Міністерство освіти і науки України

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління
(повна назва)

Кафедра електронних обчислювальних машин
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА

Пояснювальна записка

Рівень вищої освіти другий (магістерський)

Дослідження, проектування та розробка програмних компонентів для токенізації освітніх активів на основі смарт-контрактів Ethereum
(тема)

Виконав:

студент II курсу, групи СПм-20-2

Сербін О.М.
(прізвище, ініціали)

Спеціальність

123 «Комп’ютерна інженерія»
(код і повна назва спеціальності)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма

Системне програмування
(повна назва освітньої програми)

Керівник: Шматко О.В.
(посада, прізвище, ініціали)

Допускається до захисту

В.о. зав. кафедри ЕОМ _____ Волк М.О.
(підпис) (прізвище, ініціали)

2022 р.

Харківський національний університет радіоелектроніки

Факультет комп’ютерної інженерії та управління

Кафедра електронних обчислювальних машин

Рівень вищої освіти другий (магістерський)

Спеціальність 123 «Комп’ютерна інженерія»
(код і повна назва)

Тип програми освітньо-наукова
(освітньо-професійна або освітньо-наукова)

Освітня програма Системне програмування
(повна назва)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
“____” _____ 20__ р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

студенту Сербіну Олексію Максимовичу
(прізвище, ім’я, по батькові)

1. Тема роботи Дослідження, проектування та розробка програмних компонентів для токенізації освітніх активів на основі смарт-контрактів Ethereum

затверджена наказом по університету від “ 24 ” березня 2022 р. № 413 Ст

2. Термін подання студентом роботи до екзаменаційної комісії 18 травня 2022 р.

3. Вхідні дані до роботи платформа Ethereum, geth, мережа блокчейн на базі платформ ethereum

4. Перелік питань, що потрібно опрацювати у роботі

1) огляд літератури за темою роботи;
2) аналіз предметної області;
3) вибір та обґрунтування методики дослідження;
4) проведення експериментальних досліджень;
5) висновки

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (слайдів) 11 слайдів

6. Консультанти розділів роботи (заповнюється за наявності консультантів згідно з наказом, зазначеним у п.1)

Найменування розділу	Консультант (посада, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Огляд літератури за темою роботи	29.03.22-05.04.22	
2	Вибір та обґрунтування методики дослідження	06.04.22-16.04.22	
3	Вибір інструментальних засобів	17.04.22-29.04.22	
4	Проведення експериментів	30.04.22-04.05.22	
5	Оформлення матеріалів атестаційної роботи	05.05.22-10.05.22	
6	Подання кваліфікаційної роботи керівникові ті	11.05.22-12.05.22	
	Попередній захист		
7	Подання кваліфікаційної роботи на	13.05.22-17.05.22	
	рецензування		

Дата видачі завдання 28 березня 2022 р.

Студент _____
(підпис)

Керівник роботи _____ Шматко О.В.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ

Пояснювальна записка кваліфікаційної роботи: 80 с., 15 рис., 0 табл., 2 дод., 12 джерел.

КОМП’ЮТЕРНА МЕРЕЖА, ІНТЕРНЕТ, МАРШРУТИЗАТОР, ПРОТОКОЛ, СЕРВЕР, ШЛЮЗ, FIREWALL, WI-FI, WLAN.

Метою кваліфікаційної роботи є дослідження, проектування та розробка програмних компонентів для токенізації освітніх активів на основі смарт-контрактів Ethereum.

У ході виконання кваліфікаційної роботи було розроблено додаток для токенізації освітніх активів на основі смарт-контрактів Ethereum.

З ціллю створення децентралізованого розподіленого реєстру для токенізації освітніх активів, запропоновано використання технології блокчейн та смарт-контрактів.

У ході роботи запропоновано розподілений реєстр даних, який містить інформацію про студентів у вигляді цифрових токенів.

У першій главі наведено основні теоретичні відомості щодо тематики роботи.

У другій главі проведено аналіз цілісності та автентичності даних у технологіях блокчейн та смарт-контрактах, протоколів забезпечення конфіденційності.

У третій главі описано приклад використання розробленого програмного забезпечення та наведено.

ABSTRACT

Master’s thesis: 80 pages, 15 figures, 0 tables, 2 appendices, 12 sources.

FIREWALL, GATE, INTERNET, PROTOCOL, ROUTER, SERVER, WI-FI, WIRELESS NETWORK, WLAN.

The major goal of this thesis is research, design and development of software components for tokenization of educational assets based on Ethereum smart contracts.

In order to an application for tokenization of educational assets based on Ethereum smart contracts was developed.

In order to create a decentralized distributed registry for tokenization of educational assets, the use of blockchain technology and smart contracts is proposed.

In the course of the work, a distributed data register was proposed, which contains information about students in the form of digital tokens.

The first chapter provides basic theoretical information on the subject of the work.

The second chapter analyzes the integrity and authenticity of data in blockchain technologies and smart contracts, privacy protocols.

The third chapter describes an example of using the developed software and gives.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

XRP – Ripple

BTC – Bitcoin

ECDSA – Elliptic Curve Digital Signature Algorithm

ETH – Ethereum

SEC – The United States Securities and Exchange Commission

ICO – Initial coin offering

IOTA – Internet of Things

P2P – Person to Person

PoS – Proof of Stake

NIST – National Institute of Standards and Technology

PoW – Proof of Work

SHA – Secure Hash Algorithm

PI – Application programming interface

ВСТУП

На сьогоднішній день у світі відбувається революційна трансформація поничаючи з інформатизації і до оцифровки в основній людській діяльності.

Якщо інформатизація фактично передбачає модернізацію певних видів людської діяльності шляхом використання інформаційно-комунікаційних технологій, то цифрове перетворення (або оцифровка) передбачає їх якісну трансформацію починаючи з звичних видів та форм діяльності та до нових на основі цифрових моделей.

Трансформація цифрового оточення потребує як підтримки та розвитку існуючих умов програмування для появи перспективних наскрізних цифрових платформ і технологій, так і створення умов для появи нових платформ і технологій.

Приклади основних наскрізних цифрових технологій:

1) великі дані;
2) нейротехнології та штучний інтелект;
3) система розподіленої реєстрації, а саме блокчейн;
4) квантова технологія;
5) інноваційні технології виробництва;
6) промисловий Інтернет;
7) сенсорні компоненти та робот;
8) бездротовий зв'язок;
9) технологія віртуальної та доповненої реальності.

Продовжуючи робочий цикл цифрової трансформації освіти, у цій роботі розглядається та вівчається використання технології блокчейн (блокчейн) для токенізації освітніх активів та перспективи її застосування в освіті.

Ціллю даної роботи є дослідження, проектування та розробка програмного компонента для токенізації освітніх активів на основі смарт-

контрактів Ethereum.

У роботі треба вирішити наступні пункти:

1) аналіз методів забезпечення конфіденційності та цілісності даних у блочних розподілених регістрах;
2) аналіз хеш-алгоритму блокчейну;
3) вивчити засоби та механізми збереження конфіденційності даних у децентралізованих реєстрах;
4) токенізації активів освіти на основі технології блокчейн та моделювання.

Наукова новизна дослідження полягає у:

1) аналізування розвитку блокчейн технології у сфері освіти;
2) створені теоретичні концепції щодо використання блокчейн технології та організації децентралізованої систем для токенізації освітніх активів.

1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Аналіз розвитку блокчейн технології в освіті

Зазвичай виділяють три області використання блокчейн технології:

1) блокчейн 1.0;
2) блокчейн 2.0;
3) блокчейн 3.0.

Блокчейн 1.0 – це валюта. Криптовалюта використовується у різноманітних додатках, що мають відношення до коштів користувача, наприклад системи переказів та цифрових платежів.

Блокчейн 2.0 – це контракти. Цілі кластори ринкових, фінансових та економічних застосунків, у фундаменті яких можна використовувати блокчейн, працюють з різними видами фінансових інструментів – з облігаціями, ф'ючерсами, акціями, розумними активами і розумними контрактами, заставними, правовими титулами.

Блокчейн 3.0 – це додатки, область яких виходить за рамки фінансових транзакцій та ринків. До цієї області і буде належити блокчейн в освітній галузі.

За допомогою блокчейну можна покращити процеси викладання та навчання за ключовими аспектами.

Розширення прав і можливостей для учнів (самосуверенітет)

Через блокчейн дані (наприклад, облікові дані, отримані навички тощо), пов’язані з ідентичністю студентів, належать не центральному адміністратору, такому як університет, а студенту. Студенти мають можливість зберігати свої дані навчання протягом усього життя (як всередині класу, так і поза ним), повністю володіти ними та контролювати, хто має до них доступ (наприклад, роботодавці). Таким чином, учні можуть довести, що облікові дані в їхніх резюме точні та мають більше контролю над тим, до

чого можуть отримати доступ їхні роботодавці.

Варто зазначити, що навіть коли студенти отримують вигоду від «гаманців» блокчейну, де вони можуть зберігати всі свої навчальні дані та ділитися ними з різними сторонами (студенти є повними власниками даних, пов’язаних із особистістю), вони все одно отримують користь від підтримки своїх викладачів. , таким чином не залишаючись самотніми у своїх навчальних подорожах.

Підвищення безпеки та ефективності для навчальних закладів, бізнесу та учнів

Блокчейн має потенціал для забезпечення ідентичності, конфіденційності та безпеки даних студентів. Як було показано раніше в цій статті, блокчейн забезпечує безпеку та дійсність, забезпечуючи незмінність через свій хеш-ланцюжок. Наприклад, студенти не можуть змінити попередні освітні сертифікати, збережені в блокчейні, в той час як вони можуть легко зробити це за допомогою паперових записів. Крім того, конфіденційність забезпечується завдяки блокчейну, який не зберігає дані, а скоріше хеш даних. За бажанням, дані також можуть бути зашифровані перед збереженням у блокчейні.

Блокчейн гарантує, що студенти не можуть змінювати свої оцінки, ступені та сертифікати, таким чином пропонуючи роботодавцям гарантію того, що претенденти на роботу дійсно мають необхідні навички, щоб досягти успіху на робочому місці. Таким чином, блокчейн стає «якорем довіри однієї істини для облікових даних» (Tapscott and Kaplan, 2019). Крім того, цей якор також дає можливість створити кращі відповідності між шукачами роботи та роботодавцями. У більш широкому сенсі, оскільки технології розподіленої книги підтримують навчання та захищають академічні записи, вони покращують відносини між «коледжами, університетами, роботодавцями та їхніми стосунками з суспільством» через інтеграцію довіри та прозорості в транзакції та процеси обміну навичками.

