувачів сфері, наприклад для пошуку найкращого маршруту, вирахування статистики, урбаністичних досліджень [1].

Додатки нового покоління становлять собою програмне забезпечення, яке не обов'язково використовує всі концепції Web 3.0, але мінімум одну із технологій. Основним прикладом є додатки для виконання операцій з криптовалютами, для аналізу даних за допомогою штучного інтелекту, голосові асистенти для виконання розумного пошуку або управління системою розумного дому, побудови 3D-моделей ландшафту тощо [1]. Розглянемо приклади таких додатків.

Sapien є унікальним прикладом Web 3.0. Це децентралізований соціальний новинний сайт, що працює на блокчейні Ethereum. Він також неймовірно легко налаштовується. Коли справа доходить до соціальних новин, це хороша альтернатива Google або Facebook. Це дозволить користувачам створювати власні додатки без необхідності оплати або реклами. Платформа Sapien надає користувачам інструменти, необхідні для швидкого створення та запуску додатків. Sapien - це вебплатформа на основі блокчейну, яка надає додатки, послуги та технології. Завдяки набору автономних інструментів, вона дозволяє користувачам взаємодіяти, здійснювати транзакції, обмінюватися інформацією та створювати спільноти. Sapien web 3 була створена для того, щоб надати кожному можливість організувати своє життя в Інтернеті [6].

Everledger - один з перших прикладів Web 3.0 додатків, розподілений цифровий глобальний реєстр. По суті, він пропонує унікальний запис для кожного користувача, і користувачі можуть зберігати дані в хмарі з гнучким доступом до них. Основні програми Everledger зосереджені на забезпеченні надійного захисту

від шахрайства. Зокрема, додаток Everledger допомагає відстежувати діаманти, золото, вино та інші цінні активи. Він допомагає відстежувати використання товару на всьому шляху його руху в глобальному ланцюгу поставок. Клієнти можуть відсканувати наклейку Everledger, щоб переконатися, що вони купують справжню продукцію. Додаток Everledger встановлює чітку віху у сфері веббезпеки, забезпечуючи захист від шахрайства без використання персональних даних або складних паролів [6].

Storj - децентралізоване рішення для зберігання даних. Воно допомагає користувачам безпечно зберігати дані, забезпечуючи при цьому відмовостійкість і надмірність. Storj використовує технологію блокчейн, щоб революціонізувати переваги хмарного зберігання даних, хоча і в розподіленій мережі. Найважливішою перевагою Storj є гнучкість для необмеженого завантаження і вивантаження, незалежно від часу. Крім того, Storj відповідає умовам, що висуваються до додатків Web 3.0, таким як дизайн з відкритим вихідним кодом і безкоштовна модель використання. Крім того, Storj також забезпечує кращу простоту використання, оскільки ви можете почати навігацію в додатку в кілька кліків. Ще однією важливою особливістю є монета Storj, яка забезпечує роботу децентралізованої платформи зберігання даних.

Brave Browser - це новий браузер, який має на меті запропонувати блокування реклами та інші функції, що підвищують конфіденційність на мобільних пристроях. Фактично, Brave став однією з найбільш бажаних альтернатив іншим популярним браузерам, таким як Mozilla Firefox та Google Chrome [6]. Крім того, Brave також успішно став одним з найбезпечніших браузерів з перспективними функціями безпеки. Новий браузер з відкритим вихідним кодом використовує технологію блокчейн для захисту конфіденційності користувачів. Браузер Brave також використовує блокчейн для переривання небажаної реклами, а також трекерів. Найважливіше те, що браузер Brave також забезпечує гнучкість для монетизації даних користувачів в обмін на токени Brave.

Beaker Browser - в даний час розробляється в Blue Link Labs, браузер Beaker є безкоштовним браузером з відкритим вихідним кодом для Web 3.0.

Одноранговий веббраузер може допомогти користувачам у публікації вебдодатків, а також вебсайтів з самого браузера. Браузер Beaker, безумовно, є одним з найкращих web3 додатків з точки зору зручності використання, оскільки він виключає вимоги до налаштування та підтримки окремих серверів. Крім того, користувачі можуть не турбуватися про розміщення свого контенту на сторонніх серверах, тим самим підвищуючи рівень безпеки.

Sola - це децентралізована платформа соціальних медіа, заснована на блокчейні Ethereum, сховищі IPFS та розподілених вузлах. Sola використовує блокчейн та штучний інтелект для створення комбінації соціальних мереж. Платформа соціальних медіа використовує алгоритми ШІ для розповсюдження релевантної інформації користувачам відповідно до їхніх уподобань. Sola фокусується на розробці платформи соціальних медіа, де кожен має рівну свободу вираження поглядів без надання монополії рекламодавцям, впливовим особам та великим брендам [7]. Передова мережа нейронного навчання Sola, Neutral AI, допомагає оптимізувати процес обміну контентом. Крім того, користувачі також можуть досліджувати можливості монетизації для обміну та створення контенту.

Decentraland - Найбільш розрекламований проект в метапросторі або майбутнє Web 3.0, Decentraland - це віртуальний онлайн світ з безліччю цікавих можливосте [7]. Decentraland є очевидним претендентом на місце серед кращих Web 3.0 додатків з чимось незвичайним. Віртуальний світ дозволяє користувачам зустрічатися, взаємодіяти, спілкуватися та грати в ігри з іншими користувачами. Крім того, метасвіт Decentraland в основному обертається навколо налаштування віртуальних земельних ділянок та створення нових вражень. Користувачі досліджують перспективи іпотеки віртуальної власності та здачі в оренду віртуальної власності для проведення заходів у метапросторі. Фактично, Decentraland привернув увагу багатьох великих брендів, таких як Samsung, Adidas і PwC.

Steemit - це більше схоже на Reddit і працює так само, як децентралізований вебсайт соціальних блогів. Найбільшою родзинкою Steemit, як одного з прикладів додатків Web 3.0, є монетизація користувацького контенту. Користувачі Steemit

можуть отримувати платежі за свій контент на основі голосування спільноти. Маючи понад 1,7 мільйона зареєстрованих акаунтів на платформі, Steemit був розроблений на власному блокчейні під назвою Steem і відповідає принципам Web 3.0 у своїй основній діяльності.

Ethlance - застосування Web 3.0 у сфері пошуку роботи та професійного нетворкінгу призвело до появи таких платформ, як Ethlance. Онлайн-платформа допомагає користувачам подавати заявки на роботу. Цікаво, що Ethlance використовує технологію блокчейн для прискорення процесу підбору кандидатів на роботу з відповідними роботодавцями [7]. Крім того, Ethlance є одним з інноваційних прикладів Web 3.0 додатків, які відкривають нові можливості для фрілансерів. Ethlance може зруйнувати традиційні бар'єри між фрілансерами та компаніями як децентралізований ринок талантів. Роботодавці можуть знизити свої витрати на пошук талантів, в той час як фрілансери можуть досягти кращого контролю над своєю роботою.

Secretum - це новий децентралізований додаток для обміну повідомленнями. Secretum є децентралізованою альтернативою додаткам для обміну повідомленнями, таким як WhatsApp. Користувачі можуть зв'язуватися один з одним без номера телефону або адреси електронної пошти, забезпечуючи тим самим кращу безпеку і конфіденційність. Вбудована функція торгівлі в Secretum також забезпечує гнучкість для ефективної торгівлі криптовалютами, NFT та іншими токенами блокчейну.

DTube - децентралізована альтернатива гігантському відео-гіганту YouTube за допомогою DTube. Децентралізована платформа для потокового відео не контролює відео, які з'являються в стрічках користувачів [7]. Користувачі мають повний контроль над тим, що вони бачать і чим діляться на DTube. Крім того, користувачі мають право приймати рішення щодо відбору відео для монетизації. Будучи одним з ключових додатків Web 3.0 для користувачів у майбутньому, DTube також здійснює децентралізоване управління. Користувачі повністю контролюють цензуру та інші правила щодо видимості та використання контенту на платформі потокового відео.

1.2. Опис існуючих рішень

Для того, щоб зрозуміти як працюють додатки Web 3.0, і зокрема, що пов’язані із криптовалютами необхідно проаналізувати найбільший і найвідоміший - це Binance Exchange.

Binance Exchange - провідна криптовалютна біржа, заснована в 2017 році в Гонконзі. Основна увага приділяється торгівлі альткоїнами. Altcoin - це термін, який складається з комбінації слів “Alt” - альтернатива, та “Coin” - криптовалюта. Тобто це така категорія криптовалюти, що є альтернативною криптовалюті Bitcoin. Binance пропонує криптовалютну торгівлю більш ніж 600 криптовалютами та віртуальними токенами, включаючи Bitcoin (BTC), Ether (ETH), Litecoin (LTC), Dogecoin (DOGE) та власний токен Binance Coin (BNB) [8].

Ключові особливості Binance:

- Це онлайн-біржа, де користувачі можуть торгувати та обмінюватися криптовалютами.
- Вона підтримує найбільш поширені криптовалюти.
- Binance надає криптогаманець для трейдерів для зберігання своїх електронних коштів.
- Біржа також має допоміжні послуги для користувачів, які дозволяють заробляти відсотки або здійснювати транзакції за допомогою криптовалют.
- Вона також пропонує програми для майнерів та для допомоги трейдерам у прийнятті інвестиційних рішень.
- Binance має власний токен на основі блокчейну - Binance Coin (BNB).

Біржа Binance, відома насамперед завдяки торгівлі між двома криптовалютними парами та має одну з найнижчих комісій за транзакції для криптовалютних бірж. Вона має високу ліквідність і пропонує знижки користувачам, які платять у власних криптовалютних токенах BNB. Завдяки високим стандартам безпеки та захисту, а також багаторівневій та

багатокластерній архітектурі, Binance забезпечує високу пропускну здатність з можливістю обробляти близько 1,4 мільйона замовлень в секунду. Підтримує торгівлю більш ніж 600 монетами, включаючи Bitcoin, Ethereum, Litecoin та власні токени BNB. Як і інші криптовалютні біржі, Binance пропонує послуги з торгівлі, лістингу, збору коштів, а також делістингу або виведення криптовалют. Криптовалютні ентузіасти, які бажають запустити свої токени, можуть використовувати Binance для збору коштів через первинне розміщення монет (ICO). Binance використовується великою кількістю трейдерів та учасників для обміну та інвестування в різні криптовалюти. Для того, щоб розпочати торгівлю, користувачі повинні виконати необхідні KYC вимоги. Після успішного створення торгового рахунку користувачі можуть додати криптовалютні кошти на адресу свого публічного гаманця, наданого Binance, щоб почати торгівлю. Біржа Binance підтримує три основні типи торгових ордерів: лімітні, ринкові та стоп-лімітні. Лімітні ордери виконуються тільки за встановленою трейдером лімітною ціною. Ринкові ордери виконуються негайно за найкращою доступною ринковою ціною. Стоп-лімітні ордери стають дійсними ордерами тільки тоді, коли ціна досягає заданого рівня [8].

Переваги Binance:

- Низькі комісії: Binance має складну структуру комісій, яка залежить від того, яким активом ви торгуєте, як ви платите і скільки ви торгуєте. Але торгівля біткойнами, як правило, безкоштовна.
- Виконання угод: Binance відомий своїм високошвидкісним виконанням торгівельних операцій. Перед тим, як засновник компанії Чанпенг Чжао запустив Binance в Китаї в 2017 році, він розробив програмну систему для узгодження замовлень для високошвидкісних трейдерів.
- Вибір криптовалют: Хоча вона не пропонує стільки криптовалют і криптовалютних торгових пар, як її материнська компанія, вибір з більш ніж 125 криптовалют на Binance.US є конкурентоспроможним з багатьма біржами вищого рівня і все ще випереджає багато інших американських бірж.

Недоліки Binance:

- Доступність: Binance доступний не у всіх регіонах світу.
- Регуляторні питання: Глобальна материнська компанія Binance зіткнулася з регуляторними проблемами і суперечками в деяких країнах, перемістивши операції з Китаю в Японію; на технічній конференції в травні 2020 року засновник Чанпенг Чжао заявив, що у материнської компанії немає штаб-квартири, тому що "у біткоїна немає офісу".

Binance найкраще підходить для:

- P2P торгівлі.
- Торгівля між криптовалютами.

Розглянемо базовий функціонал вебдодатку Binance для контролю та виконання транзакцій з криптовалютою на прикладі переведення криптовалюти Bitcoin. Необхідно відкрити вкладку Fiat and Spot в лівому меню і далі натиснути кнопку Transfer у верхній частині сторінки (Рисунок 2).

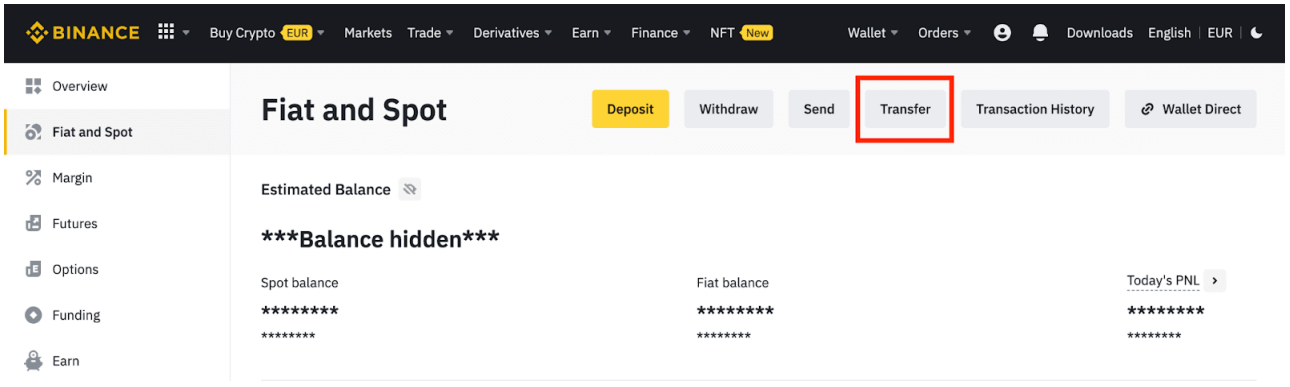

Рисунок 2 - Відкриття функціонал по переведенню криптовалюти

Далі необхідно обрати основні параметри переводу: обрати криптогаманець з якого та на який буде виконано переведення коштів, криптовалюту транзакції та перевірити кількість криптовалюти для переведення (Рисунок 3). Після виконання транзакції можна перейти до вкладки Transaction history та переглянути історію виконаних переведень (Рисунок 4).

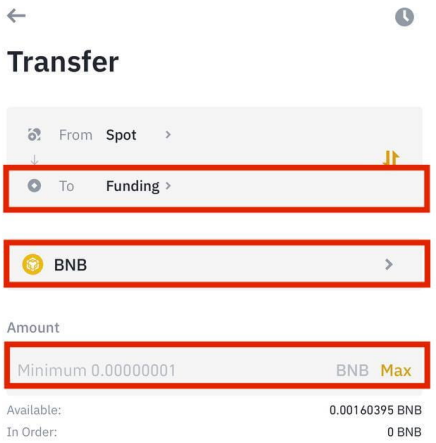

Рисунок 3 - Вибір необхідних параметрів для переведення

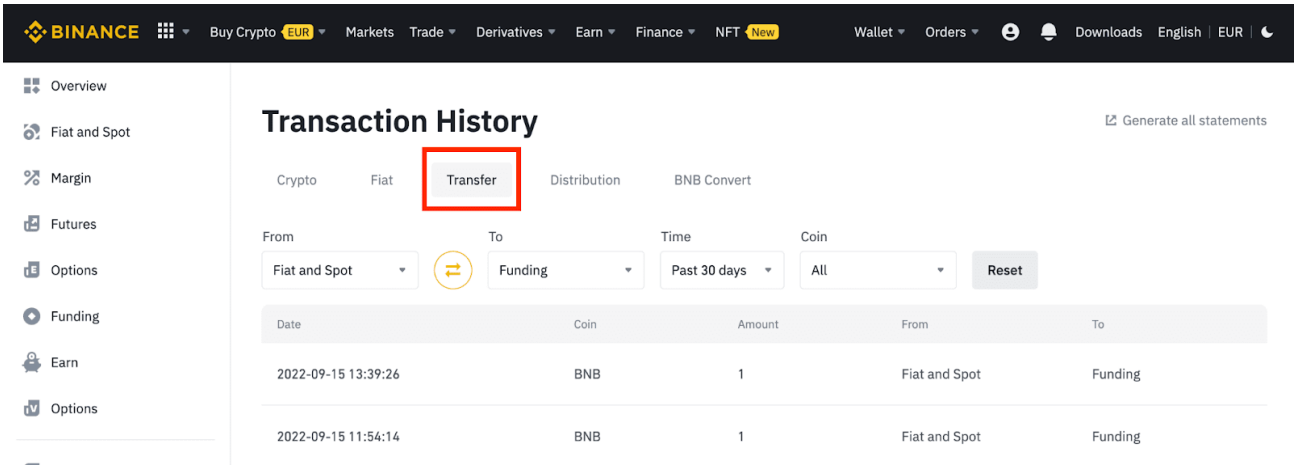

Рис 4. Перегляд історії транзакцій

1.3. Постановка задачі магістерської дисертації

Отже задачею є магістерської дисертації є аналіз технології Blockchain, її застосування в додатках нового покоління Інтернету Web 3.0 та розробити децентралізований веборієнтований засіб для контролю та виконання транзакцій з криптовалютою. Перевагою такого засобу є те, що завдяки технології Blockchain, він буде являтись швидким, прозорим та надійним. При цьому забезпечуючи конфіденційність особистих даних, надійність та детермінованість виконання операцій. Крім того децентралізований додаток ніколи не припиняє роботу

завдяки відсутності центрального сервера, який міг би зупинитися. Дані в ньому розподілені між вузлами, при цьому всі вузли діють незалежно, і якщо один з них зупиниться, інші продовжать роботу в мережі.

Практична цінність роботи полягає в тому, що розроблений вебдодаток буде дозволяти авторизуватися за допомогою свого криптогаманця, що є покращує конфіденційність, безпечно та швидко виконувати транзакцій по переводу криптовалют із одного гаманця на інший, зручно контролювати виконання транзакцій.

ВИСНОВКИ ДО ПЕРШОГО РОЗДІЛУ МД

В першому розділі дисертації було проведено аналіз версій World Wide Web для визначення найкращих методів та підходів для розробки веборієнтованого програмного забезпечення для контролю та виконання транзакцій над криптовалютою. Крім, того було проведено аналіз існуючих рішень та методів для виконання операцій над криптовалютою. Також була сформульована задача магістерської дисертації та визначена практична цінність роботи.

2. ОПИС ТЕХНОЛОГІЇ BLOCKCHAIN ТА АНАЛІЗ СПОСОБІВ РОЗРОБКИ

2.2. Принципи роботи технології Blockchain

Блокчейн - це децентралізована структура даних, внутрішня узгодженість якої підтримується завдяки консенсусу, досягнутому всіма користувачами щодо поточного стану мережі. Це технологія, яка вирішила проблему візантійських полководців (обмін повідомленнями між сторонами, яким не довіряють) і відкрила новий горизонт можливостей для безоплатних транзакцій та обміну інформацією. Якщо Інтернет демократизував піринговий обмін інформацією, то блокчейн демократизував піринговий обмін цінностями. Це ще одна революційна технологія, яка може змінити Інтернет так само, як це зробило програмне забезпечення з відкритим вихідним кодом. Так само, як і їм, блокчейну знадобиться деякий час, щоб стати доступним за ціною, щоб кожен міг використовувати його в сучасних розробках. Цей процес може зайняти роки, щоб досягти реального потенціалу, який закладений в ньому. Оскільки блокчейн є розподіленою системою обліку P2P, будь-хто може бачити записи інших користувачів, але ніхто не може їх змінити. Це підвищує рівень прозорості всієї системи [9].

Однак навряд чи вона замінить традиційну централізовану мережу баз даних, яку використовують уряди; скоріше, вона може структурувати нову систему в рамках глобальних підприємств. Оскільки нові технології впроваджуються щороку, ми стоїмо на порозі нової ери.

Оновлювати блокчейн можна лише за допомогою алгоритму консенсусу. Таким чином, коли в систему завантажується новий набір інформації, ніхто не може його змінити. Блокчейн буде містити точну і достовірну інформацію про реєстр.

Мережа може з'єднуватися з користувачами та здійснювати прямі транзакції між ними. Відмітки часу та реєстр підтримуються системою автоматично. Це означає, що вам не знадобляться жодні дані від третьої сторони в системі.

При цьому всі користувачі технології блокчейн є адміністраторами. Також на технології блокчейн можна запускати смарт-контракти. Оскільки мережа використовує алгоритми консенсусу, можна швидко створити проект з певними умовами. Чудовим прикладом може бути ринок нерухомості. Якщо при укладанні угод не виконуються певні умови, то угода не відбувається. Зазвичай, в блокчейні користувачі використовують власну валюту, визначену мережею, для оплати в якості токенів.