Блокчейн надає можливість усьому людству оптимізувати

найрізноманітніші сфери життя. Однією із сильних сторін цієї технології є те, що її практично неможливо зламати та немає необхідності у втручанні третіх осіб. Весь принцип роботи блокчейна базується на математиці та криптографії. Згодом, блокчейн вселиться в усі сфери діяльності, у тому ж числі і в освіту. На сьогоднішній день існує ряд проблем в освіті. Одну з важливих проблем становить шахрайство у сфері підробки документів та проблема збереження документів.

Інфраструктура відкритих ключів таких як підписи та друк, вимагає використання центру сертифікації як посередника для видачі сертифікатів, створення залежності, яка може бути порушена. У випадку стихійних лих чи воєн також можуть бути пошкодженні або знищені дані документи.

На сьогоднішній день процес видачі та зберігання дипломів, є вельми довгим і складнім. На рисунку 1.1 представлений цей процес.

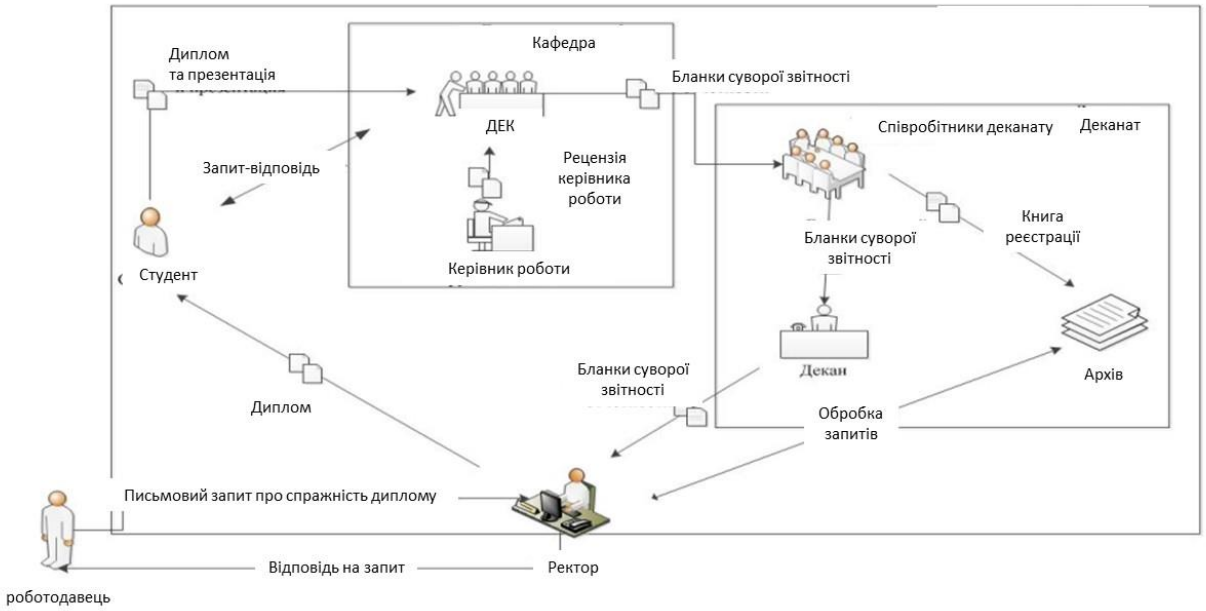


Рисунок 1.1 – Зберігання дипломів та порядок видачі

Захист дипломної роботи проводиться на нараді Екзаменаційної комісії (ЕК). Після проведення захисту, оголошується оцінки усіх робіт.

Тільки після повного заповнення бланку його ще потрібно перевірити

на відповідність та наявність недоліків.

Студент отримую диплом особисто чи висилається поштою, рекомендованим відправленням за заявою. Заява зберігається в особовій справі студента навчального закладу, так само там зберігається і копія диплома.

1.2 Аналіз та дослідження предметної області

За для вирішення гострої проблеми з зловживанням повноваженнями та корупіеї у вигляді підробки документів можна використовувати токенізацію документів освіти з використанням блокчейн технології. Рішення таких питтань є надзвичайно важливим для сучасної системи осіти, бо через такі складні перепони страждає якість освіти.

Принцип дії можна побачити на (рисунку 1.2).

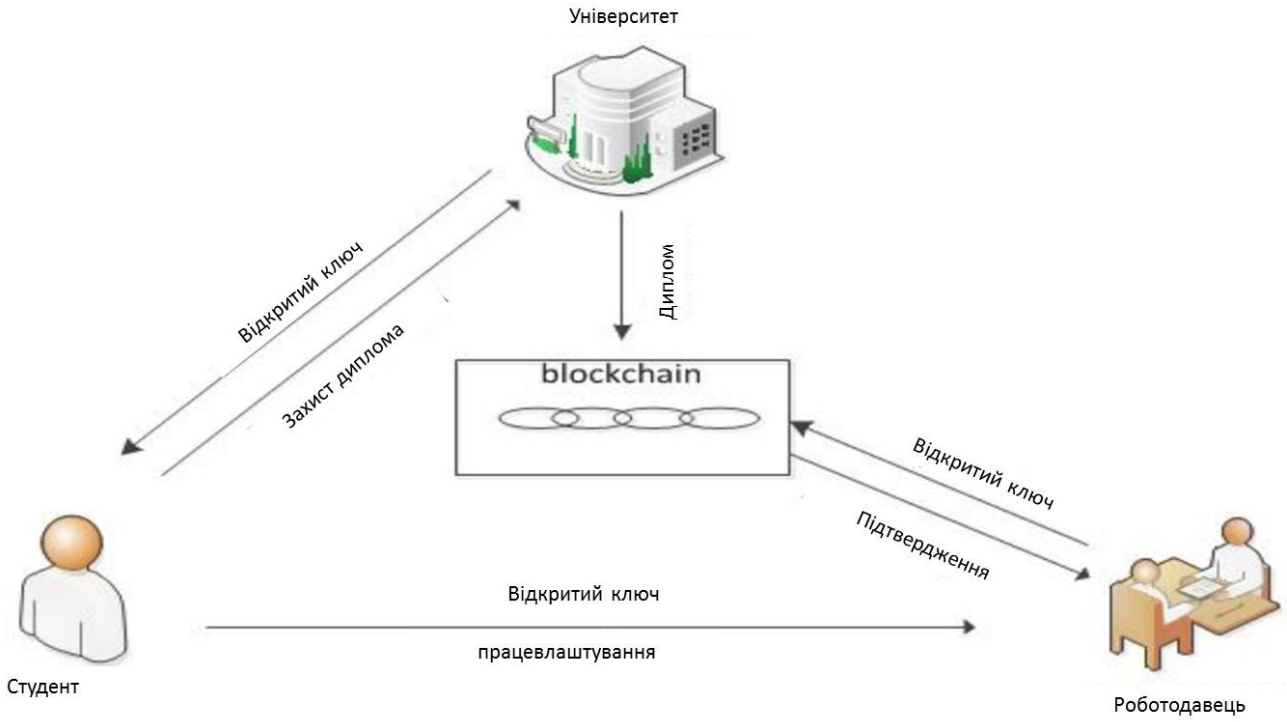


Рисунок 1.2 – Токенізації освітніх активів

Механізм роботи технологіх токенізації освітніх активів та запису його

у мережу блокчейн можна рознлянути на рисунку (рисунок 1.3).

Рисунок 1.3 – Принцип роботи технології блокчейн

Перш за все, траба створити цифровий токен, який мае базову інформацію, такі як одержувача диплома та назва університету, дату видачі, посвідчення та інше.

Наступнім кроком, навчальний заклад підписує вміст диплома закритим ключем. Тільки освітня огранізаця має доступ до закритого ключа. Підпис має бути підтвердженним мережевим вузлом та передаються в мережу блокчейн. Як результат, запис додають до блоку.

Навчальний заклад створюється хеш файл данних – короткий рядок букв та цифр(хеш), які можна використовувати задля перевірки, що ніхто не порушив вміст документу.

Переваги цього підходу полягать у тому, що докази сертифікатів будуть зберігатися без втрат, надійно та в незмінному реєстрі блокчейн. У такий спосіб, у такому віпадку, якщо заклади, що видали документ, зачинилися, то ці документи все ще можна перевірити щодо записів, що зберігаються в мережі блокчейн. Більш того, як тільки заклади видають документ, їм не потрібно робити додаткові витрати ресурсів, щоб підтвердити дійсність цього документа третім особам, так як вони зможуть перевіряти дипломи у вигляді ідентифікації ланцюжка блокчейн

безпосередньо.

1.3 Аналіз підстав впровадження блокчейн технології для токенізації

1.3.1 Блокчейн як особиста картка студента

На сьогоднішній день існує величесзна купа різних соціальних мереж такі, як електронні щоденники та інші сервіси, вже можуть надавати користувачам можливість записувати свої досягнення. Але, жоден з них не надає способи перевірки досвіду та облікових даних, описаних та включених у ці системи, тому ці додатки працюють як цифровий аналог коробки, повної паперових сертифікатів, які не отримують, практично жодних додаткових переваг або ефективності від процесу відцифрування.

Із технічної точки зору найпростішим способом реалізації, є створення перевіреного цифрового смарт-контракту. У той момент як користувачі завантажують досягнення, вони додаються в ланцюжок блоків, які вже перевіряються іншими вузлами блокчейн мережі, з використанням перевірки фактів досягнення. Як тільки певна кількість користувачів підтвердить вимогу як справжн та в залежності від репутації користувачів, які перевіряють, досягнення отримує оцінку довіри, яка є оцінкою її справжності. Вже є компанії, що тестують цей вид програмного послуг та забезпечення.

1.3.2 Блокчейн як спобіс для перевірки акредитації

На сьогоднішній день в Європі існує буквально сотні шляхів акредитації.