Блокчейн забезпечує безпеку в мережі, використовуючи поняття "ключ". Якщо ви використовуєте набір зашифрованих ключів, ви отримуєте унікальну ідентифікацію, яку ніхто не може зламати. Ви отримуєте приватний і публічний ключ і завдяки цьому використовуючи цю комбінацію, ви отримаєте унікальну ідентифікацію. Інші користувачі будуть використовувати ваш відкритий ключ, щоб знайти вас в мережі. За допомогою приватного ключа ви зможете підписувати будь-які дії або авторизувати транзакції, пов'язані з вашим публічним ключем. Щодо криптовалют, то публічний ключ буде використовуватися як адреса вашого гаманця, а ви будете використовувати приватний ключ для зняття, відправлення або купівлі цифрових грошей. Ось чому дуже важливо зберігати цей ключ у безпеці. Якщо хтось отримає доступ до вашого приватного ключа, вона може отримати доступ до всіх ваших цифрових активів і зловживати ними [9].

Отже, кожного разу, коли ви надсилаєте щось через мережу, ви повинні це авторизувати. Після транзакції вузли системи перевірять її, і вона буде додана до реєстру за допомогою унікального ідентифікатора і часових рамок. Таким чином, кожна транзакція матиме однакові ознаки: відкритий ключ, цифровий підпис, унікальний ідентифікатор та позначку часу.

Жодна система не є "незламною" в Інтернеті. Але блокчейн гарантує, що вона є однією з найбільш захищених на сьогоднішній день. Він не працює так, як традиційні засоби, такі як банки, тому, щоб зламати його, потрібно зламати весь

комп'ютер, який його використовує. Цей процес надзвичайно складний, і саме тому він вважається надзвичайно захищеним. Щоб обчислити всі ресурси, хакерам доводиться стикатися з багатьма проблемами через величезну кількість комп'ютерів [10].

Як і будь-яка інша технологія, блокчейн має пройти довгий шлях розвитку. Однак, якщо порівнювати блокчейн з централізованими платформами, то блокчейн, безумовно, виграє. Порівняємо їх за кількома основними параметрами.

Незалежність. Головною привабливою особливістю блокчейну є відсутність центрального адміністрування. Це дозволяє здійснювати транзакції напряму. Транзакції, здійснені в системі блокчейн, вимагають авторизації та підтвердження дійсності. Ці дві умови забезпечують безпеку транзакцій, на відміну від централізованих систем. Вузли, що використовуються в блокчейні, будуть перевіряти ваші транзакції і обробляти платіж, завжди синхронізовані один з одним. Зазвичай вміст стандартних баз даних зберігається на якійсь комп'ютерній системі, де ним можна легко маніпулювати. В результаті вам доведеться залежати від системи, незважаючи ні на що. Блокчейн позбавляє від цих систем і захищає дані за допомогою криптографії високого рівня. Таким чином, як тільки код написаний, ніхто не може підробити його за допомогою будь-яких засобів.

Складність. Кожен вузол системи працює незалежно в системі блокчейн. Вузли можуть це робити, тому що вони мають повний доступ до бази даних. Отже, якщо відбуваються якісь зміни, всі вони відповідно оновлюють свій реєстр. Однак, для фінансової транзакції повна прозорість не є найкращим варіантом. Стандартні централізовані системи баз даних уникають подібних проблем, обмежуючи свободу користувачів. Це може бути як хорошою, так і поганою стратегією. Централізована база даних має можливість читати або записувати будь-які дані, але з іншого боку, блокчейн дозволяє тільки записувати. Ось чому блокчейн повинен надати додатковий рівень безпеки, щоб вирішити цю проблему. Знову ж таки, використання декількох адрес блокчейну та криптографії роблять цю процедуру ще складнішою, ніж зазвичай. Однак, таким чином вона є набагато безпечнішою.

Надійність. Як цілісна система, блокчейн має вищу межу витривалості до будь-якої кількості збоїв. Всі вузли тут незалежні, що робить будь-яку кількість втрат керованою. Навіть якщо вузол або кілька вузлів виходять з ладу, інші вузли мають оновлений реєстр і можуть продовжувати роботу без будь-яких проблем. Отже, якщо ви надсилаєте будь-які дані через систему, існує ймовірність нульової помилки транзакції. Це процедура, коли у вас завжди є спосіб виконати дії, якщо вузол вийде з ладу. Звичайні бази даних не мають такого типу відмовостійкої системи. Ось чому в типовій системі може бути багато помилок транзакцій.

Продуктивність. Незважаючи на те, що блокчейн порівняно повільніший за централізовані бази даних, технологія є надзвичайно новою і зазнала багато змін для вирішення цієї проблеми. Вже було багато випадків, коли система була модифікована таким чином, щоб працювати так само швидко, як і централізована база даних. Перевірка підпису з використанням приватно-публічної криптографії займає багато часу. Оскільки вузли будуть ініціювати транзакцію між двома користувачами, ніхто не може уникнути цієї процедури. Однак багато криптовалют, таких як Nano або Litecoin, використовують технологію блокчейн, яка працює швидше, ніж зазвичай.

Різниця між блокчейном і централізованою базою даних досить велика. Однак, централізована система існує століттями і досі працює без будь-яких проблем. Оскільки блокчейн - це нова технологія, потрібно трохи більше часу, щоб звикнути до неї.

Існують різні види технології блокчейн, які можна використовувати. Один з них - публічний блокчейн. Назва говорить сама за себе. Така мережа блокчейн повністю відкрита для всіх. Нікому не потрібен дозвіл на участь у публічному блокчейні або запит про нього. Будь-який процес транзакції через такий блокчейн підтверджується для всіх. Кожен має повне право завантажити код і запустити будь-який публічний вузол на власному пристрої. Будь-хто може навіть перевірити свій поточний статус, а також прийняти рішення про додавання будь-яких блоків в ланцюг. Крім того, ви можете легко приєднатися до мережі, також вона не потребує жодного дозволу на валідацію від кого б то не було. Такі характеристики

приваблюють все більше людей, щоб стати частиною технології блокчейн, яка є загальнодоступною. Система або сервери не потребують жодного обслуговування. Тому створення та запуск децентралізованих додатків коштує не надто дорого.

Одними з провідних публічних блокчейн-мереж є Bitcoin, Ethereum, Dash, Dogecoin, Litecoin, Monero. Всі ці блокчейн-мережі мають доступ для будь-кого, кожен може приєднатися до них як учасник, використовувати свої коди, а також надсилати будь-які транзакції або навіть читати їх [10].

Ще один вид - це федеративний блокчейн. Його ще називають консорціумним блокчейном. Це свого роду протилежність публічній мережі блокчейн. Це тому, що люди не мають доступу до реєстру та будь-якого процесу перевірки транзакцій. Не кожен має права читати цей блокчейн. Федералізованими блокчейнами керує група лідерів. Це основна причина назви цього блокчейну. Конфіденційність транзакцій в цій системі краща. Вони ставляться до цього з великою обережністю. Навіть масштабованість вище, ніж у публічних. Група її лідерів контролює всі процеси консенсусу. Вона надзвичайно швидка. Як правило, банківські сектори використовують федеративні блокчейни.

У цій мережі блокчейн ніхто не має права приєднатися і брати участь у системі без будь-якого дозволу. Вам знадобиться відповідний дозвіл, перш ніж брати участь в будь-якому процесі цієї мережі.

Федеративний, також відомий як консорціумний блокчейн, замінює застарілі системи. Він усуває надмірність даних та змушує людей безпосередньо працювати з документами. Немає ніяких напівавтоматичних механізмів дотримання вимог. Вартість транзакції також є меншою в порівнянні із іншими видами мереж. B3i (страхування), R3 (банки), Corda, EWF (енергетика) - це деякі з федеративних або консорціумних блокчейнів. У деяких випадках можна сказати, що основна відмінність між приватним і федеративним блокчейном полягає в тому, що за винятком групи організацій, тільки одна організація контролює процес роботи приватної мережі блокчейн (Рисунок 5).

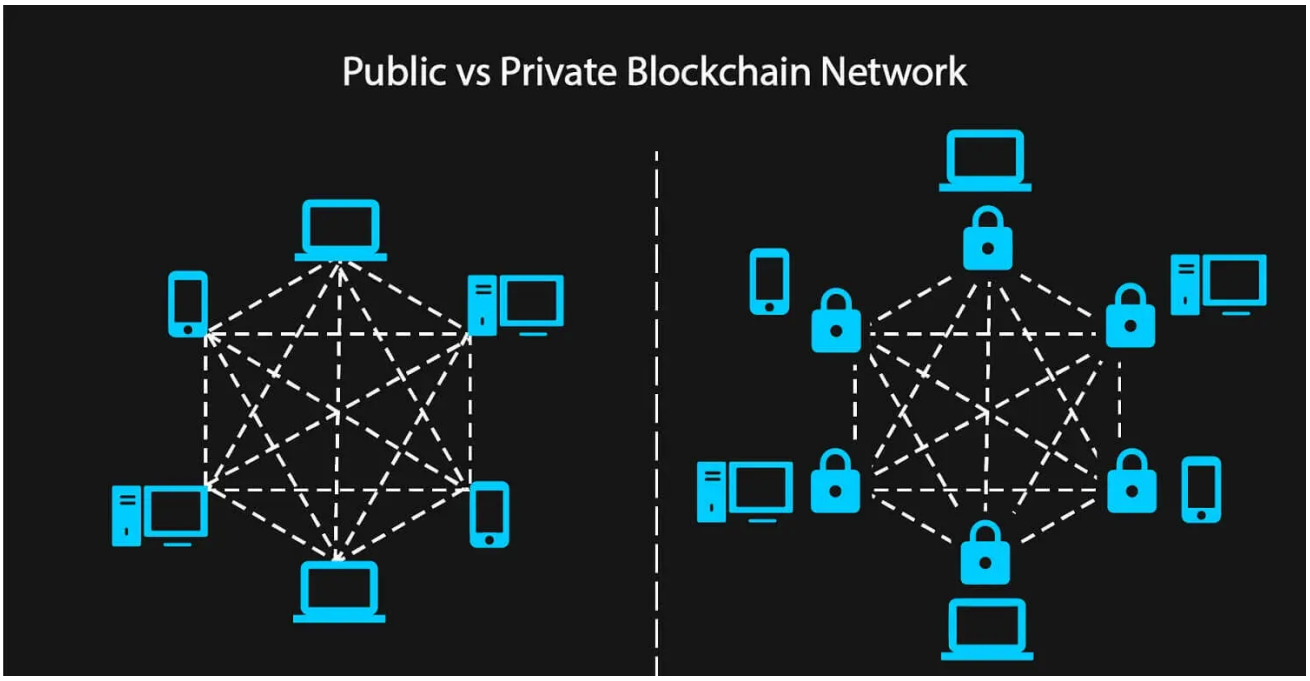

Рисунок 5 - Порівняння публічного та приватного блокчейна

Зазвичай люди не мають доступу до мережі блокчейн, наприклад, до аудиту та управління базами даних. Окремі компанії можуть отримати такий доступ за наявності відповідного дозволу та процедури. Вони зможуть отримати доступ для внутрішньої перевірки транзакцій. Але є випадки, коли громадськість може отримати можливість проводити аудит. Приватна технологія Blockchain в деяких випадках протилежна публічній. Це централізована система. З приватними блокчейнами збільшується ризик порушення безпеки.

Вартість транзакції менша в приватній мережі блокчейн. Вони змінюють системи успадкування. З цією мережею блокчейн набагато простіше працювати з документами. Multichain, MONAX - це дві приватні мережі блокчейн.

Сьогодні технологія блокчейн існує у вигляді різних версій, кожна з яких мала певні оновлення та нові можливості. Щоб краще зрозуміти перше покоління, вам доведеться ближче познайомитися з технологією Bitcoin. Біткоїн був першим, хто представив цю нову технологію, тому кожна функція цієї мережі вважається першим поколінням. Завдяки технології блокчейн, біткойн з самого початку привернув до себе загальну увагу. Цей блокчейн першого покоління має фіксовану політику, яку ніхто не може змінити. Ви зможете добути лише 21 мільйон

біткоінів, і після досягнення цього порогу ніхто не може їх добути або отримати в свої руки, якщо тільки вони не куплять їх. Політика також передбачає, що користувач може витратити свої біткоіни лише один раз. Після здійснення транзакції вони не зможуть отримати їх знову [10]. Що ж, переваг у цього типу блокчейну першого покоління дуже багато:

- Ніхто не зазнає збитків через втрату одного вузла. Дані збережуться і в кожному іншому вузлі.
- Якщо хтось спробує зламати його або підробити записи, процес буде занадто очевидним.
- Сесії також оновлюються кожні 10 хвилин, тому зламати її практично неможливо.
- Вам не доведеться сліпо довіряти жодній стороні. Кожна транзакція автоматично перевіряється і підписується кожним учасником.
- Технологія блокчейн першого покоління стала однією з революцій цифрового світу. На її основі з'явилися всі інші криптовалюти та останні версії технології блокчейн.

Ethereum є основою технології блокчейн другого покоління. Розглянемо цю систему ближче, щоб краще розібратися в ній. Ethereum часто ставлять в пару з біткоіном. Він вводить нову функцію під назвою "Розумні контракти", і ви можете відправляти гроші, використовуючи їх валюту Ether.

Смарт-контракт - це закодований програмний проєкт, який завантажується в мережу. Після виконання всіх контрактів він автоматично оновлюється в реєстрі. Контракти мають конкретні вимоги, і після їх виконання проект буде завершено. Це забезпечує повну прозорість, оскільки ніхто не може підробити дані.

Третє покоління додає додатковий рівень ефективності до всієї мережі. Візьмемо для прикладу операційну систему Apple та Android. Вони обидві мають різний набір кодів для своїх додатків. Таким чином, додаток для Android не буде працювати на iOS від Apple і навпаки. Якщо ви хочете запустити точний додаток на Apple, вам потрібно буде написати його на Swift. React native фактично вирішив цю проблему. Він дозволяє будь-кому написати будь-який додаток для обох

платформ. Цю структуру і впроваджує третє покоління. Ethereum - чудова платформа, але їй не вистачає гнучкості. Кожен децентралізований додаток, побудований на Ethereum, працює тільки на своїй EVM (Ethereum Virtual Machine). Отже, якщо ви хочете перенести його на іншу платформу, скажімо, Lisk, вам доведеться змінити код. Третє покоління дозволяє гнучко обмінюватися платформами і використовувати блокчейн-додатки в будь-якій мережі.

Виділимо головні особливості технології Blockchain:

Незмінність. Існує кілька цікавих особливостей блокчейну, але серед них незмінність є однією з ключових особливостей технології блокчейн. Незмінність означає те, що не може бути змінено або перероблено. Це одна з головних особливостей блокчейну, яка допомагає гарантувати, що технологія залишиться такою, якою вона є - постійною, незмінною мережею. Технологія блокчейн працює дещо інакше, ніж типова банківська система. Замість того, щоб покладатися на централізовані органи влади, вона забезпечує функції блокчейну через сукупність вузлів. Кожен вузол системи має копію цифрового реєстру. Щоб додати транзакцію, кожен вузол повинен перевірити її дійсність. Якщо більшість вважає її дійсною, то вона додається до реєстру. Це сприяє прозорості та робить його корупціогенним. Таким чином, без згоди більшості вузлів ніхто не може додати жодного блоку транзакцій до реєстру. Ще одним фактом, який підтверджує одну із ключових особливостей блокчейну, є те, що після додавання блоків транзакцій до реєстру, ніхто не може просто повернутися і змінити його. Таким чином, будь-який користувач мережі не зможе його відредагувати, видалити або оновити.

Децентралізованість. Мережа є децентралізованою, що означає, що вона не має жодного керівного органу або єдиної особи, яка доглядає за структурою. Замість цього мережа підтримується групою вузлів, що робить її децентралізованою. Це одна з ключових особливостей технології блокчейн, яка працює бездоганно. Блокчейн ставить нас, користувачів, у просте становище. Оскільки система не потребує жодного керівного органу, ми можемо отримати до неї прямий доступ з Інтернету і зберігати там свої активи. Ви можете зберігати що

завгодно, починаючи від криптовалют, важливих документів, контрактів або інших цінних цифрових активів. А за допомогою блокчейну ви матимете прямий контроль над ними за допомогою свого приватного ключа. Така децентралізована структура повертає звичайним людям їхню владу та права на їхні активи. Завдяки децентралізованості технологія має такі переваги:

1. Менше збоїв - все в блокчейні повністю організовано, і оскільки він не залежить від людських розрахунків, він дуже відмовостійкий. Таким чином, випадкові збої в цій системі не є звичайним явищем.

2. Контроль користувача - з децентралізацією користувачі тепер мають контроль над своєю власністю. Їм не потрібно покладатися на будь-яку третю сторону для обслуговування своїх активів. Всі вони можуть робити це одночасно самостійно.

3. Менша вразливість до атак - оскільки децентралізація є однією з ключових особливостей технології блокчейн, вона може пережити будь-яку зловмисну атаку. Це пов'язано з тим, що атака на систему обходиться хакерам дорожче і є непростим рішенням. Отже, вона менш схильна до поломки.

4. Відсутність третьої сторони - децентралізований характер технології робить її системою, яка не покладається на сторонні компанії; немає третьої сторони, немає додаткового ризику.

5. Нуль шахрайства - оскільки система працює на алгоритмах, у людей немає жодного шансу виманити у вас що-небудь. Ніхто не може використовувати блокчейн для особистої вигоди.

6. Прозорість - децентралізована природа технології створює прозорий профіль кожного учасника. Кожна зміна в блокчейні є видимою і робить його більш конкретним.

7. Автентичність - така природа системи робить її унікальним видом системи для кожного типу людей. І хакерам буде важко її зламати.

Підвищена безпека. Оскільки зникає потреба в центральному органі, ніхто не може просто так змінити будь-які характеристики мережі на свою користь. Використання шифрування забезпечує ще один рівень безпеки системи.

Додана до децентралізації, криптографія створює ще один рівень захисту для користувачів. Криптографія - це досить складний математичний алгоритм, який діє як брандмауер для атак. Кожна інформація в блокчейні хешується криптографічно. Простими словами, інформація в мережі приховує справжню природу даних. Для цього процесу будь-які вхідні дані проходять через математичний алгоритм, який виробляє різного роду значення, але довжина завжди фіксована. Це можна розглядати як унікальний ідентифікатор для кожної інформації. Всі блоки в книзі мають власний унікальний хеш і містять хеш попереднього блоку. Отже, зміна або спроба підробки даних означатиме зміну всіх хеш-ідентифікаторів, але це неможливо. Ви матимете приватний ключ для доступу до даних, але матимете публічний ключ для здійснення транзакцій. Хешування є досить складним процесом, і його неможливо змінити або скасувати. Ніхто не може взяти публічний ключ і придумати приватний ключ. Крім того, одна зміна у вхідних даних може призвести до зовсім іншого ідентифікатора, тому невеликі зміни не є розкішшю в системі. Якщо хтось захоче пошкодити мережу, йому або їй доведеться змінити всі дані, що зберігаються на кожному вузлі мережі. Це можуть бути мільйони і мільйони людей, у кожного з яких є однакова копія запису в реєстрі. Отримати доступ до мільйонів комп'ютерів і зламати їх практично неможливо і дуже дорого. Ось чому це одна з найкращих особливостей блокчейну. Оскільки її дуже важко обійти, вам не доведеться турбуватися про те, що хакери заберуть у вас всі ваші цифрові активи [10].

Розподілені реєстри. Зазвичай, публічний реєстр містить всю інформацію про транзакцію та її учасника. Це все у відкритому доступі, нікуди не сховаєшся. Хоча у випадку з приватним або федеративним блокчейном все трохи інакше. Але все одно, в тих випадках багато людей можуть бачити, що насправді відбувається в реєстрі. Це тому, що реєстр в мережі підтримується всіма іншими користувачами системи. Це розподіляє обчислювальну потужність між комп'ютерами для забезпечення кращого результату. Саме тому це вважається однією з найважливіших особливостей блокчейну. Результатом завжди буде більш

ефективна система обліку, яка може конкурувати з традиційними системами. Завдяки цій властивості досягаються наступні результати:

1. Ніяких зловмисних змін - розподілений реєстр дуже добре реагує на будь-яку підозрілу діяльність або втручання. Оскільки ніхто не може змінити реєстр і все оновлюється дуже швидко, відстежити, що відбувається в реєстрі, досить легко за допомогою всіх цих вузлів.

2. Власність верифікації - тут вузли діють як верифікатори реєстру. Якщо користувач хоче додати новий блок, інші повинні будуть перевірити транзакцію, а потім дати зелений сигнал. Це забезпечує користувачеві справедливу участь.

3. Ніяких додаткових послуг - ніхто в мережі не може отримати ніяких особливих привілеїв від мережі. Кожен повинен пройти через звичайні канали, а потім додати свої блоки. Це не означає, що у вас більше влади, тому ви отримаєте більше привілеїв.

4. Управління - для того, щоб функції блокчейну працювали, кожен активний вузол повинен вести реєстр і брати участь в перевірці.

5. Швидке реагування - як я вже говорив раніше, видалення проміжних вузлів прискорює реакцію системи. Будь-яка зміна в реєстрі оновлюється за лічені хвилини або навіть секунди.

Консенсус. Кожен блокчейн існує завдяки алгоритмам консенсусу. Архітектура розроблена з розумом, і алгоритми консенсусу лежать в основі цієї архітектури. Кожен блокчейн має консенсус, який допомагає мережі приймати рішення. Простіше кажучи, консенсус - це процес прийняття рішень для групи вузлів, активних в мережі. Тут вузли можуть прийти до згоди швидко і відносно швидше. Коли мільйони вузлів підтверджують транзакцію, консенсус абсолютно необхідний для безперебійної роботи системи. Це як про своєрідну систему голосування, де більшість перемагає, а меншість повинна її підтримати. Консенсус відповідає за те, що мережа не викликає довіри. Вузли можуть не довіряти один одному, але вони можуть довіряти алгоритмам, які працюють в її основі. Ось чому кожне рішення в мережі - це виграшний сценарій для блокчейну. Це одна з переваг блокчейну. У світі існує безліч різних алгоритмів консенсусу для блокчейнів.

Кожен з них має свій власний унікальний спосіб прийняття рішень, і вдосконалення раніше введених помилок. Така архітектура створює концепцію справедливості в Інтернеті. Однак, щоб підтримувати децентралізацію, кожен блокчейн повинен мати алгоритм консенсусу, інакше втрачається його основна цінність.

Прискорення розрахунків. Традиційні банківські системи досить повільні. Іноді на обробку транзакції після завершення всіх розрахунків може знадобитися кілька днів. Крім того, вона може бути досить легко корумпована. Блокчейн пропонує більш швидкі розрахунки в порівнянні з традиційними банківськими системами. Таким чином, користувач може переказувати гроші відносно швидше, що економить багато часу в довгостроковій перспективі.

Ці функції блокчейну полегшують життя іноземним працівникам і допомагають зрозуміти, чому блокчейн важливий. Багато людей виїжджають в іншу країну в пошуках кращого життя і роботи і залишають там свої сім'ї. Однак, відправка грошей своїм сім'ям за кордон займає багато часу і може стати фатальною в скрутну хвилину. Зараз блокчейн дуже швидкий, і вони можуть легко використовувати його для відправки грошей своїм близьким. Ще один цікавий факт - система смарт-контрактів. Це може дозволити здійснювати більш швидкі розрахунки за будь-якими видами контрактів. Це одна з найкращих переваг функцій блокчейну на сьогоднішній день. Хоча бувають випадки, коли мережа не справляється з підтримкою занадто великої кількості користувачів і швидкі розрахунки неможливі. Незважаючи на це, багато хто працює над поліпшенням цього сценарію, і незабаром ми побачимо кращий погляд на це питання.

Незважаючи на велику кількість переваг в технології Blockchain є ряд недоліків:

Надлишкова продуктивність. Насправді, обчислювальні потреби цієї технології більш повторювані, ніж у централізованих серверів. Це пов'язано з тим, що кожного разу, коли реєстр оновлюється, всі вузли також повинні оновлювати свою версію реєстру. Це пов'язано з тим, що розподілена природа облікової

системи вимагає, щоб кожен вузол мав копію облікової системи. Таким чином, він повинен проходити той самий процес знову і знову [10].

Складний процес перевірки підпису. Ще одним мінусом є процес перевірки підпису. В основному, для кожної транзакції в системі вам знадобиться приватно-публічна перевірка криптографічного підпису. Потім використовується ECDSA (алгоритм цифрового підпису еліптичної кривої), щоб гарантувати, що транзакція відбувається між правильними вузлами. Таким чином, кожен вузол повинен перевірити справжність користувача, що може бути складним процесом.

Приватні ключі. Щоб здійснювати транзакції в мережі, вам потрібно мати приватний ключ. Незважаючи на те, що інші користувачі можуть бачити ваш відкритий ключ, приватний ключ набагато важливіший, оскільки він залишається прихованим [10]. Крім того, всі адреси блокчейну матимуть приватний ключ. Якщо ви не хочете, щоб інші люди зловживали вашими активами, вам необхідно зберігати свій приватний ключ у безпеці будь-якими способами. Однак, якщо ви втратите свій приватний ключ, ви також втратите доступ до своїх коштів в мережі. Відновити їх вже не вдасться.

Відсутність власних можливостей. Оскільки технологія є відносно новою концепцією, існує не так багато здібних розробників, які можуть працювати над нею. Тому, коли підприємства намагаються розробити власне корпоративне блокчейн-рішення, знайти здібну команду для роботи над проектом стає важко [10]. Провайдери BAAS пропонують багато послуг в цьому випадку. Ці послуги пропонують висококласних розробників і маркетингові команди, які допоможуть вам вивести ваше блокчейн-рішення на ринок.

Проблеми інтеграції. Це ще один з основних мінусів технології блокчейн. В основному це стосується підприємств, які використовують застарілі мережі. Насправді, блокчейн замінить застарілі мережі. Однак процес інтеграції все ще не повністю функціонує. Більше того, багато технологій блокчейн не мають можливості працювати разом із застарілими мережами. Це означає, що для того, щоб використовувати їх належним чином, компаніям доведеться повністю позбутися своїх застарілих мереж. Це те, до чого багато хто ставиться скептично.

Невизначеність регулювання. Насправді, не всі технології блокчейн супроводжуються належним набором правил роботи в мережі. Тому багато хто взагалі не довіряє системі. З іншого боку, відсутність регулювання породжує поняття шахрайства з ICO. І не варто говорити, що багато хто став жертвою ICO-шахрайства через відсутність регулювання, пов'язаного з криптовалютами. Державні установи також намагаються прийняти його, оскільки цей сектор працює повністю на нормативних актах.

Велике споживання енергії. Щоб гарантувати, що кожна транзакція є дійсною, вона повинна пройти через процес консенсусу. Очевидно, що процес консенсусу вимагає величезної кількості зусиль для формування кожного вузла. Не кажучи вже про те, що всі вузли повинні взаємодіяти між собою, щоб переконатися, що транзакція є дійсною. З іншого боку, алгоритми консенсусу, такі як доказ роботи, вимагають великої обчислювальної потужності, що збільшує загальне енергоспоживання. Однак, ви будете раді дізнатися, що зараз кілька протоколів консенсусу споживають набагато менше енергії.

Відсутність контролю для підприємств. Підприємствам потрібен певний авторитетний процес для того, щоб використовувати технологію блокчейн. На жаль, публічні блокчейни не можуть запропонувати ці аспекти контролю найближчим часом. Однак, поява приватних та консорціумних блокчейнів, здається, пропонує як контроль, так і розподілений характер технології. Вони в основному називаються корпоративними блокчейн-фреймворками, і вони підходять тільки для організацій.

Занепокоєння щодо конфіденційності. Підприємствам потрібна конфіденційність, щоб зберегти вартість свого бренду за будь-яку ціну. Очевидно, що вони не можуть розкривати свою конфіденційну інформацію широкому загалу або своїм конкурентам. Тому багато організацій не дуже охоче використовують блокчейн для корпоративних бізнес-цілей. Однак, використання опцій приватних транзакцій та аутентифікації для конкретного користувача потроху вирішує цю проблему.

Культурна адаптація. Це скоріше культурний недолік, ніж технічний. Наші бізнес-моделі вже досить давно працюють на певній архітектурі. Але винахід блокчейну загрожує цій системі. Насправді, блокчейн вже почав змінювати те, як працює система, і порушив роботу багатьох галузей. Як результат, багато ринкових майданчиків вже застаріли.

Висока вартість. Блокчейн набагато дешевший за інші інфраструктури. Однак, він може бути і дорогим рішенням. В основному, вартість залежить від того, який тип функції ви хочете додати, і ваших потреб. Не кажучи вже про те, що заміна старої системи також буде коштувати чималих грошей. Однак, ви можете подолати цю проблему, якщо зведете свої рішення до мінімуму. Крім того, приєднання до консорціуму або використання BAAS також може допомогти. Blockchain як послуга може допомогти вам у розробці надійної бізнес-стратегії на основі блокчейну [10].

З кожною технологією вам доведеться мати справу з недоліками, якщо ви хочете отримати вигоду від переваг. Однак недоліки ніколи не повинні перекривати переваги. Тому блокчейну дійсно потрібно активізувати гру. І саме це він зараз і робить. Більшість мінусів блокчейн-додатків можуть стати плюсами, якщо компанії використовують приватні або консорціумні блокчейни. Приватні блокчейни пропонують додатковий контроль, конфіденційність, регулювання, більш швидкі транзакції, недорогі інтеграції та багато іншого.

Крім того, якщо ви хочете отримати децентралізацію блокчейну, ви завжди можете використовувати консорціумний блокчейн для своєї компанії. З плином часу все більше людей цікавляться технологією блокчейн і отримують навички, необхідні для побудови кар'єри в цій сфері. Це знімає питання нестачі кваліфікованих спеціалістів та створює нові можливості як для працівника, так і для роботодавця. Так, існує ймовірність появи нових мінусів в екосистемі блокчейн-додатків, але не варто забувати, що на противагу цьому з'являться і плюси.

Технологія блокчейн - це не просто черговий тренд, про який люди забудуть через кілька днів. З усіма її функціями та додатками, ми можемо з упевненістю

припустити, що вона залишиться надовго. Всі важливі функції блокчейну виводять Інтернет на зовсім інший рівень впливу. Хоча блокчейн викликає багато суперечок, але якщо люди зможуть використовувати ідеологію, що стоїть за всіма перевагами блокчейну, вони зможуть зробити світліше і блискучіше майбутнє для всіх. Не кажучи вже про те, що блокчейн може змінити світ [10].

2.2. Особливості використання мережі Ethereum

Ethereum - це децентралізована блокчейн-платформа з відкритим вихідним кодом та обчислювальними можливостями, які реконструюють елементарний обмін валюти в передачу вартості між користувачами за допомогою мови сценаріїв. Ethereum широко визнаний як наступник протоколу Bitcoin, що узагальнює початкові ідеї та дозволяє створювати більш різноманітні додатки на основі технології блокчейн. Ethereum має два основних компоненти. Перший - це віртуальний процесор, який може завантажувати ресурси та виконувати сценарії, що відповідає принципу Тьюринга, який називається віртуальною машиною Ethereum (EVM). Другий компонент - це токен цінності під назвою Ether, який є валютою мережі, що використовується для транзакцій між користувачами або для виплати винагороди майнерам мережі [11].

Приблизно в середині 2013 року більшість членів біткоїн-спільноти почали думати над чимось більшим, крім простої валюти. Незабаром з'явився потік нових ідей, які обговорювалися на онлайн-форумах. Деякі поширені приклади включають реєстрацію доменів, страхування активів, голосування і навіть Інтернет речей (IoT). Після того, як ажіотаж почав згасати, більш серйозний аналіз протоколу Bitcoin виявив серйозні обмеження потенційних додатків, які можуть бути побудовані на основі блокчейну.

Ключовим моментом дебатів було те, чи слід дозволити повноцінну мову сценаріїв в рамках блокчейну, або ж додатки повинні будуватися з логікою, що знаходиться за межами блокчейну. Було два ключових питання, які викликали ці дебати:

- Мова сценаріїв і OPCODES в протоколі Bitcoin були розроблені таким чином, щоб мати дуже обмежену функціональність.
- Сам протокол був недостатньо загальним, і з'явилися альтернативні валюти, такі як Namecoin та інші, які спеціалізувалися на виконанні однієї конкретної задачі.

Зрештою, з'явилося два методи щодо написання сценаріїв. Перший метод пропонував зберегти мову сценаріїв дуже обмеженою за функціональністю. Це дозволило б уникнути проблем безпеки, пов'язаних з наявністю виконуваного коду в блокчейні. У певному сенсі, виконуваний код блокчейну обмежений кількома необхідними примітивами, які оновлюють розподілені стани. Другу метод вважав блокчейн чимось більшим, ніж просто реєстр. Він уявляв блокчейн як обчислювальну платформу, яка може виконувати чітко визначені функції, використовуючи контракти та аргументи. Конструкція EVM дозволяє забезпечити повну ізоляцію виконуваного коду і безпечне виконання додатків, побудованих поверх EVM. Розглянемо принципи побудови та основну ідею, що лежить в основі Ethereum.

Перша особливість Ефіріуму - це консенсус. У криптовалютах на основі PoW, таких як Bitcoin, мережа нагороджує майнерів, які вирішують криптографічні головоломки для підтвердження транзакцій і видобутку нових блоків. В Ethereum використовується інший алгоритм консенсусу, який називається PoS. В алгоритмі PoS валідатор або творець наступного блоку вибирається псевдо випадковим чином на основі частки, яку має обліковий запис в мережі. Таким чином, якщо у вас більша частка в мережі, у вас більше шансів бути обраним в якості валідатора. Після цього валідатор підробить наступний блок і отримує винагороду від мережі. Тут валідатор дійсно підробляє блок, а не майнить, тому що в PoS ідея апаратного майнінгу замінена цією віртуальною часткою [11]. Певною мірою, доцільність використання PoS була пов'язана з високими енергетичними вимогами алгоритмів PoW, які стали частою скаргою. Першою криптовалютою, яка запустила PoS, була Peercoin, але більш відомі

нещодавні реалізації PoS можна побачити в ShadowCash, Nxt та Qora. Основні відмінності між Bitcoin та Ethereum як протоколами показані на Рисунку 6

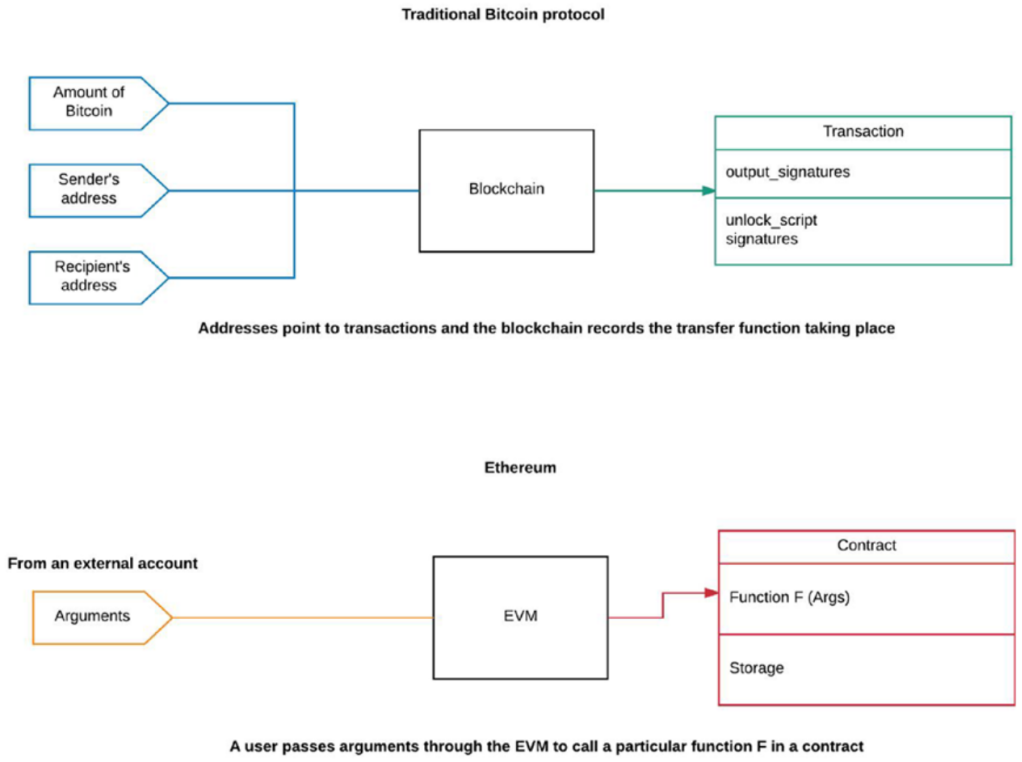

Рисунок 6 - Порівняння схем роботи протоколів Bitcoin та Ethereum

У протоколі Bitcoin адреси відображають транзакції від відправника до одержувача. Єдина програма, яка працює в блокчейні - це програма переказу. Маючи адреси та ключовий підпис, ця програма може переказувати гроші від одного користувача до іншого. Ethereum узагальнює цю концепцію, розміщуючи EVM на кожному вузлі, щоб код, який можна перевірити, міг виконуватися в блокчейні. Загальна схема полягає в тому, що зовнішній обліковий запис передає аргументи функції, а EVM направляє цей виклик до відповідного контракту і виконує функцію за умови, що відповідна кількість Ehter буде поставлена. Як наслідок, кожна транзакція в Ethereum може розглядатися як виклик функції. Виклики функцій і транзакції в Ethereum відповідають стандарту PoS, який має більш швидкий час дозволу, ніж блокчейн Bitcoin, який покладається на PoW.

Рівень безпеки цього процесу, що перевіряється мережею, також дуже високий [11].

Облікові записи є метаструктурою в Ethereum і фундаментальною операційною одиницею блокчейну. Облікові записи також служать моделлю для зберігання і відстеження інформації про користувачів в мережі. У мережі доступні два типи облікових записів.

Облікові записи користувачів - це облікові записи, контрольовані користувачем, також відомі як зовнішні облікові записи. Ці облікові записи мають баланс Ефіру, контролюються парами публічних і приватних ключів і можуть відправляти транзакції, але не мають пов'язаного з ними коду. Всі дії в мережі Ethereum ініціюються транзакціями, ініційованими зовнішніми акаунтами. У протоколах Bitcoin ми називали їх просто адресами. Ключова відмінність між обліковими записами і адресами полягає в здатності містити і виконувати узагальнений код в Ethereum [11].

Контракти - це обліковий запис, керований власним кодом. Рахунок контракту - це функціональна програмна одиниця в Ethereum, яка знаходиться в блокчейні. Цей обліковий запис має баланс Ефіру, пов'язаний з ним код, може виконувати код при отриманні транзакцій від інших облікових записів і може маніпулювати своїм власним постійним сховищем. Кожен контракт в блокчейні має власне сховище, в яке тільки він може записувати - це називається станом контракту. Будь-який учасник мережі може створити додаток з деякими арбітражними правилами, визначивши її як контракт [11].

Контракт може маніпулювати власним сховищем і оновлювати стан. В протоколі Bitcoin дані про користувачів і транзакції оформляються і зберігаються в контексті UTXO. В Ethereum застосовується інша стратегія проектування використання об'єкта стану. По суті, в стані зберігається список рахунків, де кожен рахунок має баланс, а також специфічні для блокчейну дані (код і сховище даних).

Транзакція вважається дійсною, якщо на рахунку-відправнику достатньо коштів для її оплати, тому рахунок-відправник дебетується, а рахунок-одержувач зараховує вартість. Якщо з рахунком одержувача пов'язаний код, то при отриманні

транзакції він буде виконаний. Виконання контракту або коду, пов'язаного з рахунком, може мати різні наслідки для стану: внутрішнє сховище також може бути змінено, або код може навіть створити додаткові транзакції для інших облікових записів.

Ethereum розрізняє стан та історію в мережі. Стан - це, по суті, знімок поточної інформації про стан мережі та облікових записів на певний момент часу. З іншого боку, історія - це компіляція всіх подій, що відбулися в блокчейні, таких як виклики функцій (транзакції) і зміни, що відбулися в результаті (квитанції). Більшість вузлів в мережі Ethereum ведуть облік стану [11]. Також можна сказати, що стан - це структура даних, яка містить ключові значення, що відображають адреси об'єктів облікового запису. Кожен об'єкт рахунку містить чотири значення:

- Поточна вартість номіналу
- Баланс рахунку (в ефірах)
- Код-хеш, який містить код в разі контрактів, але залишається порожнім для зовнішніх рахунків
- Корінь сховища, що містить код і дані, які зберігаються в блокчейні

Далі поговоримо про газ в Ethereum. Газ - це внутрішня одиниця обліку витрат на виконання в Ethereum. Іншими словами, це плата мікротранзакції за виконання обчислення в блокчейні. Для такої обчислювальної платформи, як Ethereum, це стає критично важливим при виконанні коду через проблему зупинки: неможливо сказати, чи буде програма працювати нескінченно, або просто має тривалий час виконання. Газ накладає обмеження на час виконання, оскільки користувач повинен платити за виконання покрокових інструкцій контракту. Природа мікротранзакцій дозволяє виконувати кроки дуже дешево, але навіть ці транзакції складатимуть дуже довгий час. Як тільки газ, що постачається за контрактом, вичерпується, користувач повинен буде платити за його продовження. Спеціальні тарифи на газ також застосовуються до операцій, пов'язаних зі зберіганням.