За для того, щоб отримати інформацію про, чи виданий документ легітимним органом, особа має перевірити наступні пункти:

1) закалад дійсно випустила саме цей диплом. Жодних доказів, як

до наданої освіти, представленого таким дипломом, не пред'являється;

2) акредитаційний орган, чи справді він акредитував установу;

3) повноваження, чи дійсно акредитуючі органи уповноважені так діяти.

Увесь цей процес, дуже трудомісткий та складний з технічної точки зору.

Існує велика кількість різних способів створення такого сценарію, кожен з яких передбачає, що акредитаційні організації публікують свої акредитаційні сертифікати, або підписи цих сертифікатів, на блокчейн.

1.3.3 Блокчейн для відстеження інтелектуальної власності.

Блокчейн можна використовувати для каталогізації та зберігання оригінальних робіт. Часто у авторів немає адекватних засобів для каталогізації своїх творів, а право власності на авторські права буває важко довести. Авторам також може бути важко побачити, хто використовує їх твір, і третім сторонам так само важко знати, у кого отримати ліцензію. Автори часто не можуть зупинити порушення чи успішно монетизувати свої роботи. З блокчейном авторські права не потрібно реєструвати і можуть виникнути автоматично після створення оригінальної кваліфікаційної роботи.

Іншою серйозною проблемою управління правами інтелектуальної власності є відстеження повного ланцюжка власності. Часто буває складно провести межу між тим, щоб надихнутися творчістю іншого музиканта та вкрасти її. Сумнозвісні випадки, пов’язані з авторськими правами в історії музики, показують, що визначення авторства та права власності часто є «неможливим». Блокчейн може вирішити ці недоліки системи. Наприклад, платформи на основі блокчейну, такі як Binded, дозволяють авторам реєструвати право власності на авторські права, яке потім можна використовувати, щоб побачити, де і як твір використовується в Інтернеті, а

також для отримання ліцензій у третіх сторін. Binded поєднує інтеграцію з Бюро авторських прав США, Instagram та Twitter для моніторингу використання зображень, захищених авторським правом. Реєстрація роботи через Binded надає цифровий сертифікат автентичності. Ця реєстрація може допомогти третім сторонам ідентифікувати автора твору та допомогти власникам IP у боротьбі з порушеннями. Наразі власники IP мають труднощі із захистом роботи IP в Інтернеті, тобто після того, як робота IP завантажується в Інтернет, стає важко контролювати цю роботу та відстежувати, хто з якою метою її використовує.

Коли робота IP зареєстрована та перевірена за допомогою платформ на основі блокчейну, автори можуть шукати в цілій низці різних джерел одночасно, щоб з'ясувати, хто використовує їх роботу. Це дає змогу власникам інтелектуальної власності виявляти та зупиняти порушення, а також полегшує ліцензування своїх робіт з інтелектуальної власності. У цьому сенсі блокчейн може служити інструментом примусового виконання. З системою реєстрації на основі блокчейну перевірити, чи порушує нова пісня існуючу IP-адресу раніше зареєстрованої пісні, буде набагато простіше. Цей тип системи виявлення на основі блокчейна можна застосувати до тексту, мистецтва та музики за допомогою штучного інтелекту. Зі структурної точки зору перевага використання цього підходу полягає в тому, що ситуація дуже схожа на існуючі системи, які вже використовуються для відстеження цитат у журнальних статтях. Однак для відстеження цитат все ще потрібно, щоб посередники обмежували використання цих статей, часто у формі обмеженого доступу, використання інтелектуальної власності та високої вартості доступу. Це обмежує використання моделі OER.

Використання блокчейну усуває посередника, дозволяючи будь-кому публічно публікувати та точно відстежувати повторне використання, незалежно від вихідного матеріалу.

Якби систему було запроваджено таким чином, це дозволило б винагороджувати вчених на основі рівня повторного використання та

фактичного використання матеріалу їхніх знань, так само, як вони тепер винагороджуються на основі посилань у наукових роботах.

1.3.4 Використання задля ідентифікації студентів

При використанні технології блокчейн, після передачі студентами своїх персональних даних до приймальної комісії навчального закладу, студенти отримують ідентифікаційну картку – ключ. Використовуючи біометричні дані, наприклад, у поєднанні з цим ключем на смартфоні, студенти зможуть ідентифікувати себе з будь-якою іншою частиною організації, яка повинна їх ідентифікувати, наприклад, їдальня, гуртожитки, бібліотека, викладачі, студентська спільнота і незабаром. Кожен із цих блоків може ідентифікувати учня, не запитуючи чи не зберігаючи будь-яку особисту інформацію.

Перевага цього підходу полягає в тому, що коли блокчейн використовується в процесі аутентифікації, доступ до даних мають лише ті, хто відповідає за перевірку особистості студента. Крім того, єдиною особою, яка володіє даними, є сам студент. Це означає, що організаціям більше не потрібно керувати складними системами прав доступу, лише захищені пристрої або мережі, які пройшли автентифікацію. Це заощадить багато ресурсів на посилення мереж, боротьбу з порушеннями даних, навчання співробітників захисту даних та управління правами доступу. Крім того, ті, хто взаємодіє зі студентами в організації, не несуть відповідальності за конфіденційність і зберігання даних, оскільки вони не потребують або зберігають цю інформацію.

1.4 Висновки до розділу 1

На сьогоднішній день застосування блокчейн технологій в освітніх організаціях знаходяться на експериментальному рівні. Але вже існують такі університети, які запустили тестові проекти з впровадження технології

блокчейн, прикладом є Массачусетський технологічний інститут а також університет Нікосії на Кіпрі.

Чи можна роботи якісь остаточні, суб'єктивні висновки базуючись на усьому, що було заначено у робі вище? На мою думку, ми ще не розглянули усі ключові аспекти цієї технології, все це ще требе розглядати більш докладно, що ми і зробимо розглянувши наступні пункти роботи.

2 ДОСЛІДЖЕННЯ ЗАСОБІВ ТОКЕНІЗАЦІЇ ОСВІТНІХ АКТИВІВ НА ОСНОВІ ТЕХНОЛОГІЇ БЛОКЧЕЙН І СМАРТ-КОНТРАКТІВ ETHEREUM

2.1 Поняття блокчейну

«Блокчейн – це постійний цифровий розподілений блокнот економічних транзакцій, який можна запрограмувати для запису не тільки фінансових транзакцій, але й практично будь-якої цінності», – Дон і Алекс Тепскотт, «Революція блокчейн» (2016) рік.

Основна система блокчейну – це постійно зростаюча послідовність блоків, які поширюються між учасниками через однорангову мережу. В результаті формується база даних, яка керується автономно без єдиного центру. Це робить блокчейн дуже зручним для запису подій і маніпулювання даними, керування ідентифікацією та перевіркою походження.

Позначки часу (хеш-суми) додаються до кожного блоку. Ці блоки розташовані в ланцюжок («блокчейн» - буквально «блокчейн») у строгому порядку. Якщо ви спробуєте змінити порядок блоків, система відхилить ланцюжок через невідповідність структури та хеш-суми. Кожен блок зберігає в собі свій хеш-код підсумований з хеш-кодом блоку перед ним, що вибудовує однозв'язного залежність між блоками в ланцюзі. Схематично це можна побачити на рисунку (Рисунку 2.1).

Щоб запобігти зміні часових позначок і правильному з точки зору системи вказану кількість хешу, блокчейни використовують кілька методів захисту: Proof of Work (PoW) і Proof of Stake (PoS).

Протокол підтвердження роботи, Ethash, вимагає від майнерів пройти інтенсивну гонку проб і помилок, щоб знайти nonce для блоку. До ланцюжка можна додати лише блоки з дійсним одноразовим номером.

Під час змагань за створення блоку майнер буде багаторазово розміщувати набір даних, який ви можете отримати лише завантаживши та

запустивши повний ланцюжок (як це робить майнер), за допомогою математичної функції. Набір даних використовується для генерування mixHash нижче цільового одноразового номера, що диктується складністю блоку. Найкращий спосіб зробити це шляхом проб і помилок.

Складність визначає ціль для хешу. Чим нижча ціль, тим менший набір дійсних хешів. Після створення це неймовірно легко перевірити іншим майнерам і клієнтам. Навіть якби одна транзакція змінилася, хеш був би зовсім іншим, сигналізуючи про шахрайство.

Хешування дозволяє легко виявити шахрайство. Але підтвердження роботи як процес також є серйозним стримуючим фактором для атаки на ланцюжок.

Майнерів стимулюють виконувати цю роботу в головній мережі Ethereum. У підгрупи майнерів мало стимулів створити власний ланцюжок – це підриває систему. Блокчейни покладаються на наявність єдиної держави як джерела істини. А користувачі завжди виберуть найдовшу або «найважчу» ланцюг.

Метою перевірки роботи є розширення ланцюжка. Найдовший ланцюжок найбільш вірогідний як дійсний, тому що в ньому виконано найбільше обчислювальних робіт. У системі PoW Ethereum майже неможливо створити нові блоки, які стирають транзакції, створюють підроблені або підтримують другий ланцюжок. Це тому, що зловмисний майнер повинен завжди розв’язувати блок один раз швидше, ніж усі інші.

Щоб постійно створювати шкідливі, але дійсні блоки, вам знадобиться понад 51% потужності мережевого майнінгу, щоб перемогти всіх інших. Вам знадобиться велика обчислювальна потужність, щоб виконати таку кількість «роботи». А витрачена енергія може навіть переважити виграш, який ви отримаєте під час атаки.

Proof-of-work також відповідає за випуск нової валюти в систему та стимулювання майнерів виконувати роботу.

Майнери, які успішно створили блок, отримують винагороду двома щойно випущеними ETH, але більше не отримують всі комісії за транзакцію, оскільки базова комісія спалюється, а чайові та винагороду за блок дістаються майнеру. Майнер також може отримати 1,75 ETH за блок дядька. Блоки Uncle — це дійсні блоки, створені майнером практично одночасно з тим, як інший майнер видобув успішний блок. Блокування дядька зазвичай відбувається через затримку мережі.

Транзакція має "кінцевість" в Ethereum, коли вона є частиною блоку, який не може змінитися.

Оскільки майнери працюють децентралізовано, два дійсні блоки можуть бути видобуті одночасно. Таким чином утворюється тимчасова вилка. Зрештою, один з цих ланцюжків стане прийнятим ланцюгом після того, як наступний блок буде видобуто та додано, що зробить його довшим.