Такі операції, як зберігання, пам'ять і обробка, коштують газу в мережі Ethereum. Поговоримо про зберігання далі. В Ethereum зовнішні облікові записи

можуть зберігати дані в блокчейні за допомогою контрактів. Контракт буде керувати процесом завантаження і зберігання, але типи даних, які можна зберігати в даний час, дуже обмежені. В даний час існує два механізми, що запобігають перевантаженню даних:

- Ліміти газу на блок, які диктують, скільки газу може бути витрачено на блок на зберігання і обчислювальні операції
- кількість грошей, яку користувач повинен буде витратити на придбання газу, необхідного для зберігання даних

Далі показана спрощена схема роботи мережі Ethereum (Рисунок 7).

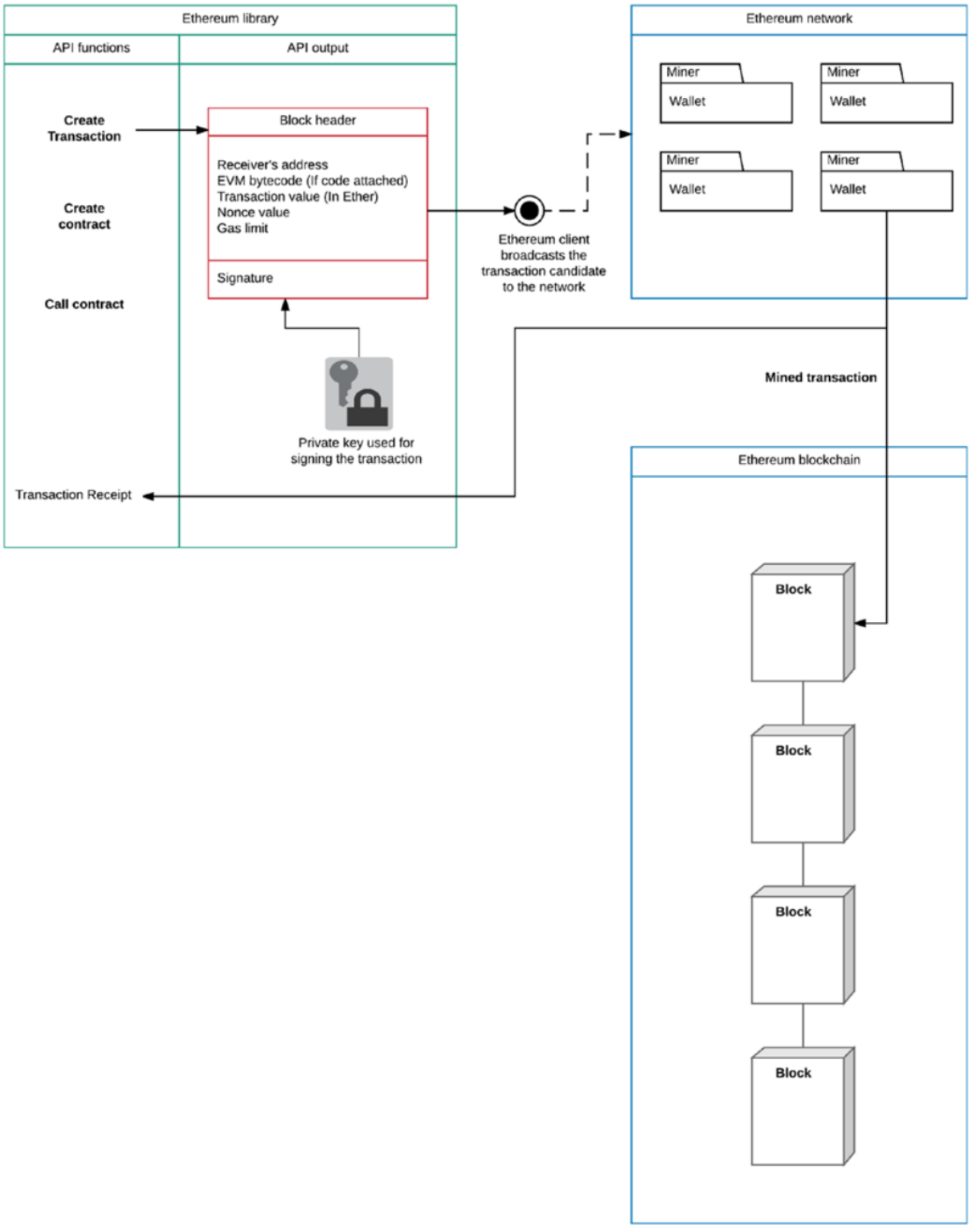

Рисунок 7 - Схема роботи мережі Ethereum

На схемі можна побачити три важливі компоненти Ethereum: API, мережа та блокчейн. JavaScript API Ethereum (також відомий як web3.js) надає широкий набір

функцій, таких як створення транзакцій та контрактів, посилання на функції та зберігання квитанцій. Клієнт гаманця для Ethereum, такий як MetaMask, бере на себе деякі з цих функцій за допомогою графічного інтерфейсу. Як тільки блок-кандидат створений, клієнт Ethereum транслює його в мережу. Валідатори в мережі визначають, чи є транзакції дійсними, і чи є дійсним будь-який код (в блоці), пов'язаний з транзакцією або контрактом. Після завершення перевірки валідатори виконують відповідний код і застосовують його до поточного стану. Блок транслюється в мережу, і майнер підробляє його, після чого перевірений блок додається в блокчейн. На цьому етапі також створюються квитанції про транзакції для кожної транзакції, включеної в блок. Новий блок також забезпечує оновлення об'єктів стану і реляційних зв'язків для стану від поточного блоку до нового блоку [11].

Що запобігатиме роздуванню мережі ethereum за рахунок дрібних невикористовуваних контрактів? В даний час не існує механізмів контролю за тривалістю життя контракту, проте є кілька пропозицій щодо тимчасових контрактів на основі підписки. В майбутньому може з'явитися два різних типи контрактів: один з постійним терміном життя (який значно дорожче створювати і обчислювати), а інший - який діє до закінчення терміну підписки (дешевший і тимчасовий; самознищується після закінчення підписки, щоб запобігти захаращенню мережі).

Розглянемо детальніше, що таке EVM (Ethereum Virtual Machine). Формально EVM є середовищем виконання смарт-контрактів в Ethereum. Контракти пишуться на мові більш високого рівня, яка називається Solidity, а потім компілюються в байт-код за допомогою інтерпретатора в EVM. Потім цей байт-код завантажується в блокчейн за допомогою клієнта Ethereum. Контракти живуть в блокчейні в цій виконуваній формі байт-коду. EVM розроблений таким чином, щоб бути повністю ізольованим від навколишнього середовища і решти мережі. Код, що виконується всередині EVM, не має доступу до мережі або будь-яких інших процесів і тільки після компіляції в байт-код контракти отримують доступ до зовнішнього світу та інших контрактів.

З операційної точки зору EVM поводиться як великий децентралізований комп'ютер з мільйонами об'єктів (облікових записів), які мають можливість підтримувати внутрішню базу даних, виконувати код і спілкуватися один з одним за допомогою передачі повідомлень. Ця модель ще не є завершеною, але в Ethereum цю концепцію часто називають ідеєю світового комп'ютера. EVM дозволяє будь-якому користувачеві в мережі виконувати довільний код в недовірливому середовищі, де результат повністю детермінований і виконання може бути гарантовано. Середовище виконання та налаштування за замовчуванням призводять до застою, тобто в мережі нічого не відбувається і стан всього залишається незмінним [11]. Однак, як ми вже згадували раніше, будь-який користувач може спровокувати дію, відправивши транзакцію з зовнішнього облікового запису. Тут ми можемо мати два варіанти розвитку подій:

1. Якщо одержувачем є інший зовнішній аккаунт, то транзакція переведе деяку кількість Ефіру, але більше нічого не відбудеться.

2. Якщо одержувачем є контракт, то контракт активується і виконує код всередині. Виконання коду всередині мережі вимагає часу, і цей процес відносно повільний і дорогий. За кожен крок в інструкціях користувачеві нараховується плата за виконання. Коли користувач ініціює виконання через транзакцію, він встановлює верхню межу для максимальної валюти, яку він готовий заплатити як газ за цей контракт або код.

Результат EVM є детермінованим. Для контрактного методу важливо, щоб кожен вузол досягав однакового кінцевого стану при однакових вхідних даних. В іншому випадку, кожен вузол, який виконує код контракту для підтвердження транзакції, закінчував би з різними результатами, і ніякого консенсусу не було б можливо досягти. Це детермінована природа EVM, яка дозволяє кожному вузлу досягти консенсусу щодо виконання контракту і однакового кінцевого стану рахунків. Вузли, що виконують контракт, схожі на гвинтики, синхронізовані для руху всередині годинника, оскільки вони працюють злагоджено і досягають відповідного кінцевого стану. Контракт також може посилатися на інші контракти, але він не може мати прямого доступу до внутрішнього сховища іншого контракту.

Кожен контракт виконується у спеціальному приватному екземплярі EVM, де він має доступ лише до деяких вхідних даних, свого внутрішнього сховища, коду інших контрактів у блокчейні та різних параметрів блокчейну, таких як нещодавні хеші блоків.

Кожен повноправний вузол мережі виконує код контракту одночасно для кожної транзакції. Коли вузол перевіряє блок, транзакції виконуються послідовно, в порядку, визначеному блоком. Це необхідно, оскільки блок може містити декілька транзакцій, які викликають один і той же контракт, і поточний стан контракту може залежати від стану, зміненого попередніми посиланнями під час виконання коду. Виконання коду контракту є відносно дорогим, тому, коли вузли отримують блок, вони виконують лише базову перевірку транзакцій: Чи достатньо Ефіру на рахунку відправника для оплати газу? Чи має транзакція дійсний підпис? Після цього майнінг-вузли виконують відносно дороге завдання по виконанню транзакції, включенню її в блок і збору комісії за транзакцію в якості винагороди. Коли повний вузол отримує блок, він виконує транзакції в блоці, щоб самостійно перевірити безпеку і цілісність транзакцій, які будуть включені в блокчейн [11]. Розглянемо EVM візуально на Рисунку 8.

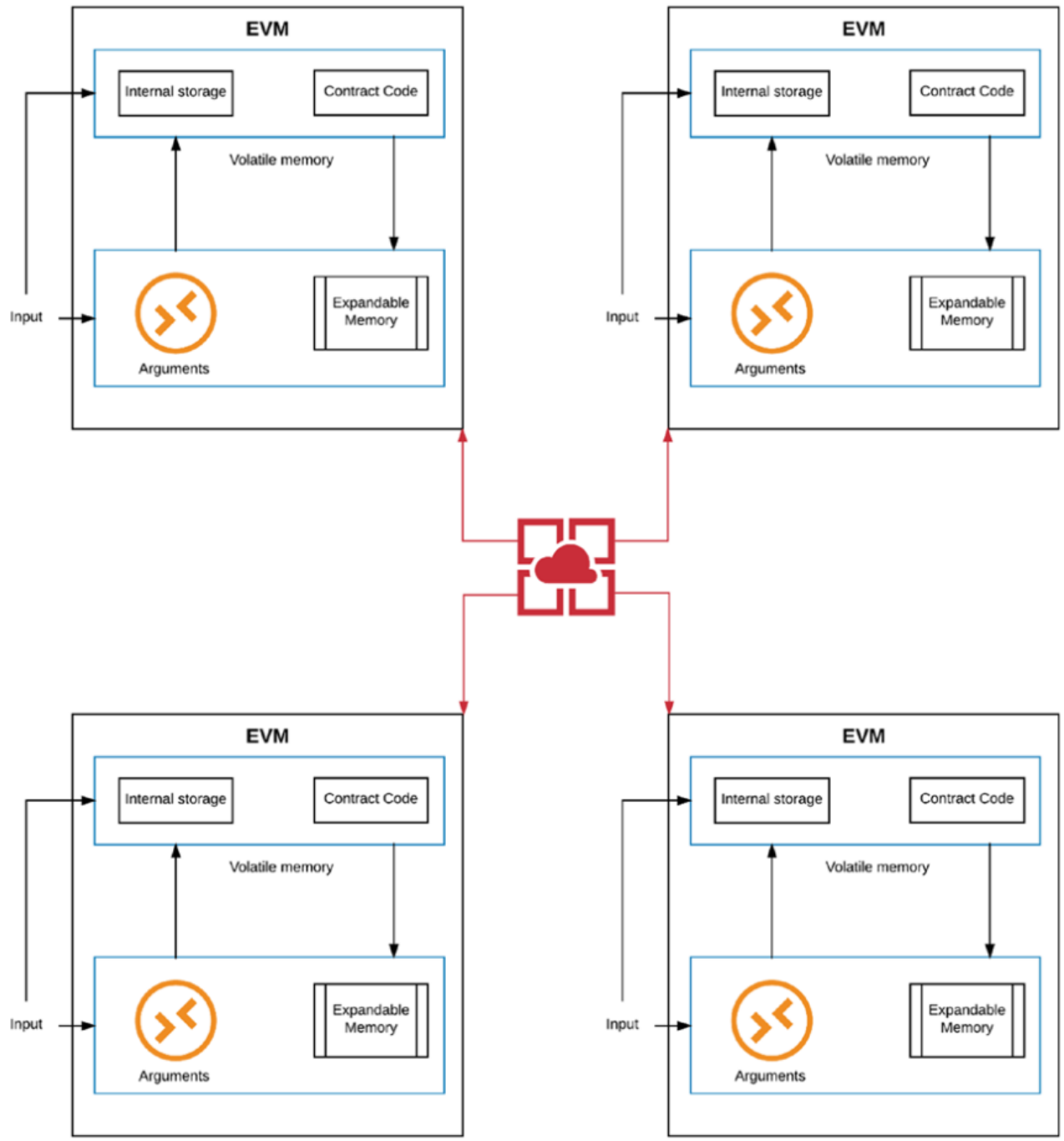

Рисунок 8 - Чотири екземпляри віртуальних машин Ethereum (EVM) працюють на чотирьох різних вузлах

Чотири EVM синхронно виконують інструкції контракту і після завершення виконання приходять до ідентичного стану рахунку. Це пов'язано з детермінованим характером EVM, який дозволяє контракту досягти консенсусу в мережі на кожному кроці виконання інструкцій. EVM має дуже просте

обґрунтування: він має єдиний цикл, який намагається виконати інструкцію - один крок за один раз [11]. В рамках цього циклу для кожної інструкції розраховується газ і, за необхідності, розширюється виділена пам'ять. Цикл буде продовжуватися до тих пір, поки EVM або не отримає код виходу, що вказує на успішне виконання, або не згенерує виняток, наприклад, закінчився газ.

2.3. Мова програмування смарт-контрактів Solidity

Solidity - це об'єктно-орієнтована мова програмування вищого рівня для написання смарт-контрактів в Ethereum. Будь-який код, написаний на Solidity, може бути виконаний на EVM після компіляції в байт-код, який є набором інструкцій для EVM. Як в байт коді кодуються посилання на інші функції і контракти, які викликаються під час виконання? Це робиться за допомогою прикладного бінарного інтерфейсу (Application Binary Interface, ABI). У загальному випадку, ABI - це інтерфейс між двома програмними модулями: інструкціями машинного рівня та мовою програмування більш високого рівня, що читається людиною [11]. Розіб’ємо цю відповідь на три складові:

- Контракт - це просто код вищого рівня, визначений на формальній мові, такій як Solidity.
- Скомпільований контракт - контракт перетворюється в байт-код для виконання на EVM, дотримуючись специфікації компілятора. Зверніть увагу, що під час компіляції імена функцій та вхідні параметри хешуються та розтушовуються. Тому, щоб інший обліковий запис міг викликати функцію, він повинен мати доступ до заданого імені функції та аргументів, і нам потрібен ще один рівень, який забезпечує інтерфейс кодування в байт-код і з нього.
- ABI - це список визначень функцій та аргументів контракту у форматі JavaScript Object Notation (JSON). Визначення функцій та вхідні аргументи хешуються в ABI. Він включається в дані транзакції та інтерпретується EVM на цільовому рахунку. ABI необхідний для того, щоб можна було вказати, яку

функцію в договорі викликати, а також отримати гарантію, що функція поверне дані в тому форматі, на який ви очікуєте.

Структура мови програмування Solidity має багато спільних рис з популярними мовами високого рівня, такими як C++. Python та JavaScript. Як об'єктно-орієнтована мова програмування, типи даних у Solidity подібні до інших мов ООП. Деякі з поширених типів даних, що підтримуються в ООП мовах, які ви можете знайти в Solidity, включаючи Boolean, Integer, String, Array, and Modifier [12].

Булевий тип даних (Boolean) оцінює умови і повертає "1" при перевірці істинності умови або "0" при виявленні помилкової умови. Тип даних Integer в Solidity слугує для того, щоб надавати цілим значенням знак або беззнаковість за допомогою Solidity. Тип даних Integer також забезпечує підтримку винятків під час виконання, а також ключових слів uint256 та uint8. Тип даних String в програмуванні Solidity для блокчейну Ethereum передбачає використання одинарних або подвійних лапок для класифікації даних як рядка. Modifier - ще один важливий тип даних в мові програмування Solidity за своєю роллю перед виконанням коду смарт-контракту. Модифікатор відповідає за перевірку раціональності певної умови перед виконанням коду. Синтаксис типів даних Array в Solidity подібний до синтаксису інших мов ООП, особливо з підтримкою як одновимірних, так і багатовимірних масивів. Ще однією суттєвою особливістю в конструкції Solidity є можливість відображення структур даних з хеш-значеннями, переліченнями та операторами повернення значень в конкретні місця зберігання [12].

Розглянемо основні особливості цієї мови програмування:

- Ключове слово pragma

Solidity програмування в Ethereum, очевидно, буде посилатися на прагматику версій і їх переваги. Прагми в Solidity в основному слугують вказівками компілятору щодо методів управління кодом. Тому важливо додавати version pragma на початку кожного рядка вихідного коду на мові програмування Solidity.

Прагма версії пропонує специфікації версії компілятора Solidity, яка підходить для даного коду. Прагма версій є важливим елементом для функціональності EVM та смарт-контрактів у Solidity. Вона допомагає запобігти несумісності коду з майбутніми версіями компілятора, які можуть містити значні зміни.

- Ключове слово contract

Ключове слово "Contract" є ще однією важливою вимогою для початку роботи з програмуванням в Solidity та його функціями. Ключове слово допомагає оголосити контракт, який полегшить інкапсуляцію коду.

- Оголошення змінних стану

Змінні стану є важливим класом змінних в програмуванні Solidity. Змінні стану в блокчейні Ethereum постійно зберігаються в сховищі смарт-контрактів. Ви можете оголосити змінну стану з іменем var1 як тип uint за допомогою рядка "uint public var1". Змінна стану var1 в цьому випадку буде беззнаковим 256-бітним цілим числом. Важливо, що оголошення змінних стану більше схоже на введення додаткових слотів в базі даних.

- Оголошення функцій

Ще однією важливою базовою практикою є оголошення функцій. Візьмемо наступний приклад програмного коду Solidity для оголошення функції:

function set(uint a, uint b) public

function get() public views return (uint)

У даному прикладі можна звернути увагу на функцію з іменем set. Вона є функцією з модифікатором доступу типу public і використовує в якості параметрів змінні 'a' та 'b' з типом даних uint. Приклад такого простого смарт-контракту буде зосереджений на оновленні значення змінних var1 і var2. При цьому, користувачі, які мають доступ до блокчейну Ethereum, могли б використовувати функцію "set" для зміни значень var1 та var2. Крім того, додавання значень змінних var1 та var2 могло б допомогти в обчисленні змінної суми. Також смарт-контракт може отримати та роздрукувати значення суми змінних стану, використовуючи функцію "get".

- Виконання коду

Існує два різних способи виконання програми Solidity, наприклад, через онлайн і офлайн режими. Розглянемо способи виконання Solidity-програми більш детально.