Але щоб ще більше ускладнити ситуацію, транзакції, відхилені на тимчасовій форці, могли бути включені в прийнятий ланцюжок. Це означає, що воно може бути зворотним. Таким чином, остаточність відноситься до часу, який ви повинні чекати, перш ніж вважати транзакцію незворотною. Для Ethereum рекомендований час становить шість блоків або трохи більше 1 хвилини. Після шести блоків можна з відносною впевненістю сказати, що транзакція пройшла успішно. Ви можете чекати довше, щоб отримати ще більші гарантії.

При розробці програмних додатків слід пам’ятати про остаточність. Було б поганим користувацьким враженням відображати інформацію про транзакції в оману, особливо якщо транзакція має високу цінність.

Пам’ятайте, що цей час не включає час очікування транзакції, підібраної майнером.

Простіше кажучи, механізм PoW надає вузлам мережі можливість перевіряти, чи майнери дійсно виконували обчислення. Цей процес включає спробу знайти хеш заголовка блоку, який відповідає цій складності.

У методі Proof-of-Stake вузли також намагаються хешувати дані,

шукаючи результати, менші за певне значення, але складність у цьому випадку пропорційна та розподілена на основі балансу вузла. Іншими словами – на основі кількості токенів в обліковому записі користувача.

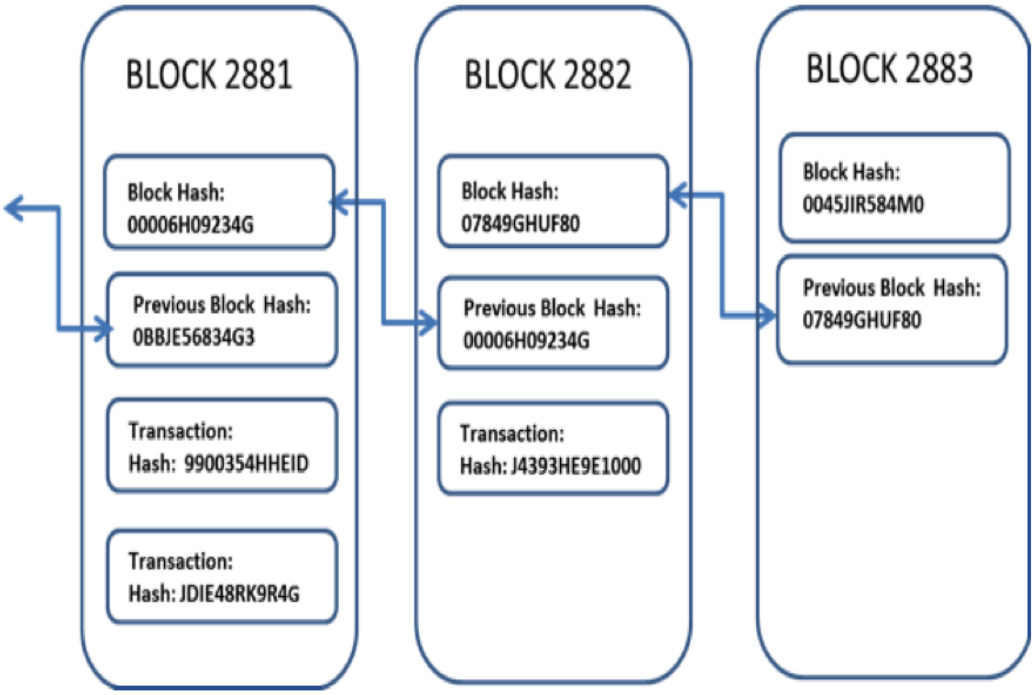


Рисунок 2.1 – Типова структура блокчейн

Підсумовуючи всі факти, неважко зробити висновок, що блокчейн дуже надійний і водночас децентралізований. Усі учасники, які підтримують послуги мережі, рівні. Тут відсутній сервер чи будь-який процесинговий центр.

У центрі уваги всій технології є формування та закриття блоків. Як зазначалося вище, кожна ланка ланцюга (блока) містить певний ключ. Блок не буде закритий, доки не буде розшифрований. Майнінг відповідає за розшифровку в криптовалюту. Майнери, які займаються майнінгом криптовалюти, роблять це за допомогою відеокарт і процесорів. У свою чергу, вони виконують обчислювальні операції, основна мета яких полягає в пошуку криптографічного підпису блоку у вигляді хешу. Після вибору – блок закривається і майнер отримує винагороду у вигляді криптовалюти.

Роботу блокчейну та його безпеку забезпечують майнери та інші учасники блокчейну. Їх ще називають вузлами або вузлами. Повні вузли відносяться до майнерів і звичайних користувачів зрілих гаманців. Це означає, що вони мають повну версію блокчейну на своєму комп'ютері чи іншому пристрої. Чим активніше повні вузли в блокчейні, тим швидше може бути оброблена інформація про транзакцію.

Підводячи проміжні підсумки, нижче надані основні особливості технології блокчейн:

1) прозорість системи – інформація про транзакції, контракти та таке інше зберігається у відкритому доступі. При цьому ці дані неможливо змінити;

2) теоретична необмеженість кількості блоків – теоретично блокчейн можна доповнювати записами до нескінченності. Саме із цієї причини його часто намагаються порівняти із суперкомп'ютером;

3) децентралізація системи – в блокчейн ланцюжку немає сервера. Усі учасники мережі блокчейн – це і є сервер. Він забезпечує роботу усієї мережі;

4) надійність системи – для створення нового запису необхідний погодження вузлів блокчейна. Це дозволяє записувати тільки легітимні транзакції та фільтрувати операції, тому здійснити підміну хеша неможливо.

Блокчейн – це спосіб розподіленого сховища. Технологію можна використовувати для запису та відстеження будь-якої інформації, від медичних записів до виборів.

Основна відмінність блокчейна від стандартних баз даних – децентралізація. При цьому, по-перше, процес не контролюється жодним контролюючим органом чи організацією.По-друге, інформація не зосереджена на серверах в одному місці, а розподілена по величезних комп'ютерних мережах по всьому світу. Перед тим, як блок зможе увійти в ланцюг, має відбутися ряд подій.

По-перше, протокол повинен бути верифікований. На відміну від

класичних транзакцій, схвалених банками або платіжними системами, транзакції в блокчейні підтверджуються мережею комп'ютерів. Зазвичай мережа складається з тисяч або навіть мільйонів машин по всьому світу.

По-друге, після перевірки транзакції інформація відправляється в блок. Він містить дату, час, суму та цифрові підписи обох сторін.

Вважаю необхідним перерахувати сильні сторони технології:

1) Ваші дані є конфіденційними та вирішальними, і блокчейн може суттєво змінити те, як розглядається ваша критична інформація. Створюючи запис, який не можна змінити та зашифрований наскрізь, блокчейн допомагає запобігти шахрайству та несанкціонованій діяльності. Проблеми з конфіденційністю також можна вирішити в блокчейні, анонімізуючи персональні дані та використовуючи дозволи для запобігання доступу. Інформація зберігається в мережі комп'ютерів, а не на одному сервері, що ускладнює перегляд даних хакерам.
2) Без блокчейну кожна організація повинна вести окрему базу даних. Оскільки блокчейн використовує розподілену книгу, транзакції та дані записуються однаково в кількох місцях. Усі учасники мережі з дозволеним доступом бачать однакову інформацію одночасно, забезпечуючи повну прозорість. Усі транзакції записуються про незмінність і мають штамп часу та дати. Це дозволяє учасникам переглядати всю історію транзакцій і практично виключає будь-яку можливість шахрайства.
3) Блокчейн створює аудиторський слід, який документує походження активу на кожному кроці його шляху. У галузях, де споживачі стурбовані екологічними проблемами чи проблемами прав людини, пов'язаними з продуктом — або галуззю, яка страждає від підробок і шахрайства — це допомагає надати докази. Завдяки блокчейну можна ділитися даними про походження безпосередньо з клієнтами. Дані про відстеження також можуть виявити слабкі місця в будь-

якому ланцюжку поставок — де товари можуть стояти на вантажній доці в очікуванні транзиту.

4) Традиційні процеси, що містять багато паперу, займають багато часу, схильні до людських помилок і часто вимагають посередництва третьої сторони. Завдяки оптимізації цих процесів за допомогою блокчейну транзакції можна виконувати швидше та ефективніше. Документація може зберігатися в блокчейні разом з деталями транзакцій, що усуває необхідність обміну паперу. Немає необхідності узгоджувати кілька бухгалтерських книг, тому очищення та розрахунки можуть бути набагато швидшими.
5) Трансакції можна навіть автоматизувати за допомогою «розумних контрактів», які підвищують вашу ефективність і ще більше прискорюють процес. Після виконання попередньо визначених умов автоматично запускається наступний крок транзакції або процесу. Розумні контракти зменшують людське втручання, а також залежність від третіх сторін у перевірці дотримання умов контракту. Наприклад, у страхуванні, як тільки клієнт надає всю необхідну документацію для подання претензії, претензія може бути автоматично врегульована та оплачена.

Існує кілька способів побудувати мережу блокчейн. Вони можуть бути державними, приватними, з дозволом або створені консорціумом.

А також існує різни види мереж, а саме:

1) Публічний блокчейн – це той, до якого може приєднатися будь-який бажаючий, наприклад біткойн. Недоліки можуть включати значну необхідну обчислювальну потужність, низьку конфіденційність транзакцій або її відсутність та слабку безпеку. Це важливі міркування щодо корпоративного використання блокчейну.
2) Приватна мережа блокчейн, подібна до публічної мережі блокчейн, є децентралізованою одноранговою мережею. Однак одна організація керує мережею, контролюючи, кому дозволено брати участь,

виконувати протокол консенсусу та підтримувати спільний реєстр. Залежно від варіанту використання це може значно підвищити довіру та впевненість між учасниками. Приватний блокчейн можна запускати за корпоративним брандмауером і навіть розміщувати в приміщенні.

3) Компанії, які створили приватний блокчейн, зазвичай створюють дозволену мережу блокчейну. Важливо зазначити, що публічні мережі блокчейну також можуть мати дозвіл. Це накладає обмеження на те, кому дозволено брати участь у мережі та в яких транзакціях. Учасникам необхідно отримати запрошення або дозвіл на приєднання.
4) Кілька організацій можуть розділити відповідальність за підтримку блокчейну. Ці попередньо відібрані організації визначають, хто може здійснювати транзакції або отримувати доступ до даних. Блокчейн консорціуму ідеально підходить для бізнесу, коли всі учасники повинні мати дозвіл і нести спільну відповідальність за блокчейн.

2.2 Хешування у блокчейн

2.2.1 Криптографія

Блокчейн – це новаторська технологія, яка робить можливими криптовалюти. Без безпеки та можливості запису блокчейну криптовалюта не мала б реальної цінності, оскільки будь-хто міг створити будь-яку суму грошей, яку забажає.

Більшість із нас знає, що блокчейн – це технологія розподіленого реєстру, що захищає біткойн, Ethereum, Cardano (ADA), монети Binance (BNB), dogecoin та інші криптовалюти.

Ми можемо шифрувати дані кількома різними способами. Кожен має

переваги та недоліки, і ми також можемо використовувати їх для створення більш надійного процесу шифрування. Давайте розглянемо три основні типи криптографії.

Симетрична криптографія – або криптографія симетричного ключа – була першим типом шифрування, який використовувався в Інтернеті. Симетрична криптографія перетворює інформацію в шифр або зашифрований код. Щоб розшифрувати шифр, потрібен ключ.

У симетричній криптографії і відправник, і одержувач використовують один і той же ключ для шифрування та дешифрування даних. Оскільки це настільки просто, симетрична криптографія може дуже швидко обробляти великі обсяги даних.

Однак, як ви можете собі уявити, спільний доступ до ключів став проблемою. Подумайте, коли ви намагаєтеся комусь поділитися паролем. Якщо ви надішлете цей пароль текстом або електронною поштою, хакери легко його побачать. Вам майже потрібен пароль для вашого пароля!

Аналогічно, обмін ключами від відправника до одержувача створив уразливість, яку хакери могли досить швидко використати.

При асиметричній криптографії відправник і одержувач мають різні ключі. Один ключ використовується для шифрування інформації, а окремий ключ використовується для розшифрування цієї інформації на іншому кінці.

Але якщо у двох людей по суті різні паролі, як ви гарантуєте, що, коли інформація надсилається, її зможе відкрити лише правильний одержувач? Коротше кажучи, без спільного доступу до ключів, як ви вкажете, щоб код відкривався для потрібної людини?

Для вирішення цієї проблеми асиметрична криптографія використовує систему з двох ключів на користувача: відкритий ключ і закритий ключ. Ваш відкритий ключ унікальний для вас, але всі інші також можуть його бачити. Ніхто не знає вашого приватного ключа, крім вас. Це як PIN-код вашого банківського рахунку.

Загальний і закритий ключі працюють разом. Таким чином, під час

транзакції особа, яка надсилає інформацію, може надіслати її на ваш відкритий ключ. Потім, щоб розшифрувати дані, надіслані на ваш відкритий ключ, ви повинні мати закритий ключ, щоб його розблокувати.

Щоб надіслати комусь повідомлення, ви повинні зашифрувати його відкритим ключем. Тоді лише вони зможуть розблокувати його за допомогою свого приватного ключа. Крім того, якщо хтось додає цифровий підпис до набору даних за допомогою свого приватного ключа, будь-хто онлайн може використовувати свій відкритий ключ, щоб розшифрувати підпис і перевірити, що це дійсно він.

Останнім типом криптографії є хешування. Криптографічний хеш - це набір тексту. Будь-яку відкриту текстову інформацію можна пропустити за допомогою алгоритму хешування та перетворити на унікальний рядок тексту. Текст нічого не означає. Наприклад, слово «Привіт» можна перетворити на хеш sha1: «f7ff9e8b7bb2e09b70935a5d785e0cc5d9d0abf0».

Після того, як вихідні дані проходять через криптографічну хеш-функцію, ви не можете змінити процес. Ось чим криптографічне хешування відрізняється від симетричного або асиметричного шифрування, яке можна розшифрувати за допомогою ключа. Неможливо почати з хешу, якого ви ніколи раніше не бачили, і зробити висновок, якими були вихідні дані.

Поки використовується той самий алгоритм хешування, одні й ті ж дані завжди будуть тим самим хешем. Тож, чи змінилися дані на цьому шляху, користувачі можуть визначити, порівнявши їх із остаточним хешем. Однак хакери знайшли способи захопити багато хешів, а потім порівняти їх із хешами для поширених слів і фраз. Якщо вони знайдуть відповідність, то знають, що означає хеш. Ось як хакери крадуть паролі під час порушення даних.

Іншою важливою особливістю хешування є те, що ви можете звести цілий масив даних до невеликого рядка тексту в хеші. Хеші завжди мають однакову довжину, незалежно від того, наскільки довгі чи великі дані. Отже, хешування — це спосіб стиснення інформації. Ми пояснимо докладніше,

чому це так важливо, але спочатку давайте введемо блокчейн.

2.2.2 DAPP

Децентралізована програма (dapp) – це програма, побудована на децентралізованій мережі, яка поєднує смарт-контракт і інтерфейс користувача. В Ethereum смарт-контракти доступні та прозорі, як і відкриті API, тому у вашому програмному забезпеченні можна навіть включити смарт-контракт, який написав хтось інший.

Dapp має свій бекенд-код, що працює в децентралізованій однорангові мережі. Порівняйте це з додатком, де серверний код виконується на централізованих серверах.

Dapp може мати код інтерфейсу та користувацькі інтерфейси, написані будь-якою мовою (так само, як програма), щоб здійснювати виклики до свого бекенда. Крім того, його інтерфейс може бути розміщений на децентралізованому сховищі, такому як IPFS.

Різновидності використання DAPP:

1) Децентралізовані – dapps працюють на Ethereum, відкритій публічній децентралізованій платформі, де жодна особа чи група не контролюють;
2) Детерміновані - dapps виконують одну і ту ж функцію незалежно від середовища, в якому вони виконуються;
3) Тьюрінга завершено - dapps можуть виконувати будь-які дії з урахуванням необхідних ресурсів;
4) Ізольовані – dapps виконуються у віртуальному середовищі, відомому як віртуальна машина Ethereum, тому, якщо в смарт-контракті є помилка, це не перешкоджатиме нормальному функціонуванню мережі блокчейну.

2.2.3 SHA-256

Алгоритм SHA-256 є одним із різновидів SHA-2 (Secure Hash Algorithm 2), який був створений Агентством національної безпеки в 2001 році як наступник SHA-1. SHA-256 — це запатентована криптографічна хеш-функція, яка виводить значення довжиною 256 біт.

Що таке хешування? Під час шифрування дані перетворюються в захищений формат, який неможливо прочитати, якщо одержувач не має ключа. У зашифрованому вигляді дані можуть мати необмежений розмір, часто стільки ж, скільки й у незашифрованому вигляді. У хешуванні, навпаки, дані довільного розміру відображаються на дані фіксованого розміру. Наприклад, 512-бітовий рядок даних буде перетворений на 256-бітовий рядок за допомогою хешування SHA-256.

У криптографічному хешуванні хешовані дані змінюються таким чином, що робить їх абсолютно нечитабельними. Було б практично неможливо перетворити 256-бітовий хеш, згаданий вище, назад до його початкової 512-бітної форми. Тож навіщо вам створити зашифроване повідомлення, яке не можна відновити? Найпоширенішою причиною є перевірка вмісту даних, які необхідно тримати в секреті. Наприклад, хешування використовується для перевірки цілісності захищених повідомлень і файлів. Хеш-код захищеного файлу можна опублікувати для всіх, щоб користувачі, які завантажили файл, могли підтвердити, що вони мають автентичну версію, без розкриття вмісту файлу. Аналогічно хеші використовуються для перевірки цифрових підписів.

Перевірка пароля є особливо важливою програмою для криптографічного хешування. Зберігання паролів користувачів у текстовому документі є рецептом катастрофи; будь-який хакер, якому вдасться отримати доступ до документа, виявить скарбницю незахищених паролів. Тому безпечніше зберігати хеш-значення паролів. Коли користувач вводить пароль, хеш-значення обчислюється, а потім порівнюється з таблицею. Якщо

він відповідає одному із збережених хешів, це дійсний пароль, і користувач може отримати доступ.

SHA-256 є однією з найбільш безпечних функцій хешування на ринку. Уряд США вимагає від своїх агентств захищати певну конфіденційну інформацію за допомогою SHA-256. Хоча точні деталі того, як працює SHA-256, засекречені, ми знаємо, що він побудований за допомогою структури Меркла-Дамгарда, похідної від односторонньої функції стиснення, яка сама була створена за допомогою структури Девіса-Меєра зі спеціалізованого блочного шифру.

Три властивості роблять SHA-256 таким безпечним. По-перше, відновити вихідні дані з хеш-значення практично неможливо. Атака з грубою силою повинна зробити 2256 спроб для створення вихідних даних. По-друге, наявність двох повідомлень з однаковим хеш-значенням (так звана колізією) вкрай малоймовірна. Маючи 2256 можливих хеш-значень (більше, ніж кількість атомів у відомому Всесвіті), ймовірність того, що два будуть однаковими, нескінченно мала, неймовірно мала. Нарешті, незначна зміна вихідних даних змінює хеш-значення настільки, що стає неочевидним, що нове хеш-значення походить із подібних даних; це відомо як лавинний ефект. Графічне уявлення однієї ітерації обробки блоку даних продемонстровано на (Рисунку 2.2).