Перший вибір для роботи зі смарт-контрактами Solidity EVM відноситься до офлайн-режиму роботи. Однак перед тим, як спробувати виконати смарт-контракт Solidity в автономному режимі, важливо дотримуватися трьох важливих умов. Першою важливою умовою для роботи смарт-контракту Solidity в офлайн-режимі є завантаження та встановлення node.js. Також необхідно встановити Truffle і Ganache-CLI як обов'язкові умови для роботи смарт-контрактів на Solidity в офлайн-режимі.Нижче наведені важливі кроки, які необхідно виконати для виконання коду смарт-контракту Solidity в автономному режимі:

1. Розробити проект Truffle і створити для нього надійну мережу розробки.
2. Розробити та розгорнути смарт-контракт для проекту.
3. Взаємодіяти зі смарт-контрактом через інформаційну панель Truffle.
4. Розробляти тести для оцінки основних функцій Solidity

Процес використання програмування Solidity в Ethereum також пропонує гнучкість для виконання коду в онлайн-середовищі. Remix IDE є найкращим вибором для всіх, хто зацікавлений в компіляції та запуску смарт-контрактів Solidity.

- Структури даних

Структури даних у Solidity виконують важливі функції. Однією з перших структур даних є Enum, яка допомагає відстежувати перелічені списки в мові програмування Solidity. Structs є ще одним прикладом популярних структур даних, які допомагають у визначенні ваших власних типів даних. Structs в основному можуть допомогти вам у моделюванні будь-якого типу даних відповідно до вимог з довільними атрибутами різних типів даних. Аналогічно, ви також можете зіткнутися з масивами (Array) як з одним із важливих елементів серед структур даних.

Mappings в Solidity допомагають зберігати пари "ключ-значення" і є невід'ємною частиною у функціоналі мови програмування. Структура даних "Mappings" в Solidity працює більше як хеш-таблиця або асоціативний масив, пов'язаний з іншими функціями.

- Видимість функцій

Видимість функцій відіграє вирішальну роль у визначенні обсягу функцій смарт-контракту в мові програмування Solidity. Наприклад, публічна видимість функції дозволяє легко викликати функцію через облікові записи за межами смарт-контракту. Також, внутрішня видимість корисна для створення функцій, які можуть бути використані тільки в рамках смарт-контрактів.

Доцільність програмування на Solidity для блокчейну Ethereum багато в чому залежить від переваг, які вона надає. Крім базового функціоналу, Solidity пропонує безліч цікавих можливостей, які підтверджують, що це кращий варіант, ніж багато мов програмування на основі Ethereum. Першою перевагою програмування на Solidity є підтримка складних типів даних і змінних-членів поряд з основними типами даних.

Крім того, мова програмування Solidity також пропонує двійковий інтерфейс додатків (Application Binary Interface або ABI) для забезпечення безпеки типів даних. ABI може допомогти у швидкому виявленні помилок, коли компілятор розпізнає будь-яку невідповідність типів даних для конкретних змінних. Ще одна популярна особливість програмування Solidity в Ethereum вказує на ефект "Специфікації природної мови". Solidity використовує специфікацію природної мови для перетворення специфікацій, орієнтованих на користувача, в мову, зрозумілу машинам [12]. Серед інших важливих переваг використання Solidity для створення децентралізованих додатків і смарт-контрактів можна виділити наступні:

- Легкий доступ до об'єктно-орієнтованих атрибутів в смарт-контрактах, таких як багаторівневе успадкування ознак.

- Solidity підтримує кілька варіантів допоміжних ролей, використовуючи двійковий інтерфейс додатків (Application Binary Interface або ABI).
- Більш простий синтаксис допомагає початківцям у вивченні основ смарт-контрактів та розробки блокчейну.
- Розробка смарт-контрактів на мові програмування Solidity допоможе забезпечити безпечні та надійні процеси для різних платформ, що мають справу з угодами між двома сторонами.

2.4. Архітектурні особливості програмного забезпечення у вигляді децентралізованих вебдодатків

DApp (Decentralized Applications) - це безсерверний додаток, який працює на стеку Ethereum і взаємодіє з кінцевим користувачем через HTML/JavaScript інтерфейс, який може здійснювати виклики до бекенд-стеку. Як правило, мобільний або вебдодаток має внутрішню частину, що працює на централізованих виділених серверах, однак, DApp має внутрішній код, що працює в децентралізованій одноранговій мережі. Структура DApp показана на Рисунку 9.

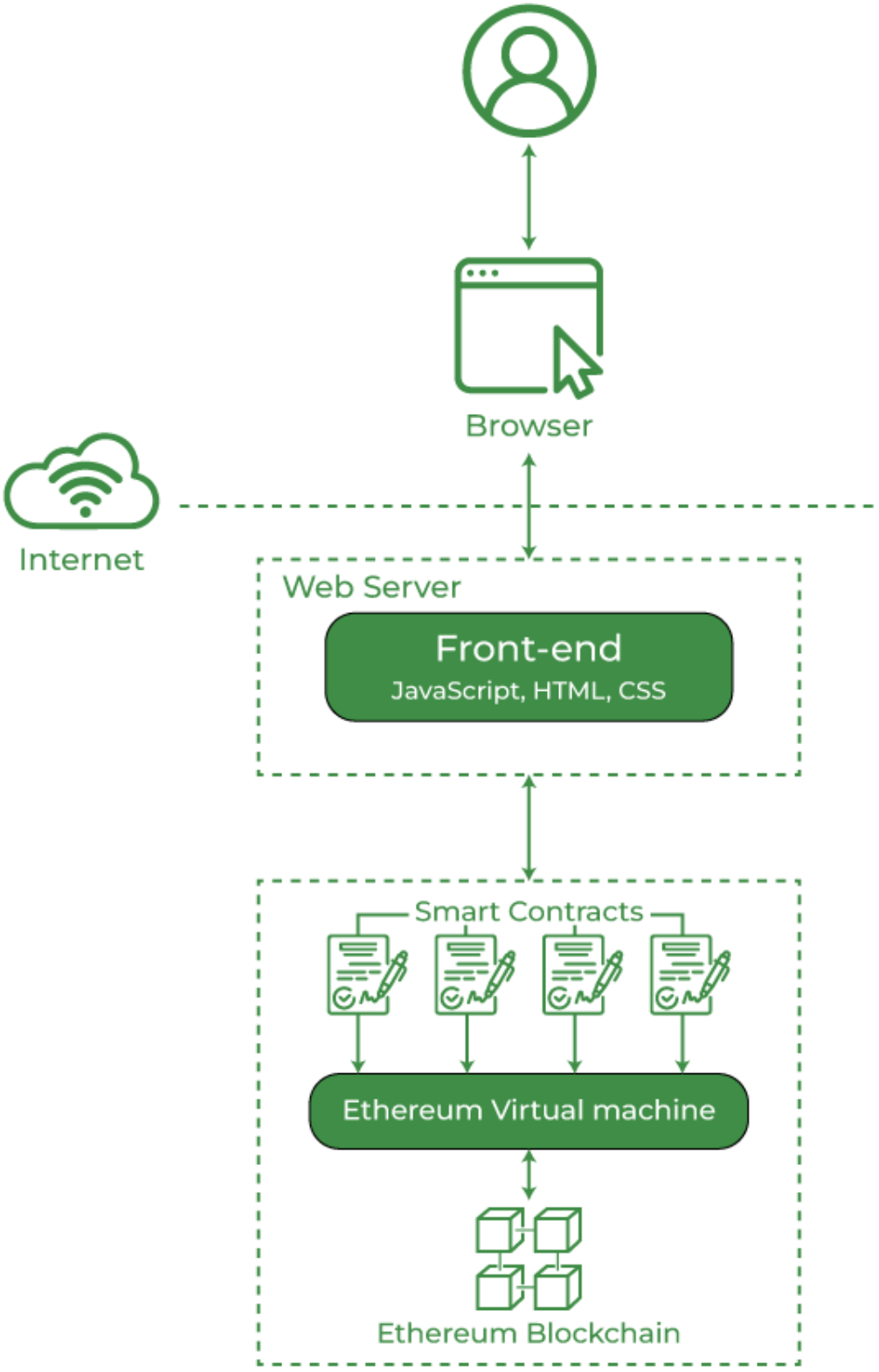

Рисунок 9 - Структура DApp

Нижче наведені основні компоненти архітектури DApp:

- Блокчейн Ethereum.

- Смарт-контракти.
- Віртуальна машина Ethereum (EVM).
- Front-end

Ethereum - це децентралізована блокчейн-платформа з відкритим вихідним кодом, яка створює однорангову мережу, що безпечно виконує та перевіряє код додатків, які називаються смарт-контрактами. Децентралізовані додатки мають свій внутрішній код (смарт-контракти), що працює в децентралізованій мережі, а не на централізованому сервері. DApps використовують блокчейн Ethereum для зберігання даних [11].

DApps використовують смарт-контракти для визначення змін стану, що відбуваються в блокчейні. Смарт-контракт - це набір коду і даних, який знаходиться за певною адресою в блокчейні Ethereum і працює на блокчейні Ethereum. Вони написані на мовах високого рівня, таких як Solidity і Vyper.

Віртуальна машина Ethereum - це глобальний віртуальний комп'ютер, який виконує логіку, визначену в смарт-контрактах, і обробляє зміни стану, які відбуваються в цій мережі Ethereum [11].

Front-end - це частина DApps, яку користувачі можуть бачити і з якою вони можуть взаємодіяти, наприклад, графічний інтерфейс користувача (GUI), але інтерфейс також взаємодіє з логікою програми, визначеною в смарт-контрактах.

Щоб викликати функції, фронт-енд повинен взаємодіяти зі смарт-контрактами. Але Ethereum - це децентралізована мережа. Кожен вузол в мережі Ethereum зберігає копію всіх станів на віртуальній машині Ethereum, включаючи дані і код, пов'язані з кожним смарт-контрактом.

Для взаємодії з даними і кодом в блокчейні необхідно взаємодіяти з одним з цих вузлів. Це пов'язано з тим, що будь-який вузол може транслювати запит на виконання транзакції на EVM. Тоді робота майнера полягає в тому, щоб виконати транзакцію і поширити отриману в результаті зміну стану в мережі. Транзакцію можна транслювати двома способами:

- Налаштувати вузол, на якому працює програмне забезпечення блокчейну Ethereum.

- Використовувати вузли, що надаються сторонніми сервісами.

Провайдер Ethereum (тобто ноди) реалізує специфікацію JSON-RPC. Це гарантує, що існує єдиний набір методів, через які додатки хочуть взаємодіяти з блокчейном. Підключившись до блокчейну через провайдера, можна прочитати стан, що зберігається в блокчейні [11]. Але якщо хтось хоче записати стан перед тим, як відправити транзакцію в блокчейн, потрібно зробити ще одну річ, а саме "підписати" транзакцію за допомогою приватного ключа. Кожного разу, коли додатку потрібно, щоб користувач підписав транзакцію, він звертається до Metamask. Тут Metamask грає роль підписанта. Роль Metamask в архітектурі DApp (Рисунок 10):

- Metamask - це плагін для браузера, який служить в якості гаманця Ethereum. Ми знаємо, що блокчейн Ethereum - це мережа, і нам потрібно спеціальне розширення для браузера, щоб підключити цю мережу [13].
- Metamask дозволить нам з'єднати блокчейн з нашим особистим кабінетом і взаємодіяти зі смарт-контрактом.
- Закриті ключі користувача зберігаються Metamask в браузері, і коли користувач хоче записати в стан, йому потрібно "підписати" транзакцію за допомогою закритого ключа.
- Metamask виступає і як провайдер, і як підписувач, оскільки Metamask забезпечує з'єднання з блокчейном (як "провайдер"), і оскільки він потрібен для підписання транзакцій, тому він також є підписувачем [13].

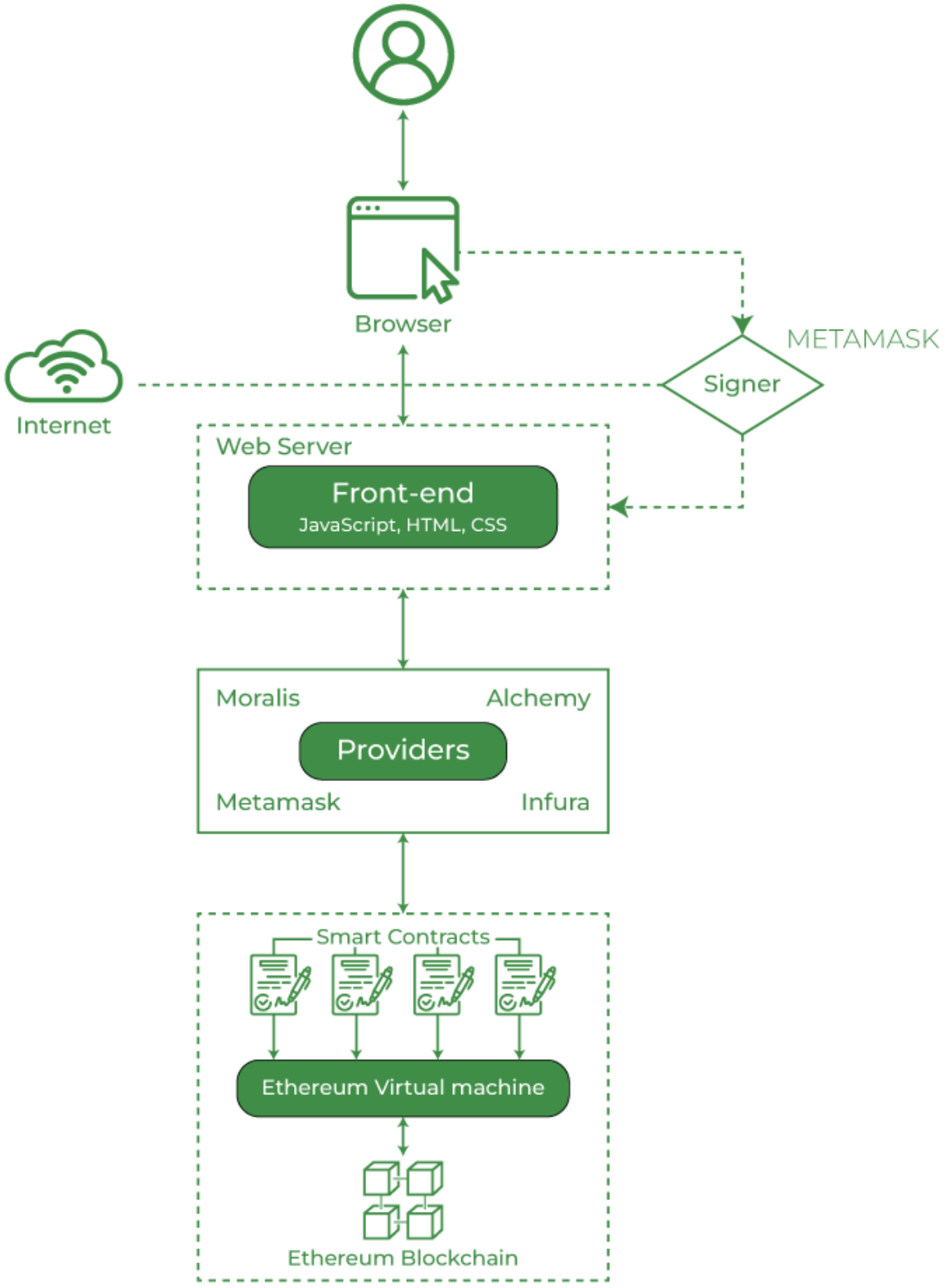

Рисунок 10 - Роль Metamask в архітектурі DApp

2.5. Опис фреймворку React для створення графічних інтерфейсів

ReactJS - це бібліотека JavaScript-компонентів для створення користувацьких інтерфейсів для вебсайтів і додатків. Розроблена Джорданом Воком у 2011 році, вона створює новий прецедент для розробки швидких та динамічних вебсайтів з використанням JavaScript. Вона надає програмістам ресурси для розробки багаторазових і легко інтегрованих компонентів інтерфейсу користувача, які скорочують час розробки і забезпечують адаптивний користувацький інтерфейс для вебсайтів, веб та мобільних додатків [14].

Інтерфейс користувача або UI - це все, що користувач використовує для взаємодії з вебсайтом, наприклад, кнопки, посилання, вкладки меню або рядки пошуку. React пропонує велику колекцію таких компонентів інтерфейсу користувача, які використовують фронтенд-розробники для створення персоналізованих інтерактивних елементів. Багато користувачів описують React як фреймворк, а не бібліотеку. Вони не помиляються в цьому. Адже React - це більше, ніж просто якісь заздалегідь написані фрагменти коду і функції. Сьогодні за допомогою ReactJS та React Native можна створювати цілі додатки для веббраузерів та смартфонів. Він має надійну екосистему і є надзвичайно гнучким, поставляється з власним набором супутніх бібліотек і фреймворків для реалізації складних функцій. За допомогою таких функцій, як Virtual DOM, JSX, вражаюче управління станами та незалежні один від одного компоненти багаторазового використання, користувачі React можуть значно скоротити час розробки. Можливо створювати чисті та керовані коди завдяки суворим правилам структури React та односпрямованому потоку даних. Тестувати і підтримувати ці вебдодатки і вебсайти також не складе труднощів.

Основні переваги React для вибору його для створення вебдодатків:

React легше вивчити в порівнянні з Angular. React - це елементарна та легка бібліотека, яка працює лише з рівнем перегляду вебсторінки. Вона має легку криву навчання, що робить її більш легкою для початку, ніж її аналог Angular при використанні JavaScript. Якщо у вас є функціональне розуміння HTML-CSS і

базове розуміння концепцій програмування, ви можете швидко почати працювати з React. Якщо ви переходите з іншого фреймворку або бібліотеки, може знадобитися деякий час, щоб звикнути до інноваційних методів React. Крім того, немає жодної концепції або утиліти React, яка була б занадто складною для використання. В Інтернеті є велика кількість документації, яка допоможе вам з усім, що пов'язано з ReactJS.

React має велику спільноту розробників. Серед топ-5 трендових репозиторіїв на GitHub, React - це бібліотека з відкритим вихідним кодом, створена Джорданом Уоком (Jordan Walke). Вона зібрала величезну кількість прихильників серед розробників JavaScript, які регулярно розробляють нові рішення та інструменти. В офіційну бібліотеку включено багато додатків, розроблених користувачами. Ви можете отримати доступ до великої спільноти експертів для вирішення будь-яких завдань [14].

React пропонує багаторазові компоненти. Компоненти в ReactJS - це незалежні, багаторазово використовувані частини коду. Ви можете надати вхідні дані, якщо це необхідно (так звані пропси), і компонент поверне відрендерений HTML-елемент. Ви можете використовувати їх як первинну функцію JavaScript або компонент класу, клас JavaScript, який використовує метод рендерингу. Кожен React-компонент, який ви розробили, може бути повторно використаний в інших частинах програми, або ви можете створити обгорткові компоненти, які забезпечують структуру і можливість повторного використання. Ви можете продовжувати створювати обгорткові компоненти, використовуючи вже існуючі, і в кінцевому підсумку у вас буде кореневий компонент, який і є вашим додатком. Як правило, оновлення системи за допомогою багаторазових компонентів часто призводить до багатьох ускладнень, оскільки кожна зміна може вплинути на роботу інших компонентів. Але в React управління оновленнями є простим для розробників, оскільки всі компоненти є ізольованими, і зміни в одному з них не впливають на інші.

Віртуальний DOM. Document Object Model або DOM - це інтерфейс, який представляє HTML та XML код у вигляді дерев. веббраузер створює DOM-подібну

модель для відображення вихідних даних, розглядаючи кожен об'єкт HTML-коду як вузол у дереві DOM. Кожного разу, коли відбувається зміна в HTML-коді, або через взаємодію з користувачем, або через оновлення значень, DOM-дерево має бути відрендерено знову, що призводить до значних витрат часу та енергії. ReactJS вирішує цю проблему, використовуючи віртуальні DOM. React просто створює копію DOM, підтримуючи своєрідну кеш-пам'ять. Кожного разу, коли вносяться зміни, він перевіряє віртуальний DOM і точно визначає, які вузли дерева і компоненти потрібно оновити. З невеликою зміною в DOM, дерево може бути оновлене швидко і ефективно. Це економить розробникам купу часу і робить додаток надзвичайно швидким [14].

JSX підвищує продуктивність та ефективність ReactJS. JSX або JavaScript XML - це синтаксичне розширення для JavaScript. Facebook розробив його, щоб розширити функціональність структур HTML в JavaScript. З JSX немає необхідності в окремих HTML і JS кодах. Ви можете використовувати декларативний синтаксис HTML безпосередньо в JavaScript-коді за допомогою ReactJS. Браузери декодують HTML-документи для відображення інтерфейсу користувача. Вони роблять це, створюючи DOM-дерева, а JavaScript дозволяє нам модифікувати це DOM для створення інтерактивного інтерфейсу. JSX підвищує ефективність маніпулювання DOM-деревами в рази. Розробники можуть передавати HTML і ReactJS елементи в деревовидні структури браузера і писати чистий, керований код. JSX, разом з Virtual DOM, підвищує продуктивність і ефективність ReactJS-додатків. JSX також можна використовувати з іншими фреймворками та бібліотеками.