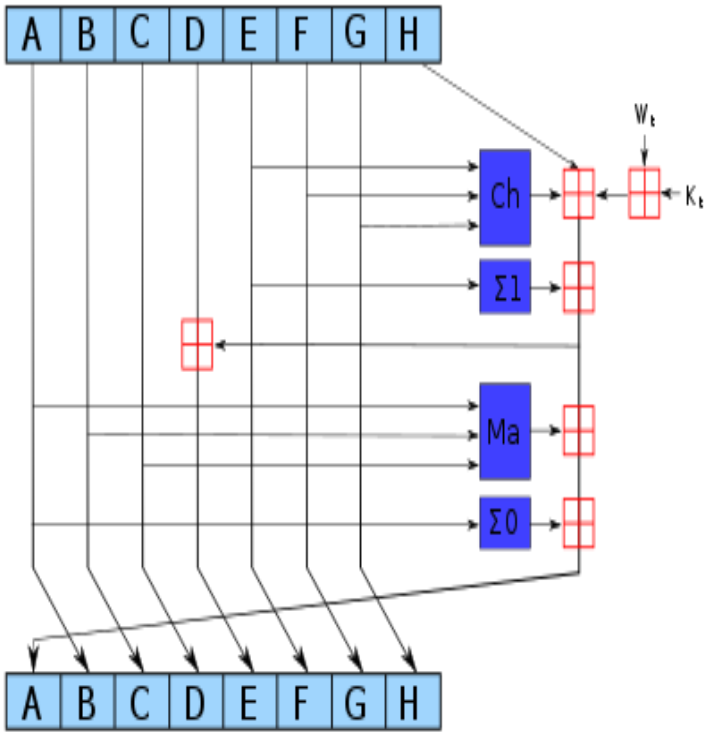


Рисунок 2.2 – Обробка блоку у SHA-256

Хешрейт для криптовалют, працюючих на основі SHA-256, обчислюється в одиницях Gigahash за секунду (GH/s). На створення блоку йде від шести до десяти хвилин.

2.2.4 Ethash

Ethash – це алгоритм криптовалюта, створений, розроблений спеціально для Ethereum. Алгоритм базується на псевдовипадковий набір даних, ініціалізований поточного розміру ланцюжка блоків (DAG – файли, які відновлюються кожні 30 000 блоків, приблизно 5 днів.

Хід роботи алгоритму хешування Ethash можливо узагальнити як показано на (Рисунку 2.3).

Заголовок, отриманий з останнього блоку ланцюга та Поточне число в поєднанні з використанням SHA-3-подібного алгоритму, створюють

первинні 128 байти міксу. Mix використовується у процесі задля обчислення того, яка 128-байтовую сторінка з групи DAG витягується.

Хейшрейт алгоритму Ethash можна вімурювати в умовних одиницях, наприклад Megahash в секунду (MH / s).

Чи можна роботи якісь остаточні, суб'єктивні висновки базуючись на усьому, що було заначено у робі вище? На мою думку, ми ще не розглянули усі ключові аспекти цієї частини розділу, все це все ще требе розглядати більш докладно, що ми і зробимо розглянувши наступні пункти роботи.

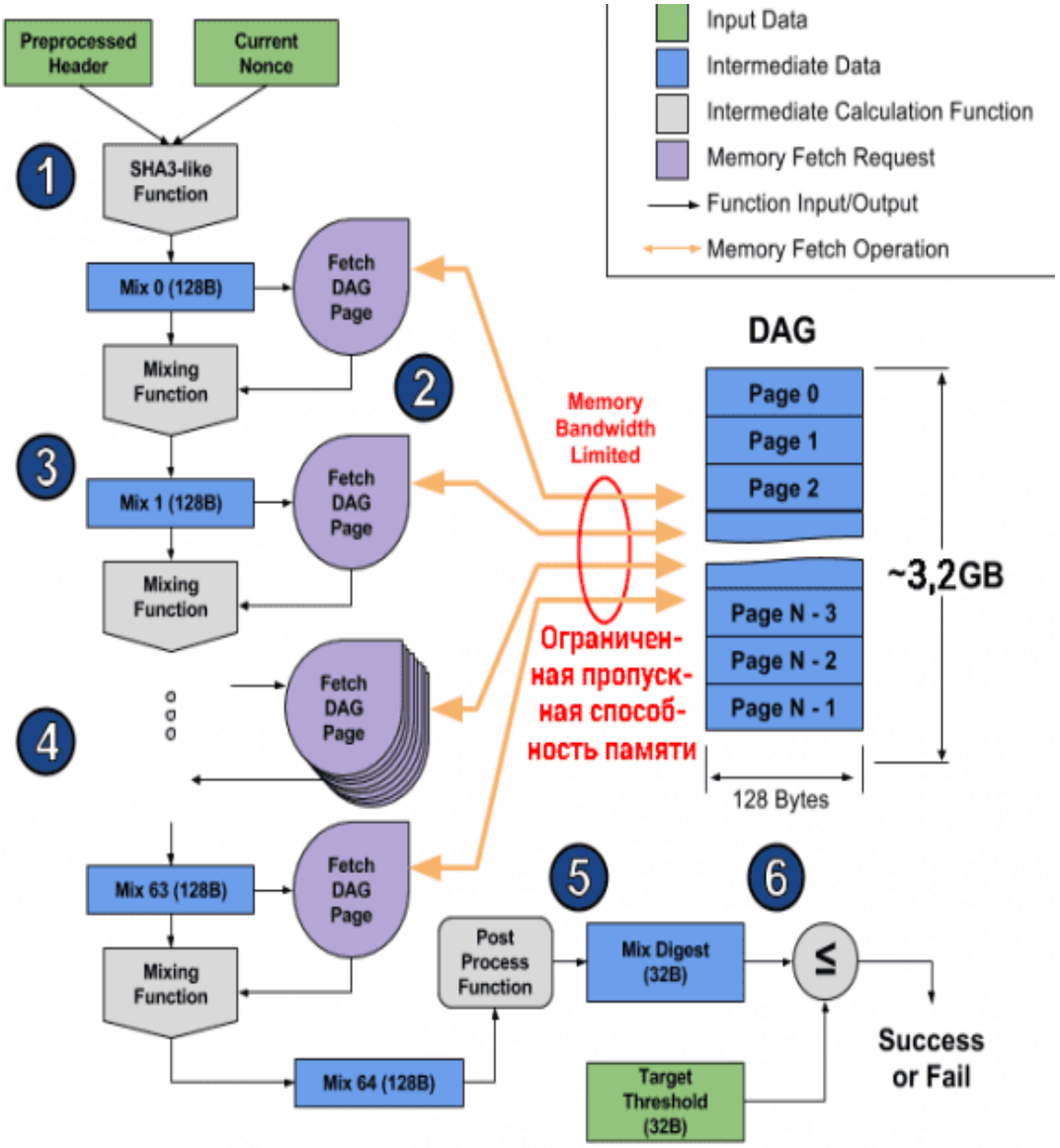


Рисунок 2.3 – Процес роботи алгоритму Ethash

2.3 Ethereum

У всесвіті Ethereum існує єдиний канонічний комп’ютер (так званий віртуальна машина Ethereum або EVM), стан якого погоджується всіма в мережі Ethereum. Кожен, хто бере участь у мережі Ethereum (кожен вузол Ethereum), зберігає копію стану цього комп’ютера. Крім того, будь-який учасник може транслювати запит на виконання довільного обчислення цього комп’ютера. Щоразу, коли такий запит транслюється, інші учасники мережі перевіряють, підтверджують і виконують ("виконують") обчислення. Це виконання спричиняє зміну стану в EVM, яка фіксується та поширюється по всій мережі.

Запити на обчислення називаються запитами транзакцій; запис усіх транзакцій і поточний стан EVM зберігається в блокчейні, який, у свою чергу, зберігається та узгоджується всіма вузлами.

Криптографічні механізми гарантують, що після того, як транзакції будуть перевірені як дійсні та додані в блокчейн, вони не можуть бути підроблені пізніше. Ті самі механізми також гарантують, що всі транзакції підписуються та виконуються з відповідними «дозволами» (ніхто не повинен мати можливість надсилати цифрові активи з облікового запису Аліси, крім самої Аліси).

Ефір (ETH) є рідною криптовалютою Ethereum. Мета ефіру — створити ринок для обчислень. Такий ринок забезпечує економічний стимул для учасників перевіряти та виконувати запити на транзакції та надавати обчислювальні ресурси мережі.

Будь-який учасник, який транслює запит на транзакцію, також повинен запропонувати деяку кількість ефіру в мережі як винагороду. Ця винагорода буде присуджена тому, хто в кінцевому підсумку виконає роботу з перевірки транзакції, її виконання, передачі її в блокчейн і трансляції в мережу.

Внутрішня валюта платформи Ethereum – це ether або ефір. Скорочене позначення – ETH. Ефіри можна використовувати не тільки в якості валютної

одиниці, а також Ефіри гарантують виконання обчислювальнь контрактів, приймаючи на себе роль такого собі «судді» для мережі блокчейн.

В Ethereum кожна з транзакцій відбувається через дію комп'ютерної програми, яка перевіряє всі умови угоди, та у випадку, якщо зобов'язання між одержувачем коштів та відправником не виконані, то і вона не буде проведена. Саме у такий спосіб, усі угоди залишаються чесними від самого початку і до самого кінця. Обійти або скасувати смарт-контракт неможливо.

Ефіріум має як переваги, так і недоліки. Серед переваг даної платформи можна виділити такі:

1) мережа Ethereum може використовиватися для передачі інформації і реєстром для її зберігання та реструктуризації.
2) платіжна система універсальна та дозволяє створювати власні цифрові види валют.
3) умови договору залишаються незмінними від моменту підписання контракту та до його завершення, якщо самі учасники не включили таку умову в угоду. При цьому всі вимоги прописуються на внутрішній мові «Solidity».
4) в Ethereum використовуються безпечні смарт-контракти, угоди укладаються без наявності третіх осіб. Система блокчейн сама здатна оцінити статуси усіх з учасників угоди щодо рівня виконання його умови та виробляє транзакцію, коли підтверджені виконання їх зобов'язань;
5) час на проведення транзакцій набагато менший, а ніж у системі біткоіни, а комісія за переказ коштів менше;
6) не існує необхідності в посередниках, що дає значну економію в засобах і часі;
7) на віртуальному гамані користувача відразу після укладення угоди буде зарезервована сума для оплати замовлення, як тільки буде підтверджено його виконання;
8) до транзакції не можливо втрутитися, оскільки вся інформація про неї додається до лацюжка блокчейн мережі;

9) Ethereum можна реалізовувати фактично використовуючи, яку завгодно мову програмування, а доступний та простий її код дозволяє фактично проводити експериментувати з ним без будь яких обмежень.

2.4 Протоколи консенсусу

Технологія блокчейн – це децентралізована та прозора мережа, в якій жоден корпоративний орган чи уряд не контролюють або підтверджують транзакції. Технологія блокчейн – це цифровий реєстр, який фіксує кожну транзакцію, що відбувається в мережі; ці транзакції дуже безпечні та незмінні в тому сенсі, що хакери не можуть маніпулювати будь-якою інформацією, а вся транзакційна діяльність у мережі блокчейн доступна для всіх.

Оскільки мережа блокчейну є автономною та децентралізованою, потрібні автоматизовані протоколи для забезпечення того, щоб вузли-учасники погоджувалися лише на дійсні транзакції. Ці протоколи створені для запобігання зловмисних дій, таких як атаки «подвійних витрат», щоб забезпечити функціональну послугу в мережі блокчейн.

Ці протоколи є не що інше, як алгоритми, які контролюють всю активність у мережі блокчейн. У цій статті ми обговоримо різні протоколи консенсусу, які використовуються в мережі блокчейн, включаючи переваги та недоліки цих протоколів, а також у яких ситуаціях вони найбільш корисні.

Ми розглянемо такі протоколи консенсусу:

1) Proof of Work (PoW);
2) Proof of Stake (PoS);
3) Delegated Proof of Stake (DPoS);
4) Transaction as Proof of Stake (TaPoS)
5) Delegated Byzantine Fault Tolerance (dBFT);
6) Casper;
7) Proof of Importance (PoI);

8) Proof of Elapsed Time (PoET);
9) Proof of Burn (PoBr).

Першим протоколом консенсусу, використаним у мережі блокчейн, був Proof of Work (PoW). Синтія Дворк і Моні Наор представили його в 1993 році, а Сатоші Накамото, творець мережі біткойн, знову представив його в 2008 році.

PoW – це протокол, який витрачає багато часу та енергії, в якому валідатори безперервно запускають дані із заголовка блоку через криптографічну хеш-функцію. Для представлення блоків використовується лінійна структура, причому кожен блок є сукупністю транзакцій.

PoW використовує особливий тип комп'ютера (ASIC) для вирішення своїх складних криптографічних проблем, що вимагає великої обчислювальної потужності. Майнери працюють над вирішенням цих складних проблем, і перша людина, яка вирішить проблему, винагороджується біткойнами.

За допомогою відкритих і закритих ключів, призначених кожному користувачеві, кожна транзакція перевіряється та підписується.

Консенсус PoW найкраще використовувати розробником, який хоче створювати програми, які вимагають безпеки або ідентифікації керування вузлами, де користувачів потрібно буде ідентифікувати, авторизувати або аутентифікувати перед доступом до служб або систем. Біткойн, одна з найбільших мереж блокчейн, прийняла цей протокол.

Однією з переваг PoW є те, що він має високу масштабованість, що означає, що він підходить для різноманітних додатків, таких як майнінг криптовалюти, транзакції перевірки або видобуток нових токенів.

Недоліком PoW є ймовірність «атаки 51%. Це момент, коли один зловмисник отримує контроль над більш ніж половиною обчислювальної потужності мережі, що робить децентралізацію неефективною.

На відміну від PoW, який перевіряє блоки за допомогою

криптографічної хеш-функції, PoS перевіряє блоки на основі ставки валідаторів (майнерів), де валідатори ставлять частину своєї криптовалюти.

Ці валідатори вибираються випадковим чином PoS на основі виділеної суми. Чим вищі ставки валідатора, тим більша ймовірність, що вони будуть обрані.

Ethereum, одна з найбільших блокчейн-мереж, прийняла протокол PoS, щоб покращити масштабованість мережі та зменшити споживання електроенергії.

PoS можна використовувати для підвищення безпеки, перевірки транзакцій і підвищення продуктивності. Деякі з криптовалют, які використовують протокол PoS, це Steem, Tezos і Gridcoin.

На відміну від консенсусного протоколу PoW, протокол PoS не потребує будь-яких спеціальних комп'ютерних чи складних криптографічних проблем для вирішення.

Крім того, PoS є більш енергоефективним, ніж PoW, що вимагає від майнерів використовувати високий рівень електроенергії для виконання своїх завдань.

Недоліком PoS є те, що зловмисник може скасувати транзакції жертви та підкупити майнерів, щоб вони підтвердили їх.

По-друге, PoS приносить користь багатим. Потужність майнінгу в Proof of Stake визначається кількістю монет, розміщених валідатором. Учасники, які ставлять більше монет, мають більше шансів бути обраними для додавання нових блоків.

Валідитори підтримують мережу блокчейн і перевіряють транзакції за цим протоколом консенсусу, і вони винагороджені за свої зусилля платою за транзакції. Цей протокол заснований на системі голосування, в якій обираються валідатори, щоб допомогти в консенсусному стані нових блоків. Ті, у кого більше монет, мають більше права голосу.

DPoS найкраще використовувати для програм на основі голосування, які вимагають високої швидкості та пропускної здатності перевірки.

Система голосування DPoS є прозорою. Якщо користувачі помічають будь-які ознаки зловмисної активності, вони можуть негайно проголосувати за видалення порушника.

DPoS частково централізований, тому ті, у кого більше монет, мають більше влади в мережі. Крім того, він уразливий для атак, оскільки лише кілька людей відповідають за підтримку мережі.

TaPoS – це консенсусний протокол, який запобігає зловмисним діям, таким як захоплення транзакції з одного блокчейну та шахрайське її повторення в іншому. Це відомо як «відтворення транзакцій».

Кожна транзакція в системі повинна містити хеш останнього заголовка блоку. Це транзакції, які відбуваються на процесорах і сприяють вирішенню завдання, що призводить до успішного майнінгу.

Цей консенсусний протокол є моделлю консенсусного протоколу PoS, отже, він має ті самі варіанти використання.

Цей протокол володіє всіма перевагами PoS, включаючи запобігання повторення транзакцій у різних ланцюгах.

Недоліком цього протоколу є те, що він, на жаль, не отримав широкого поширення.

Делегований візантійський протокол відмовостійкості був винайдений NEO, китайським децентралізованим блокчейном. У мережі блокчейн це складне поняття. Цей протокол вирішує проблему візантійських генералів у теорії ігор.

Делегована візантійська відмовостійкість ефективніша, ніж інші алгоритми, у роботі з ненадійними учасниками блокчейну. Стисле пояснення того, як працює dBFT, можна знайти тут.

Одна з найбільших бірж криптовалют Binance використовує цей консенсусний протокол.

Протокол dBFT найкраще використовувати для створення швидких та економічно ефективних додатків, які заохочують остаточність транзакції.

Однією з переваг dBFT є його швидке виконання. Генерація нового блоку в ланцюжку займає від 15 до 20 секунд. Цей протокол не має розгалужень, що забороняє кардинальні зміни в протоколі мережі, які дозволяють раніше недійсним блокам і транзакціям стати дійсними.

Ще одна перевага dBFT полягає в тому, що транзакції мають абсолютну остаточність після підтвердження, що означає, що блок не може бути розділений, а отже, транзакція не може бути відкликана або відкатана.

Недоліком цього протоколу є відсутність анонімності, оскільки делегати повинні діяти під справжніми особами, щоб бути обраними. Крім того, він не повністю децентралізований, оскільки для цього протоколу необхідний регульований блокчейн.

Casper – це модель протоколу Proof of Stake (PoS), заснована на заставі. Це означає, що валідатори повинні зробити депозит, щоб обслуговувати консенсус шляхом створення блоків.

Ethereum вибрав протокол Casper, який зміщує фокус з майнінгу на стейкинг. Ethereum Casper складається з двох спільно розроблених реалізацій в мережі Ethereum: Friendly Finality Gadget (FFG) (гібрид консенсусних протоколів PoW і PoS, зосереджених на багатоетапному переході до впровадження PoS для мережі Ethereum) і Correct-by -Конструкція (CBC), яка використовує протокол коректного побудови.

Casper все ще знаходиться на ранній стадії розробки, і йому знадобиться більше часу, перш ніж він буде готовий до широкого поширення.

Протокол Casper забезпечує кращу безпеку, завдяки чому валідатор швидко видаляється, якщо він виконує будь-яку шкідливу дію, і значна частина їхньої ставки зменшується.

По-друге, це знижує ризик атак подвійних витрат. Подвійні витрати – це можливість зловмисника скасувати транзакцію, використовуючи той самий вхід, що й інша транзакція, яка вже була перевірена в мережі.

Нарешті, Casper усуває проблему «нічого для ставок». Nothing-to-stake

є проблемою безпеки в протоколі PoS, в якій валідаторам нічого втрачати, якщо мережа розривається.

Протокол Casper найкраще використовувати для створення програм з кращою масштабованістю та безпекою, запобігаючи атаці подвійних витрат.

Протокол Casper все ще вразливий до атаки 51%. Ethereum Casper не зможе завершити блокування, якщо система перевірки Ethereum буде скомпрометована.

PoI використовується для демонстрації корисності вузлів у системі криптовалюти, що дозволяє їм створювати блоки. PoI оцінює вузли за допомогою різноманітних показників: чисті перекази, кількість наданої валюти та кластери діяльності – це кілька прикладів.

Доказ важливості також забезпечує вищий бал інвесторам, які регулярно здійснюють операції з іншими в мережі. Саме це робить PoI більш стабільним і надійним, оскільки він стимулює обіг монет, а не накопичення монет.

Протокол доказу важливості найкраще використовувати для програм, які заохочують моделювання даних і запобігають накопиченню монет і подвійним витратам.

Однією з переваг PoI є зменшення накопичення монет. Користувачі повинні будуть поставити свою валюту на учасника замовлення.

Крім того, цей протокол не вимагає спеціалізованого обладнання або високого споживання енергії чи обчислювальної потужності.

У протоколі консенсусу PoI відсоток майнінгу обмежений. Це означає, що майнери будуть майнити пропорційно кількості наявної у них криптовалюти. В результаті багаті стають ще багатшими.

Концепція PoET була представлена в 2016 році корпорацією Intel (INTC). PoET – це консенсусний протокол, який часто використовується в дозволених мережах блокчейн для визначення прав на майнінг або переможців блоків.

Дозволені мережі блокчейну вимагають підтвердження особи будь-

якого потенційного учасника, перш ніж він зможе приєднатися. PoET пропонує рівні шанси на виграш для найбільшої кількості учасників мережі та використовує випадковий вибір у стилі лотереї, щоб визначити, який вузол виграє новий блок.