Ефективне налагодження та перевірка помилок з односпрямованим потоком даних. У React інформація, успадкована від батьківського компонента, називається пропсами. Пропси - це незмінні об'єкти, значення яких не можуть бути змінені дочірніми компонентами завдяки односпрямованому потоку даних в React. Спрямоване вниз зв'язування робить код стабільним і послідовним, оскільки будь-які зміни в дочірніх компонентах не впливають на їхніх братів і сестер або батьківські компоненти. Щоб змінити об'єкт, потрібно лише оновити його стан.

ReactJS автоматично змінить дійсні дані, щоб зберегти узгодженість. Налагодження та перевірка помилок набагато ефективніші в React завдяки однонаправленому зв'язуванню даних, що забезпечує вищий контроль над доступом до даних компонентів [14].

ВИСНОВКИ ДО ДРУГОГО РОЗДІЛУ МД

В другому розділі дисертації було проаналізовано принципи роботи технології Blockchain та визначено її переваги та недоліки для використання в створенні децентралізованого програмного забезпечення. Визначено особливості використання блокчейн мережі Ethereum, порівняно із мережею Bitcoin. Розглянуто мову програмування Solidity, як інструмент для створення смарт-контрактів. Крім того, досліджено архітектурні особливості програмного забезпечення у вигляді децентралізованих вебдодатків. Також описано бібліотеку React для створення графічних інтерфейсів користувача.

3. СТРУКТУРА ТА ОПИС РОБОТИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ КОНТРОЛЮ ТА ВИКОНАННЯ ТРАНЗАКЦІЙ З КРИПТОВАЛЮТОЮ

3.1. Вибір інструментарію для реалізації

Основними інструментами для розробки децентралізованого веборієнтоваго засобу для контролю та виконання транзакцій з криптовалютою є:

- WebStorm - це інтегроване середовище розробки для JavaScript і суміжних технологій. Як і інші IDE JetBrains, воно робить процес розробки більш приємним, автоматизуючи рутинну роботу і допомагаючи з легкістю справлятися зі складними завданнями.
- Node.js - це крос платформне середовище виконання з відкритим вихідним кодом для розробки серверних та мережевих додатків. Додатки Node.js написані на мові JavaScript і можуть бути запущені в середовищі виконання Node.js на OS X, Microsoft Windows і Linux.
- React.js - це бібліотека JavaScript-компонентів для створення користувацьких інтерфейсів для вебсайтів і додатків. Вона надає програмістам ресурси для розробки багаторазових і легко інтегрованих компонентів інтерфейсу користувача, які скорочують час розробки і забезпечують адаптивний користувацький інтерфейс для вебсайтів, веб та мобільних додатків.
- Tailwind CSS - це CSS-фреймворк для швидкого створення користувацьких інтерфейсів. Це низькорівневий фреймворк CSS, який легко налаштовується і дає вам всі будівельні блоки, необхідні для створення індивідуальних дизайнів.
- Solidity - це об'єктно-орієнтована мова високого рівня для реалізації смарт-контрактів. Смарт-контракти - це програми, які керують поведінкою облікових записів всередині держави Ethereum.
- Metamask - це плагін для браузера, який служить в якості гаманця Ethereum. Ми знаємо, що блокчейн Ethereum - це мережа, і нам потрібно спеціальне розширення для браузера, щоб підключити цю мережу.

- Vite - це інструмент збірки, який значно покращує досвід розробки інтерфейсів. Ви можете використовувати Vite для налаштування середовища розробки для таких фреймворків, як Vue та React, і навіть для звичайного JavaScript-додатку з сервером розробки та гарячим перезавантаженням.
- Hardhat - це середовище, яке розробники використовують для тестування, компіляції, розгортання та налагодження додатків на основі блокчейну Ethereum. Таким чином, воно допомагає програмістам та розробникам керувати багатьма завданнями, які притаманні розробці dApps та смарт-контрактів. Поряд з наданням розробникам належних інструментів для управління цим процесом, Hardhat також допомагає автоматизувати деякі з цих кроків і надає розробникам нові корисні функції.

3.2. Принцип роботи

Головні структурні елементи додатку можна побачити на Рисунку 11.

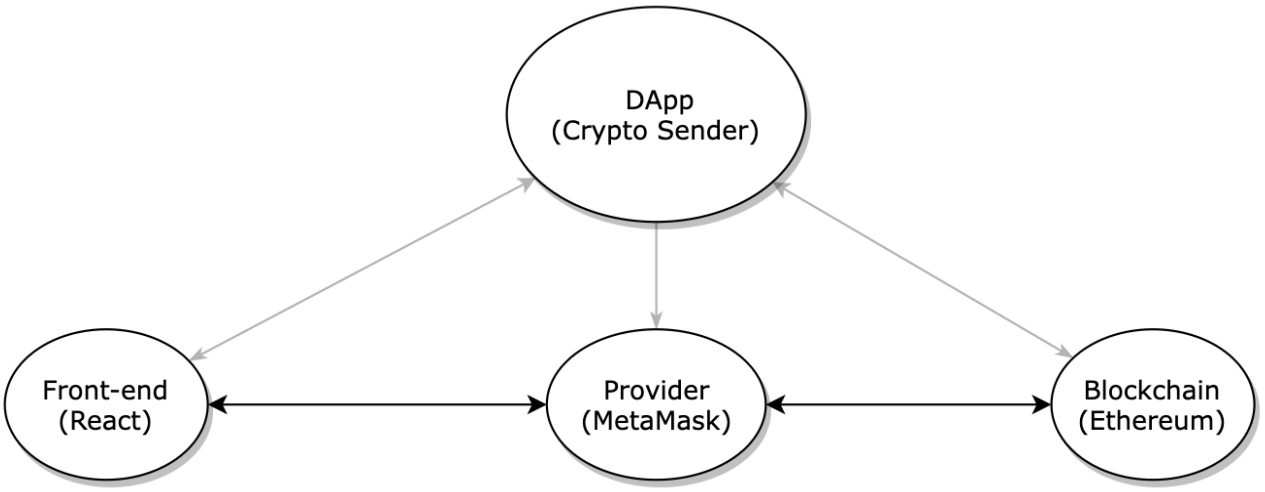

Рисунок 11 - Головні структурні елементи додатку та взаємодія між ними

Першим елементом є інтерфейсна частина децентралізованого додатку, яка виконана за допомогою React та супутніх технологій, таких як HTML, CSS,

JavaScript. За допомогою цих технологій була побудована форма для відправки криптовалюти Ether із одного гаманця на інший, інтерфейс для авторизації за допомогою свого криптогаманця та частина для відображення останніх виконаних транзакцій пов’язаних з поточним користувачем.

Другим елементом є розширення для браузера MetaMask, що надає інструменти для з’єднання додатку із блокчейном Ethereum, при цьому воно виступає також у ролі підписувача для транзакцій, тобто перевіряє їх валідність та дозволяє проведення, при умові, що не виникло ніяких помилок.

Останнім та головним елементом додатку є блокчейн Ethereum. Він дозволяє виконувати логіку, визначену в смарт-контрактах написаних на мові програмування Solidity, і обробляти зміни стану, які відбуваються в цій мережі, після їх виконання. Сам блокчейн виступає як децентралізована база даних, що виконує та зберігає інформацію про виконані транзакції.

Алгоритм роботи додатку (Рисунок 12):

1. Користувач надсилає запит на авторизацію за допомогою свого криптогаманця MetaMask.
2. Якщо запит успішний то, сервіс MetaMask надає доступ додатку до блокчейн мережі Ethereum та авторизує користувача за допомогою його криптогаманця.
3. Якщо запит помилковий, то повернення до кроку 1.
4. Користувач заповнює форму та надсилає запит на виконання транзакції до MetaMask.
5. MetaMask перевіряє валідність транзакції.
6. Якщо транзакція пройшла перевірку, то MetaMask дозволяє її виконання та передає керування до віртуальної машини Ethereum, перехід до кроку 8.
7. Якщо транзакція із якоїсь причини не пройшла перевірку, то MetaMask забороняє її виконання, повідовляючи про це користувача, повернення до кроку 4.

8. Віртуальна машина Ethereum виконує смарт-контракт, який відповідає за проведення та збереження транзакцій в додатку

9. MetaMask повідомляє додаток про успішне проведення транзакції

10. Користувач отримує дані про проведення транзакції

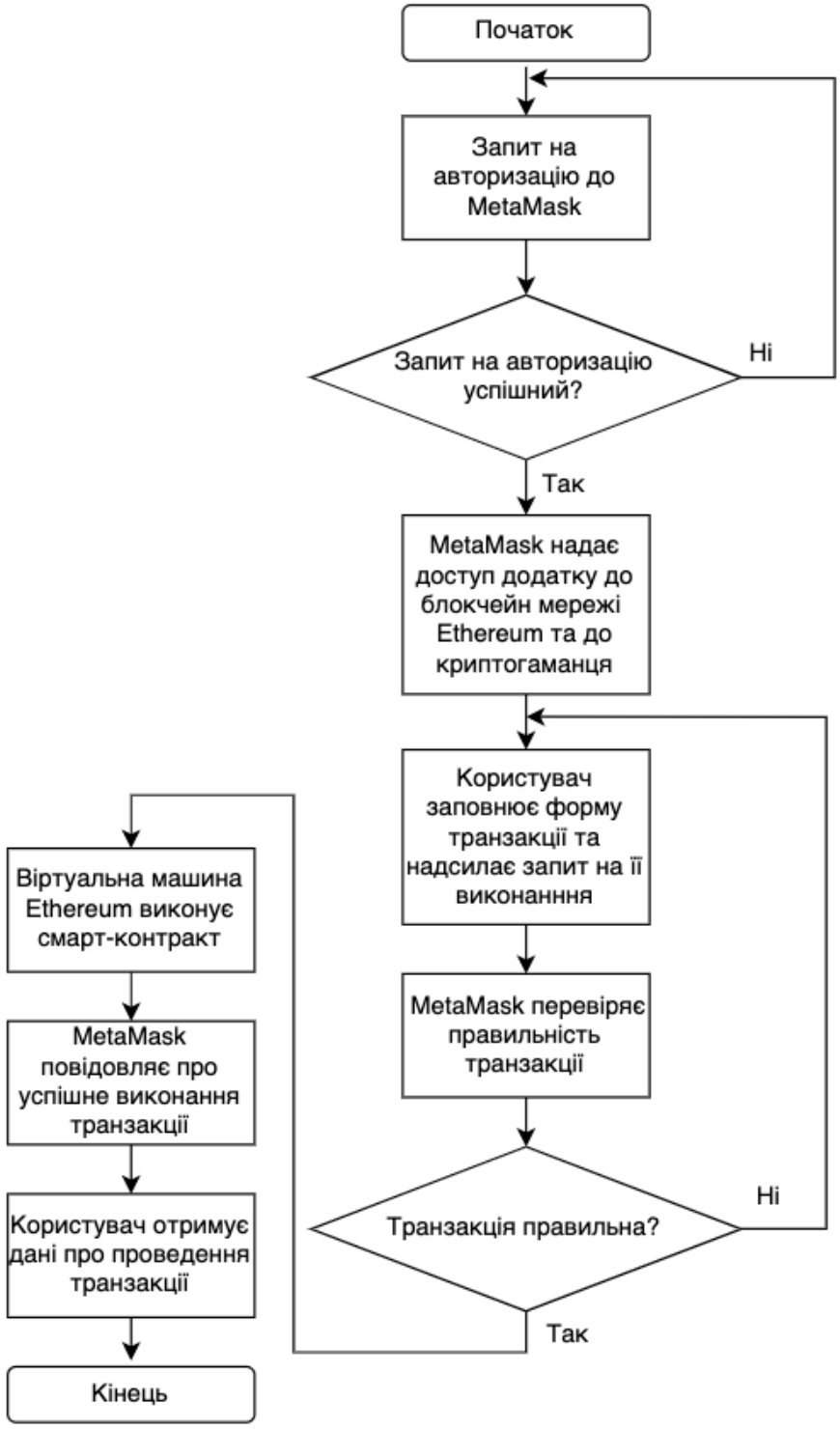

Рисунок 12 - Алгоритм роботи додатку

3.3. Опис архітектури вебдодатку

Архітектура коду проєкту децентралізованого вебдодатку для контролю та виконання транзакцій з криптовалютою поділена на 2 під-проєкти, код яких розміщений в різних репозиторіях (Рисунок 13):

- **client** - містить інтерфейсну частину додатку створену за допомогою React та супутніх технологій та логіку взаємодії інтерфейсної частини із блокчейн мережею Ethereum та провайдером MetaMask.
- **smart_contract** - містить смарт контракт, який відповідає за логіку проведення транзакцій та запускається на віртуальній машині Ethereum, також містить код для розгортання даного смарт-контракту в блокчейн мережі Ethereum.

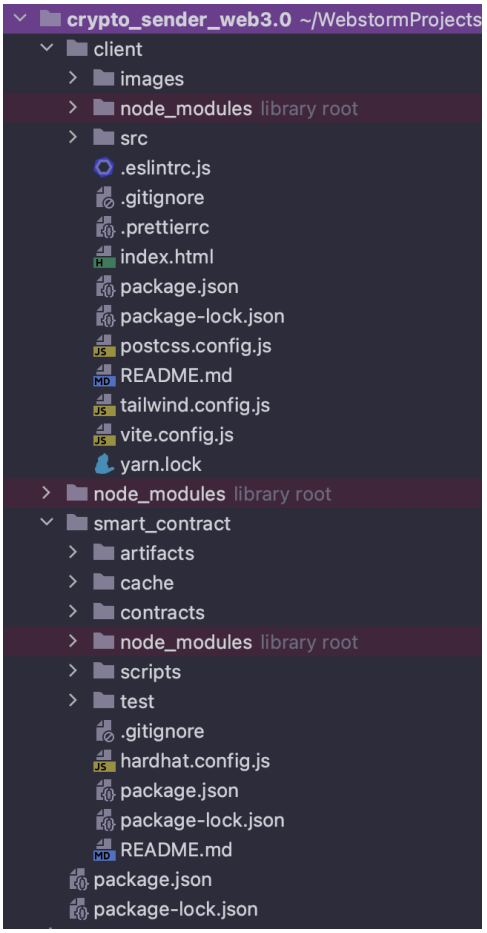

Рисунок 13 - Архітектура коду проєкту

Розглянемо детальніше вміст директорії **client**. На першому рівні вкладеності, вона містить файли конфігурацій різних технологій та базові файли:

- **.eslint.js** - конфігурація ESLint для перевірки правильності написання коду
- **.prettierrc** - конфігурація Prittier для форматування коду
- **postcss.config.js** - конфігурація PostCSS для трансформування коду стилів
- **Tailwind.config.js** - конфігурація бібліотеки Tailwind для полегшення написання стилів
- **Vite.config.js** - допоміжний файл Vite - утиліти для покращення розробки та збірки проєкту
- **package.json** - це JSON-файл, який існує в корені проекту Javascript/Node та містить метадані, що відносяться до проекту, і використовується для управління залежностями проекту, скриптами, версією і багатьом іншим.
- **index.html -** основа односторінкового додатку куди вбудовується весь інший код

Основний вихідний код міститься в директорії **src.** Вона містить такі файли та директорії:

- **App.jsx** - React компонент верхнього рівня
- **main.js** - точка входу в додаток
- **index.css** - файл стилів
- Директорія **components** - містить інші компоненти React, такі як **Loader.jsx, Sender.jsx, Transactions.jsx**, що вбудовуються в **App.jsx**
- Директорія **context** - містить один файл **TransactionContext.jsx**, що містить логіку взаємодії додатку із блокчейн мережею Ethereum та MetaMask
- Директорія **utils** - містить допоміжні функції та константи для використання по всьому додатку.

Кожен React компонент відповідає за логіку роботи та відображення частини інтерфейсу:

- **Sender.jsx** - інтерфейс форми для відправки транзакції та авторизації за допомогою криптогаманця
- **Loader.jsx** - відображення компонента завантаження

- **Transactions.jsx** - відображення карток із транзакціями пов’язаними з поточним криптогаманцем

Файл **TransactionContext.jsx** відіграє важливу роль в функціонуванні цього додатку. Він відповідає за логіку взаємодії додатку із блокчейн мережею Ethereum та MetaMask. За допомогою такої концепції як React.Context він інкапсулює всю потрібну логіку для взаємодії із блокчейном в одному місці, та дозволяє використовувати отримані дані по всьому дереву компонентів додатку. В цьому місці інкапсульована логіка для створення екземпляру контракту Ethereum, отримання всіх транзакцій, перевірки чи підключений криптогаманець, перевірки чи існують виконані транзакції, підключення гаманця, виконання відправки транзакції.

Розглянемо детальніше вміст директорії **smart_contract**. Як і **client,** це окремий під-проєкт, що також містить свій **package.json,** та відповідає за логіку роботи і розгортання смарт-контракту для проведення та контролю транзакцій з криптовалютою Ether. Ві також містить деякі файли конфігурацій, наприклад **hardhat.config.js** - конфігурація утиліти для розгортання та тестування смарт-контрактів локально. Основними файлами тут є файли, що знаходяться за шляхами **contracts/Transactions.sol** та **scripts/deploy.js.**

Transactions.sol - це файл, що містить код смарт-контракту, що написаний на мові програмування Solidity, та містить логіку по обробці транзакцій в блокчейн мережі Ethereum. Основними функціями контракту є:

- **addToBlockchain** - додавання транакції до блокчейн мережі
- **getAllTransactions** - отримання всіх створених транзакцій із мережі
- **getTransactionsCount** - отримання кількості виконаних транзакцій
- Подія **Transfer**, що інформує про проведення транзакції

При цьому в контракті використовується структура **TransferStruct**, що містить в собі всі дані про транзакцію:

1. Адреса відправника
2. Адреса отримувача
3. Кількість криптовалюти для пересилання

4. Текстове повідомлення
5. Час виконання транзакції
6. Ключове слово

Файл **deploy.js** призначений для розгортання створеного смарт-контракту **Transactions.sol** в блокчейн мережу Ethereum за допомогою утиліти hardhat. Для цього необхідно виконати команду **npx hardhat run scripts/deploy.js --network goerli,** де **goerli -** це назва тестової мережі Ethereum, яка була обрана для тестування створеного додатку.

Інші директорії в цьому під-проєкті слугують для покращення відлагодження та розробки смарт-контрактів, також після розгортання смарт-контракту з’являться декілька допоміжних директорій, які містять інформацію про сам контракт, наприклад директорія **artifacts, cache.** За шляхом **smart_contract/artifacts/contracts/Transactions.sol/Transactions.json** знаходиться головний згенерований файл, що містить в собі ABI створеного смарт-контракту, який використовується в інтерфейсній частині додатку при створенні екземплярів контракту в файлі **TransactionContext.jsx.**

ВИСНОВКИ ДО ТРЕТЬОГО РОЗДІЛУ МД

Даний розділ дисертації присвячений опису програмної реалізації децентралізованого вебдодатку. Описано інструментарій, що був обраний для реалізації продукту, архітектуру вебдодатку, та принципи роботи основних модулів. Було використано JavaScript бібліотеку React для реалізації клієнтської частини та мову програмування Solidity для створення смарт-контракту.

4. ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

4.1. Тестування програмного забезпечення

Для тестування коректності роботи створеного децентралізованого веборієнтованого засобу для контролю та виконання транзакцій з криптовалютою у вигляді вебдодатку під назвою Crypto Sender, необхідно створити свій криптогаманець MetaMask у валюті Ether. Для цього необхідно встановити розширення для браузера (Рисунок 14) та створити за допомогою нього криптогаманенць, це доволі легкий процес із яким впорається кожен.

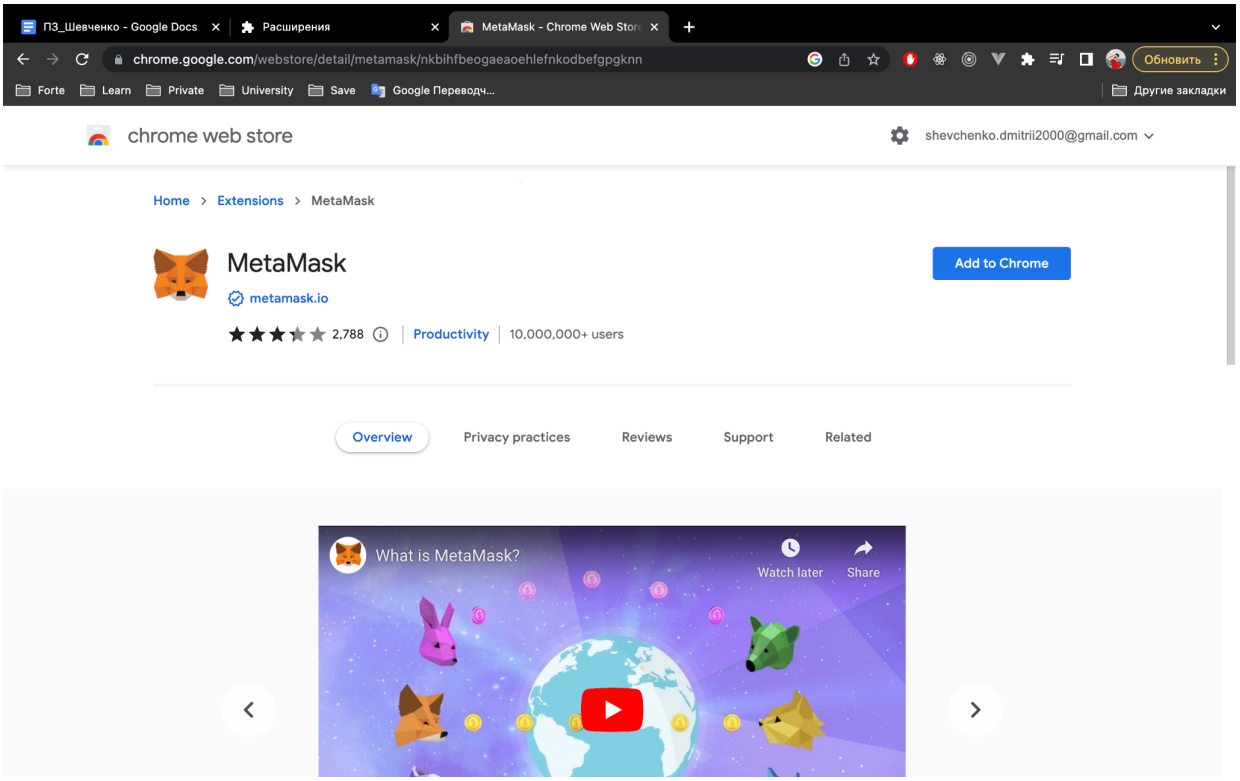

Рисунок 14 - Розширення для браузера MetaMask

Після успішного створення криптогаманця (Рисунок 15) необхідно в налаштуваннях дозволити показ тестових мереж (Рисунок 16) і потім вибрати необхідну для тестування мережу, наприклад Goerli. Такий вид мереж

призначений саме для перевірки коррексності роботи смарт-контрактів перед розгортанням їх в головній блокчейн мережі Ethereum.

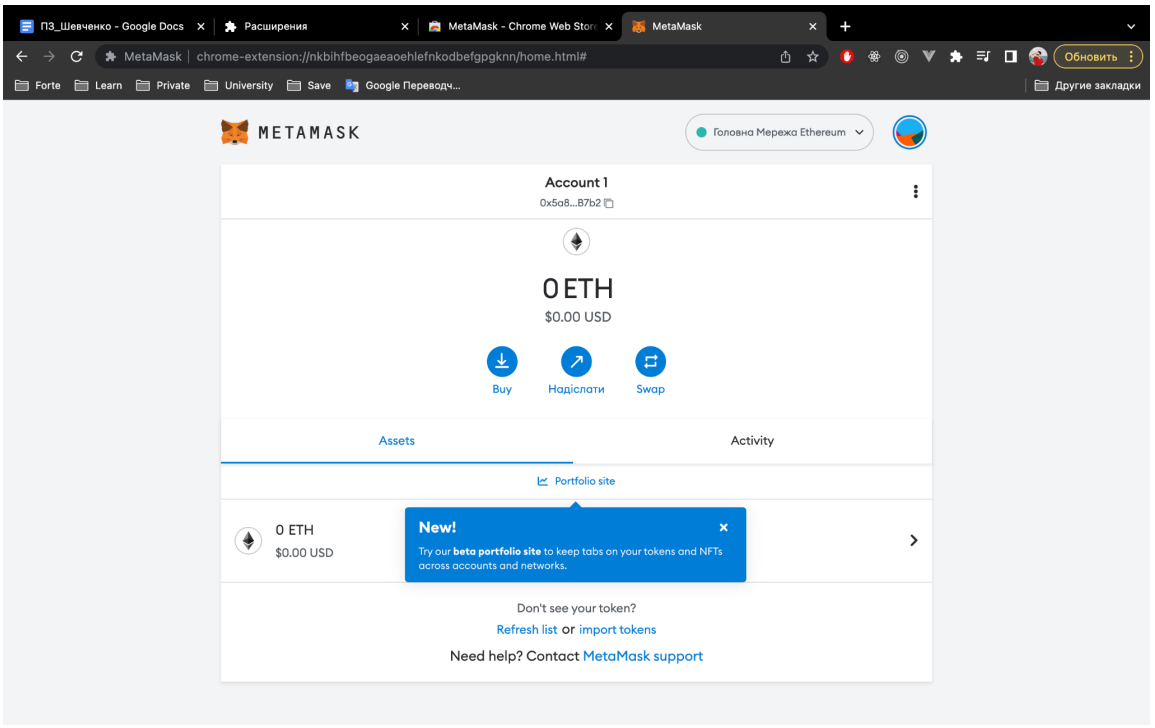

Рисунок 15 - криптогаманець MetaMask

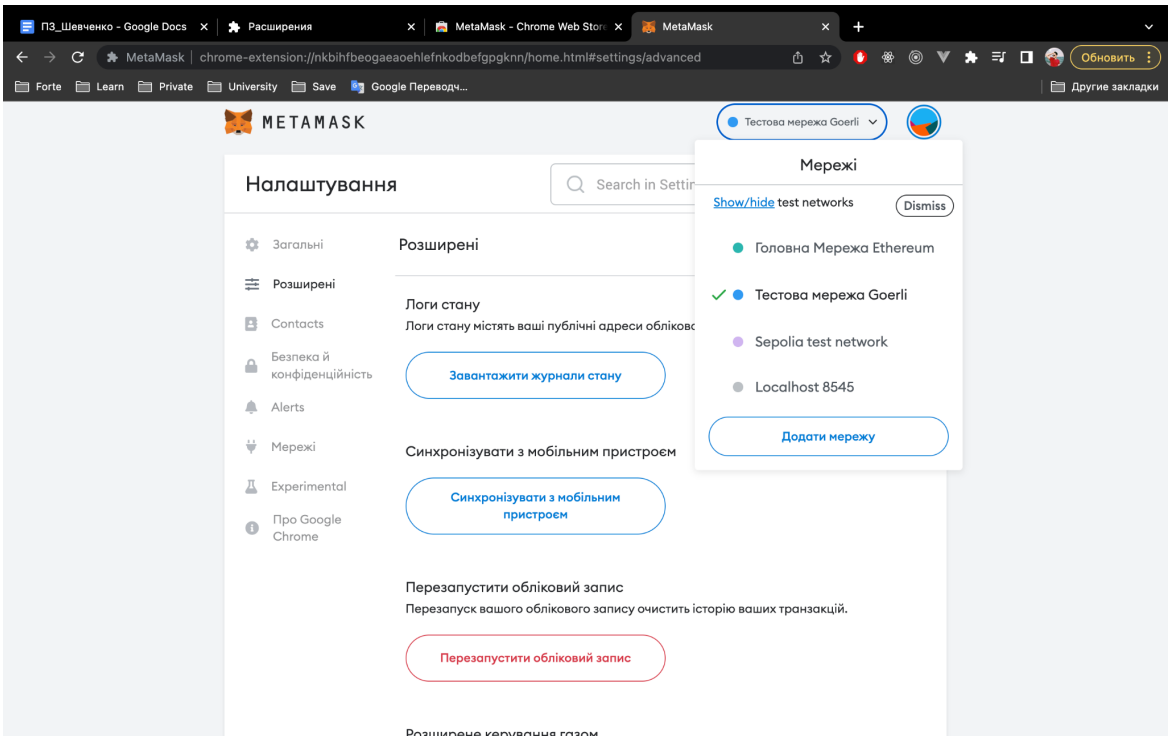

Рисунок 16 - вибір тестової мережі Goerli

Після цього необхідно отримати якусь невелику кількість криптовалюти Ether на створений гаманаць. Щоб це зробити, можна перейти на спеціалізований ресурс під назвою goerlifaucet.com та вставивши в поле адресу свого гаманця, протягом деякого часу отримати 0.2 ETH (Рисунок 17).

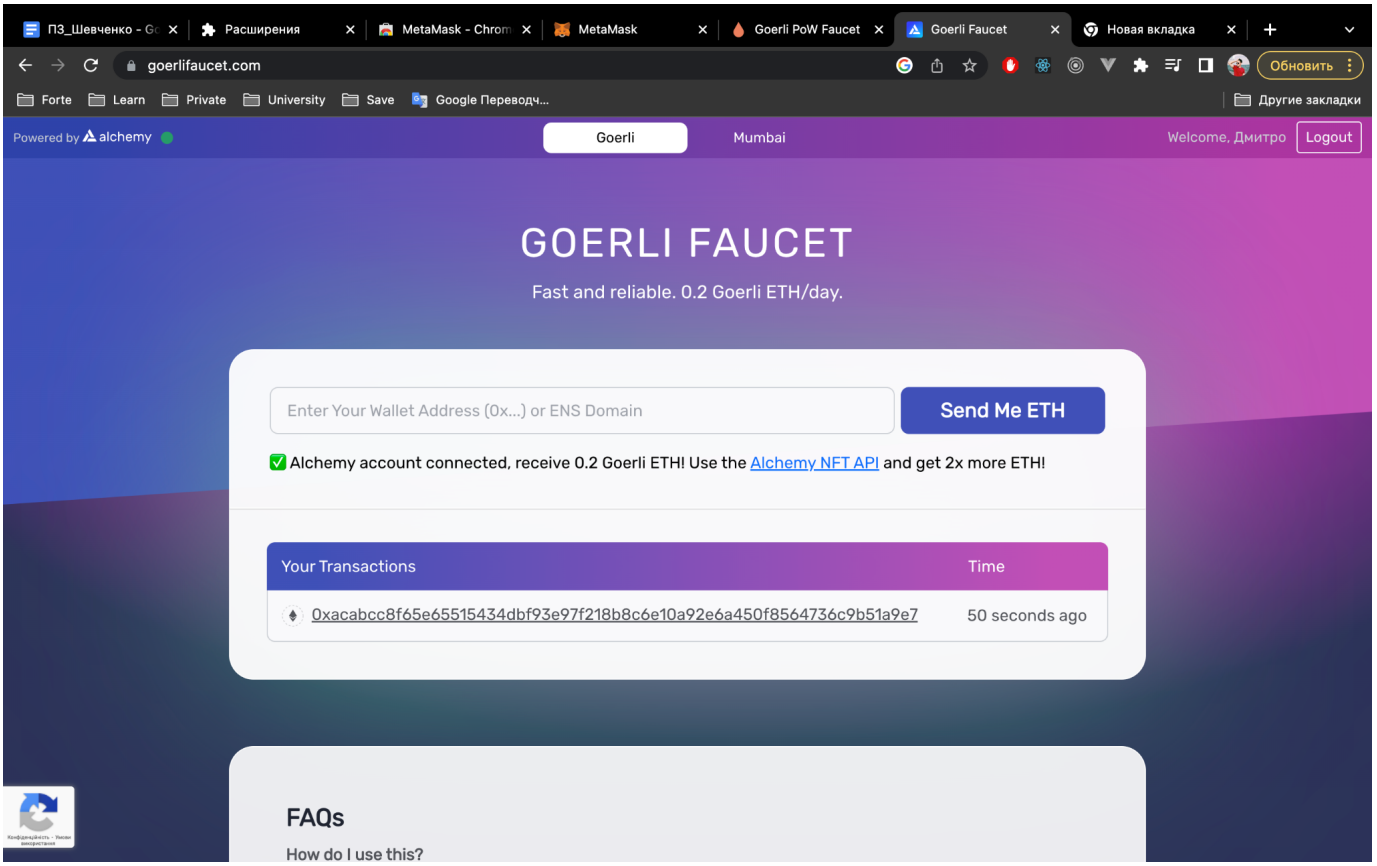

Рисунок 17 - отримання криптовалюти Ether для тестування

Після перевірки отримання криптовалюти на гаманець необхідно розгорнути створений смарт-контракт до тестової мережі Goelri за допомогою сервісу Alchemy (Рисунок 18).

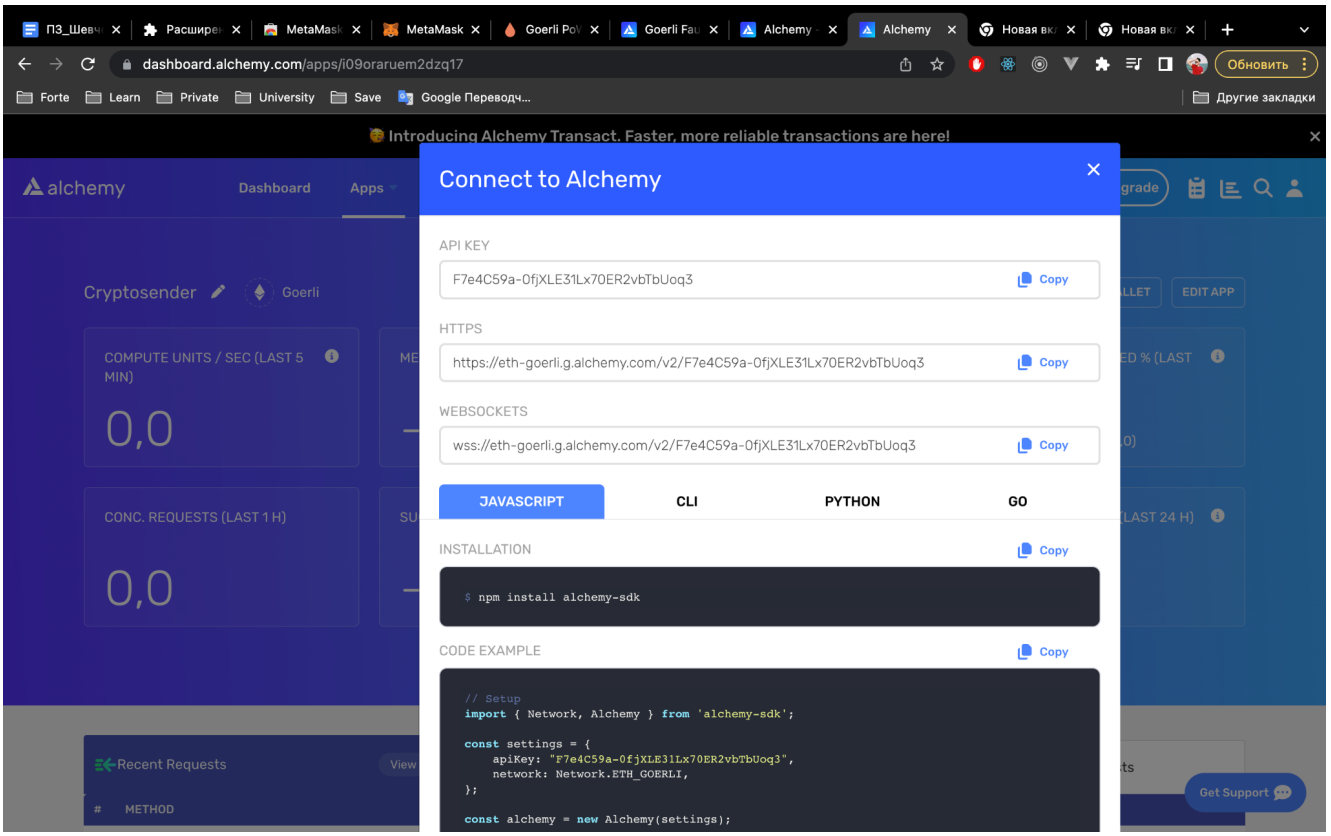

Рисунок 18 - Розгортання смарт-контракту до тестової мережі

Тепер можна запустити створений додаток та розпочати тестування (Рисунок 19). При відкритті сторінки додатку Crypto Sender можемо побачити 3 основні блоки:

- Зліва - блок авторизації за допомогою криптогаманця
- Справа - блок із формою для відправки транзакції
- Знизу - блок із останніми транзакціями з поточним криптогаманцем

Для початку роботи необхідно авторизуватися за допомогою існуючого криптогманця MetaMask. Для цього потрібно натиснути кнопку Connect Wallet (під'єднати гаманець). Після натискання з'явиться вікно розширення MetaMask із вибором гаманця та підтвердженням під'єднання (Рисунок 20).

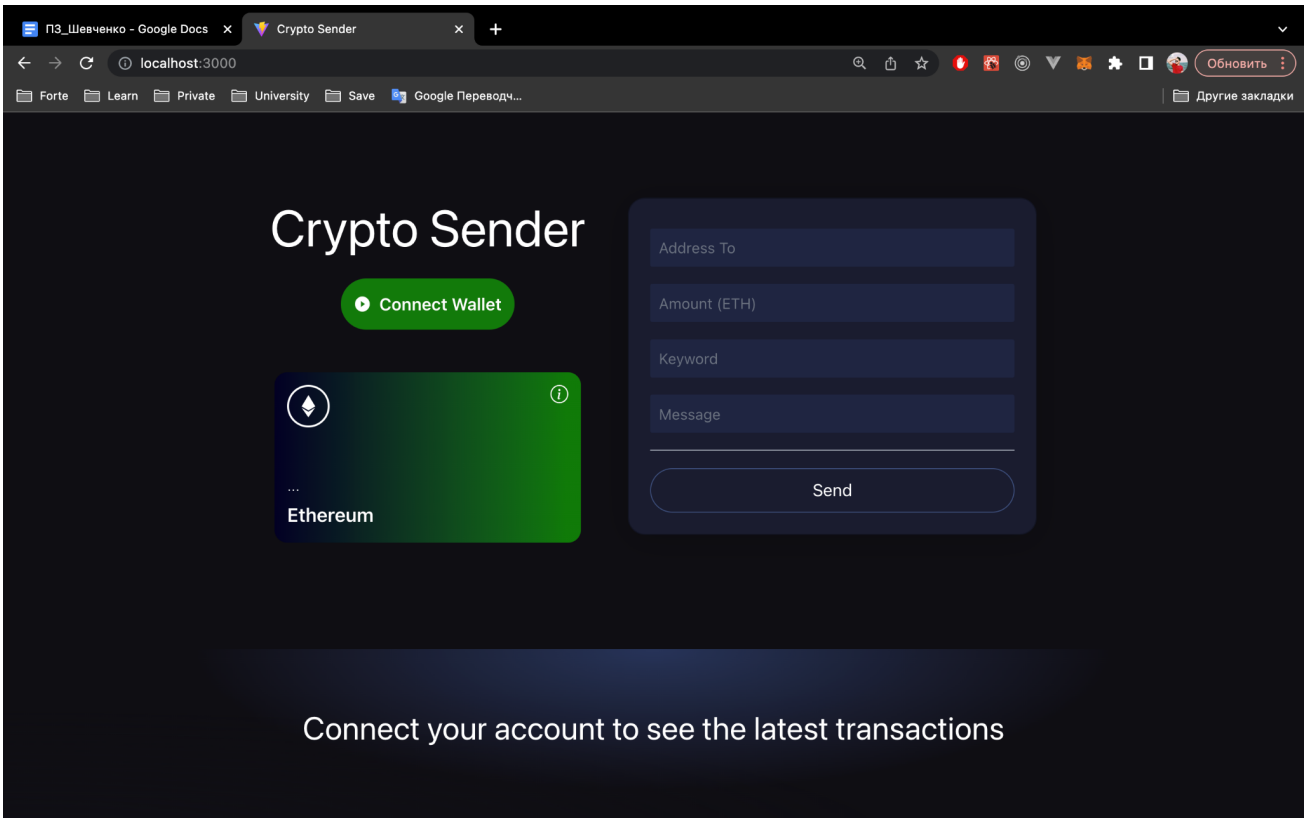

Рисунок 19 - Головна сторінка додатку Crypto Sender

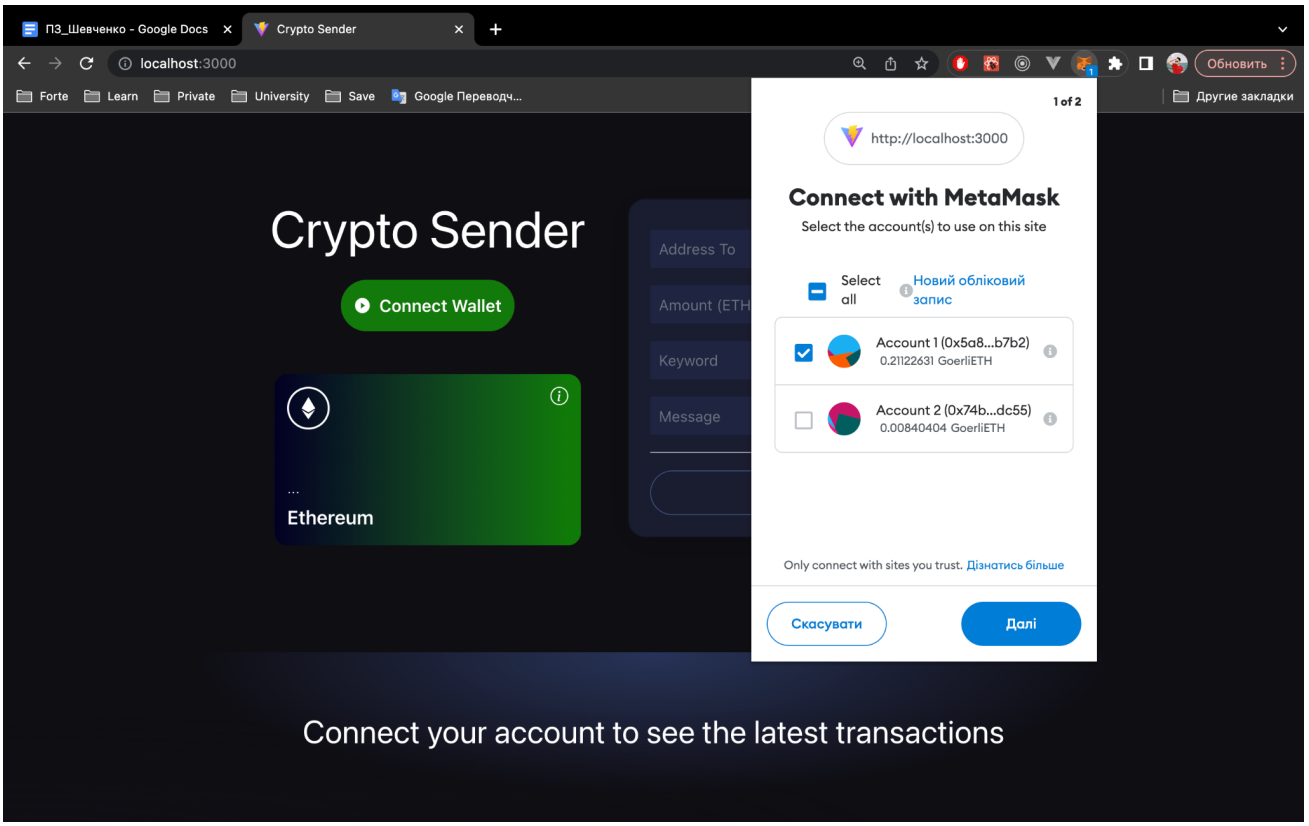

Рисунок 20 - авторизація за допомогою криптогаманця MetaMask

Після успішного під’єднання криптогаманця можна розпочати виконання транзакцій. Для того, щоб виконати підправку криптовалюти Ether необхідно заповнити форму, яка містить такі поля:

- Address to - адреса отримувача
- Amount - кількість криптовалюти для відправки в ETH
- Keyword - ключове слово для шифрування
- Message - коротке опціональне повідомлення

Коли форма буде заповнена, необхідно натиснути кнопку Send (відправити), щоб розпочати процес відправки. Потім з’явиться нове вікно MetaMask для підтвердження виконання транзакції із даними про неї. В даному випадку з поточного криптогаманця підправляється 0.05 ETH на інший створений завчасно гаманець із ключовим словом test та повідомленням it is testing (Рисунок 21).

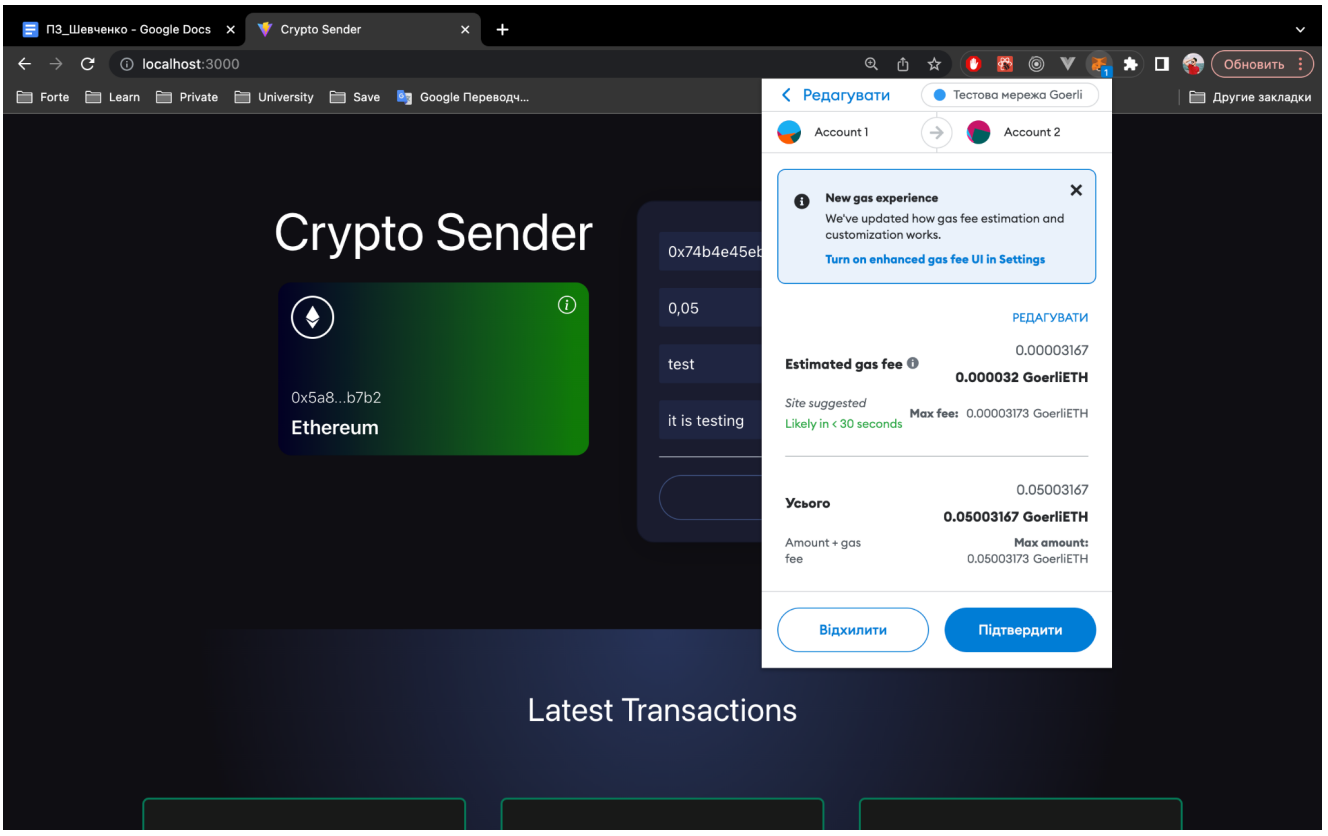

Рисунок 21 - підтвердження виконання транзакції

Через короткий час очікування, близько 5 секунд, обробка транзакції буде завершена і в блоці із останніми транзакціями з’явиться нова картка із щойно

виконаною операцією (Рисунок 22). Вона містить інформацію із адресами відправника та отримувача, кількість переведеної криптовалюти, коротке повідомлення та точний час виконання операції. При цьому адреси є гіперпосиланнями на які можна натиснути та перейти на спеціальний сервіс etherscan.io на якому можна додатково перевірити виконання кожної транзакції із даною адресою (Рисунок 23).

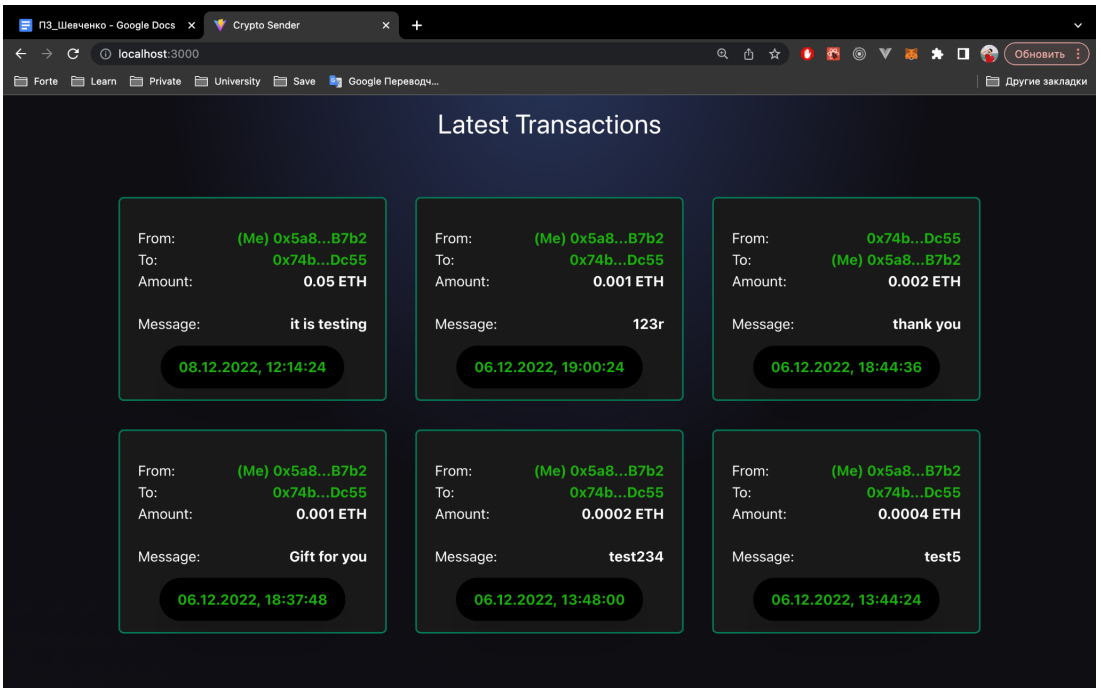

Рисунок 22 - Перегляд останньої виконаної транзакції

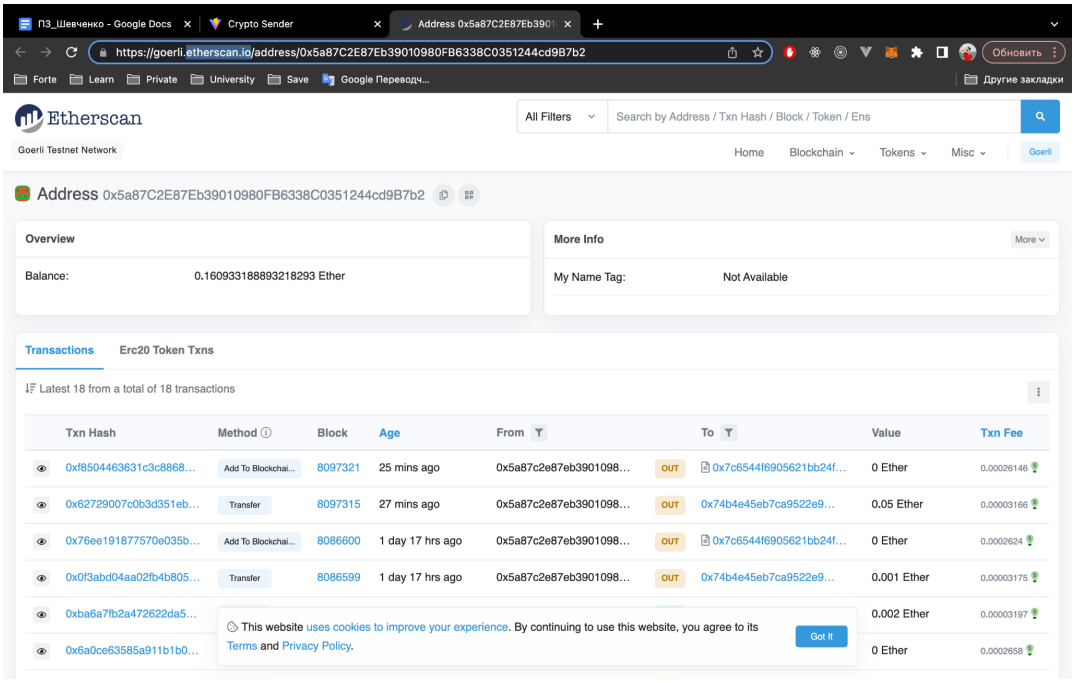

Рисунок 23 - Перевірка транзакції через etherscan.io

Також для перевірки правильності виконання даної операції можна відкрити розширення MetaMask та переглянути баланс кожного із криптогаманців. Та впевнитись, що відправка криптовалюти відбулася корректно, тобто із першого рахунку знялось 0.05 ETH з урахуванням комісії під назвою газ, а на інший додалась така ж кількість криптовалюти (Рисунок 24).

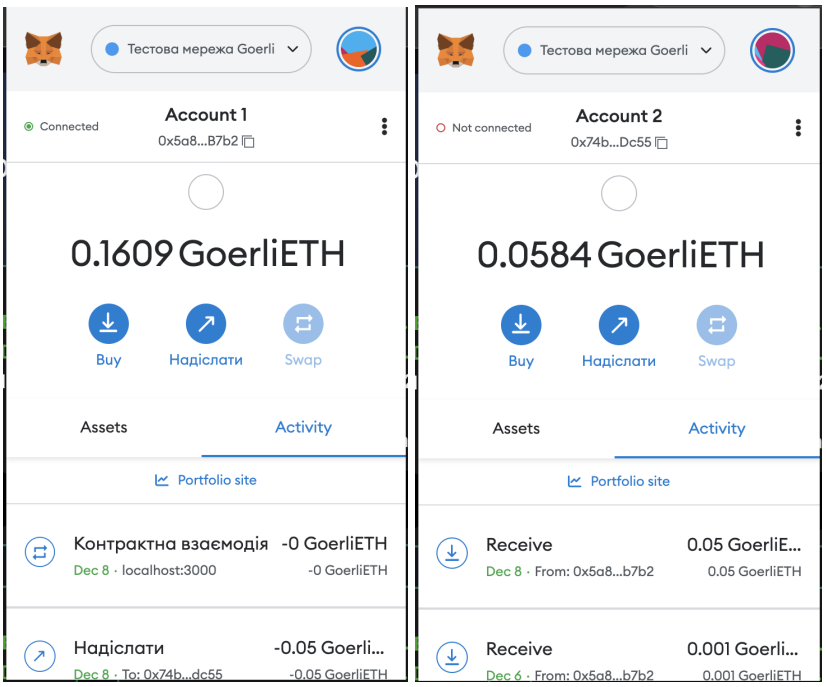

Рисунок 24 - Стан криптогаманців після виконання операції

4.2. Аналіз результатів роботи

В процесі тестування було перевірено роботу вебдодатку під назвою Crypto Sender. Результати, що були отримані в процесі функціонального тестування свідчать про те, що інтерфейсна частина додатку та розгорнутий смарт-контракт на блокчейн мережі Ethereum працюють коректно. Тобто користувач має змогу успішно авторизуватися за допомогою свого криптогаманця, виконувати транзакції по пересиланню криптовалюти із одного гаманця на інший та зручно контролювати та переглядати виконані транзакції в вигляді карток. Після аналізу результатів можна зробити висновок, що мети досягнуто - розробити швидке,

прозоре та надійне децентралізоване веборієнтоване програмне забезпечення для контролю та виконання транзакцій із криптовалютою з використанням технології Blockchain.

Крім, того даний вебдодаток має потенціал для покращення. Основним напрямком є додавання нового функціоналу у вигляді підтримки інших видів блокчейн мереж, крім Ethereum. Також при такому векторі розвитку, було б корисно додати можливість для конвертації різних криптовалют.

ВИСНОВКИ ДО ЧЕТВЕРТОГО РОЗДІЛУ МД

В даному розділі описано процес тестування розробленого програмного забезпечення у вигляді децентралізованого вебдодатку для контролю та виконання транзакцій з криптовалютою з використанням технології Blockchain. Також був проведений аналіз результатів отриманих під час тестування. Крім того, було запропоновано можливі варіанти для покращення даного вебдодатку у найближчому майбутньому, такі як додавання інших видів блокчейн мереж та криптовалют.

ВИСНОВКИ

В ході роботи над дисертацією досліджено принципи роботи технології Blockchain, проведено порівняння різних блокчейн мереж Ethereum та Bitcoin, проаналізовано архітектурні особливості створення децентралізованих вебдодатків, а також розглянуто інші технології, що були використані для створення децентралізованого веборієнтованого засобу для контролю та виконання транзакцій з криптовалютою. Перевагою такого засобу є те, що завдяки технології Blockchain, він є швидким, прозорим та надійним. Крім того децентралізований додаток ніколи не припиняє роботу завдяки відсутності центрального сервера, який міг би зупинитися.

В першому розділі дисертації було проведено аналіз версій World Wide Web для визначення найкращих методів та підходів для розробки веборієнтованого програмного забезпечення для контролю та виконання транзакцій над криптовалютою. Крім, того було проведено аналіз існуючих рішень та методів для виконання операцій над криптовалютою. Також була сформульована задача магістерської дисертації та визначена практична цінність роботи.

В другому розділі було розглянуто принципи роботи технології Blockchain та визначено її переваги та недоліки для використання в створенні децентралізованого програмного забезпечення. Визначено особливості використання блокчейн мережі Ethereum, порівняно із мережею Bitcoin. Розглянуто мову програмування Solidity, як інструмент для створення смарт-контрактів. Крім того, досліджено архітектурні особливості програмного забезпечення у вигляді децентралізованих вебдодатків. Також описано бібліотеку React для створення графічних інтерфейсів користувача.

Третій розділ дисертації присвячений опису програмної реалізації децентралізованого вебдодатку. Описано інструментарій, що був обраний для реалізації продукту, архітектуру вебдодатку, та принципи роботи основних модулів. Було використано JavaScript бібліотеку React для реалізації клієнтської частини та мову програмування Solidity для створення смарт-контракту.

В четвертому розділі описано процес тестування розробленого програмного забезпечення у вигляді децентралізованого вебдодатку для контролю та виконання транзакцій із криптовалютою з використанням технології Blockchain. Також був проведений аналіз результатів отриманих під час тестування. Крім того, було запропоновано можливі варіанти для покращення даного вебдодатку у найближчому майбутньому, такі як додавання інших видів блокчейн мереж та криптовалют.

Отже поставлену задачу було вирішено успішно. Був розроблений децентралізований веборієнтований засіб для контролю та виконання транзакцій з криптовалютою, який завдяки технології Blockchain є швидким у роботі у порівнянні з централізованими засобами проведення фінансових транзакцій, прозорим завдяки використанню децентралізованої структури, надійним та конфіденційним для користувачів через авторизацію за допомогою криптогаманця.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Клятченко Я.М., Шевченко Д.С. Порівняльний аналіз версій WORLD WIDE WEB. Збірник тез доповідей XV науково-практичної конференції магістрантів та аспірантів факультету прикладної математики «Прикладна математика та комп'ютинг – 2022», 18 листопада 2022 р. Київ : Київ. Національний технічний університет України ім. І. Сікорського С. 245–248.
2. Перший у світі веб-сайт. [Електронний ресурс] – Режим доступу : https://babel.ua/texts/23307-28-rokiv-tomu-zapustili-pershiy-u-sviti-veb-sayt-mi-pokazali-yak-viglyadav-bi-thebabel-v-1990-roci
3. What is Web 1.0, 2.0, and 3.0 and How They Compare. [Електронний ресурс] – Режим доступу : https://www.simplilearn.com/what-is-web-1-0-web-2-0-and-web-3-0-with-their-difference-article#what_is_web_30
4. Web 3.0 explained. [Електронний ресурс] – Режим доступу : https://www.investopedia.com/web-20-web-30-5208698
5. Arya Ghobadi. Everything about Web 3.0, 2021, - pp. 1- 30
6. Patrick Ejeke. Web3: What Is Web3? Potential of Web 3.0, 2022, - pp. 2-18
7. Transformation to Web 3.0. [Електронний ресурс] – Режим доступу: https://hacken.io/insights/web-3-0-transformation-explained-cybersecurity-implications/
8. Криптобіржа Binance: URL: https://www.binance.com/ (дата звернення: 20.09.2022)
9. Massimo Ragnedda. Blockchain and Web 3.0, 2019
10. Jared Tate. BlockChain 2035, 2019, - pp. 33-48
11. Andreas Antonopoulos, Gavin Wood. Mastering Ethereum: Building Smart Contracts and DApps. 1st Ed, 2018 - pp. 12-50
12. Документація мови програмування Solidity [Електронний ресурс] – Режим доступу : https://docs.soliditylang.org/en/v0.8.17/

13. Документація сервісу MetaMask [Електронний ресурс] – Режим доступу : https://docs.metamask.io/guide/
14. Документація бібліотеки React [Електронний ресурс] – Режим доступу : https://reactjs.org/docs/