Кожному учаснику мережі надається випадкова кількість часу для очікування, і той, хто першим закінчить очікування, отримує наступний блок у блокчейні.

Однією з переваг PoET є те, що він гарантує, що результати можуть бути перевірені зовнішніми учасниками та організаціями, підвищуючи прозорість мережевого консенсусу.

Крім того, цей протокол гарантує, що всі мають рівні шанси на перемогу з двома факторами. По-перше, вузли-учасники вибирають дійсно випадковий час, а не менший, який навмисно вибирають учасники для перемоги. По-друге, переможець повинен пройти період очікування.

Як і доказ роботи, PoET вимагає спеціалізованого обладнання, розробленого Intel.

Оскільки він вимагає використання спеціалізованого обладнання, протокол PoET найкраще використовувати для публічних розподілених реєстрів без дозволів і не може бути широко поширений.

Це також вимагає довіри до самого апаратного забезпечення сторонніх розробників, що суперечить одній із основних концепцій блокчейну, а саме: воно повинно бути «недовіреним».

PoBr вирішує проблеми високого споживання енергії в системі PoW. Тут майнерам дозволяється «спалювати» свій токен віртуальної валюти, щоб отримати права записувати блоки пропорційно кількості спаленої монети.

Майнери спалюють монети, надсилаючи їх за адресою, яку можна перевірити. Майнери можуть спалювати рідну валюту або валюту альтернативного ланцюга, наприклад біткойн, залежно від реалізації. В обмін вони отримують винагороду в токені рідної валюти блокчейну.

У порівнянні з Proof of Work, Proof of Burn є ефективнішим для

забезпечення справедливого розподілу монет. На відміну від майнінгу PoW, де потрібне спеціалізоване обладнання, що може призвести до посилення централізації майнінгу.

По-друге, це заохочує шахтарів брати участь у довгострокових проектах. Довгострокові інвестори менш схильні продавати чи витрачати свої монети, тому ціна монети має бути більш стабільною.

Протокол Proof of Burn найкраще використовувати для транзакцій, які генерують хеші запису, які подібні до значень хешування, які використовуються для визначення лідерів блоків PoW. Він підходить для зменшення обігу монет. Slimcoin - це одна криптовалюта, яка використовує PoBr як механізм консенсусу.

Недоліком PoBr є те, що спалювання монет не завжди є прозорим або легко піддається перевірці пересічним користувачем, що означає часті затримки в перевірці роботи майнерів.

2.5 Смарт-контракти

Смарт-контракт – це просто програма, яка працює на блокчейні Ethereum. Це набір коду (його функції) і даних (його стан), який знаходиться за певною адресою в блокчейні Ethereum.

На практиці учасники не пишуть новий код щоразу, коли хочуть запитати обчислення на EVM. Швидше, розробники додатків завантажують програми (фрагменти коду для повторного використання) у стан EVM, а користувачі роблять запити на виконання цих фрагментів коду з різними параметрами. Програми, які завантажуються в мережу і виконуються в мережі, ми називаємо смарт-контрактами.

На самому базовому рівні ви можете уявити смарт-контракт як свого роду торговий автомат: сценарій, який при виклику з певними параметрами виконує певні дії або обчислення, якщо виконуються певні умови. Наприклад, простий смарт-контракт постачальника може створити та

призначити право власності на цифровий актив, якщо абонент надсилає ефір конкретному одержувачу.

Будь-який розробник може створити смарт-контракт і зробити його загальнодоступним у мережі, використовуючи блокчейн як рівень даних, за плату, що сплачується мережі. Будь-який користувач може потім викликати смарт-контракт, щоб виконати його код, знову ж таки за плату, сплачену мережі.

Таким чином, за допомогою смарт-контрактів розробники можуть створювати та розгортати довільно складні додатки та сервіси для користувачів, такі як: ринки, фінансові інструменти, ігри тощо.

Розумні контракти вперше були запропоновані в 1990-х роках вченим-комп'ютерником і юристом на ім'я Нік Сабо. Сабо порівняв розумний контракт з торговим автоматом. Уявіть собі машину, яка продає банки газованої води за чверть. Якщо ви кладете долар у автомат і вибираєте газовані напої, апарат буде підключено до того, щоб виготовити ваш напій і 75 центів на здачу, або (якщо ваш вибір розпроданий), щоб запропонувати вам зробити інший вибір або повернути долар. Це приклад простого смарт-контракту. Подібно до того, як автомат із содою може автоматизувати продаж без посередника, розумні контракти можуть автоматизувати практично будь-який вид обміну.

На даний момент Ethereum є найпопулярнішою платформою для смарт-контрактів, але багато інших блокчейнів криптовалюти (включаючи EOS, Neo, Tezos, Tron, Polkadot і Algorand) можуть запускати їх. Смарт-контракт може бути створений і розгорнутий у блокчейні будь-хто. Їх код прозорий і підданий публічній перевірці, а це означає, що будь-яка зацікавлена сторона може точно побачити, якої логіки дотримується смарт-контракт, коли отримує цифрові активи.

Смарт-контракти є різновидом облікового запису Ethereum. Це означає, що у них є баланс і вони можуть надсилати транзакції через мережу. Однак вони не контролюються користувачем, натомість вони розгортаються в

мережі та працюють, як запрограмовано. Потім облікові записи користувачів можуть взаємодіяти зі смарт-контрактом, надсилаючи транзакції, які виконують функцію, визначену в смарт-контракті. Смарт-контракти можуть визначати правила, як звичайний контракт, і автоматично застосовувати їх за допомогою коду. Смарт-контракти не можна видалити за замовчуванням, і взаємодія з ними незворотнаПереваги смарт-контрактів:

1) Як тільки умова виконується, контракт виконується негайно. Оскільки смарт-контракти є цифровими та автоматизованими, не потрібно обробляти документи та не витрачати часу на узгодження помилок, які часто виникають внаслідок заповнення документів вручну;
2) оскільки третя сторона не залучена, а зашифровані записи трансакцій поширюються між учасниками, немає необхідності сумніватися, чи була інформація змінена для особистої вигоди;
3) записи транзакцій блокчейну зашифровані, тому їх дуже важко зламати. Більше того, оскільки кожен запис пов’язаний з попереднім і наступними записами в розподіленій книзі, хакерам доведеться змінити весь ланцюжок, щоб змінити один запис;
4) смарт-контракти усувають потребу в посередниках для обробки транзакцій і, відповідно, пов’язаних з ними затримок і зборів.

Не складно зрозуміти, що головна перевага смарт-контрактів – проведення угод без залучення третіх осіб (в звичайних умовах вони виступають гарантами виконання договору).

Увесь процес угоди можна представити у вигляді такої схеми, представленої на малюнку (рисунок 2.4), на прикладі уявної ситуації переказу грошей.

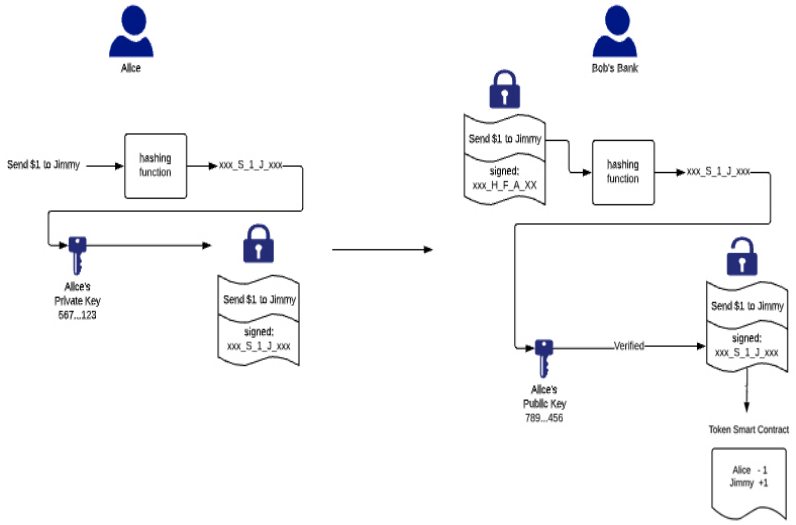


Рисунок 2.4 – Стандартна транзакція в середовищі Ethereum

2.6 Висновки до розділу 2

У другому розділі вивчено базові поняття технології блокчен. Досліженно існуючі методи хешування, які використовуються у блокчейн технології. Розглянуто та порівнянно основні криптографічні функції: Ethash та SHA256 та генерації цифрових токенів з їх використанням. Досліженно смарт-контракти та платформу Ethereum.

Аналіз проведенної роботи дозволяє зробити висновок, що блокчейн технологія є безбечною для зберігання та створеннння освітніх активів з використанням цифрових токенів.

Чи можна роботи якісь остаточні, суб'єктивні висновки базуючись на усьому, що було заначено у робі вище? На мою думку, ми ще не роглянули реалізацію додатку задля того, щоб зробити якісь остаточні.

3 ПРАКТИЧНА ЧАСТИНА. РОЗРОБКА ПРОГРАМНЫХ КОМПОНЕНТІВ ДЛЯ ТОКЕНІЗАЦІЇ ОСВІТНІХ АКТИВІВ НА ОСНОВІ СМАРТ-КОНТРАКТІВ ETHEREUM

3.1 Вибір програмних засобів для реалізації блокчейн для токенізації освітніх активів

У просторі блокчейн повно людей і підприємств, що прагнуть втілити в життя нові ідеї. Визначити, яку мову використовувати під час розробки, необхідно, щоб знайти найкращий спосіб створення цифрових маркерів.

C++ – це потужна мова програмування та вихідна мова, з якої створено біткойн. Вона об’єктно-орієнтована мова, що дозволяє йому методично зв’язувати фрагменти даних і робить його придатною мовою для створення блокчейнів. C++ допомагає розробникам керувати ресурсами, краще контролювати пам'ять і швидко обробляти взаємодії. Один із способів переконатися в цьому – за допомогою численних з’єднань, перевірки транзакцій та будівельних блоків між користувачами та майнерами.

Java є дуже популярною мовою програмування в спільноті блокчейн, завдяки своєму об'єктно-орієнтованому підходу, який ми вже бічили у C ++. Його надзвичана ціннічть для розробників та спільноти блокчейн - його мобільність. Завдяки віртуальній машині Java, Java не має обмежань у архітектурі пристрою та надзвичайо цінуеться за своєю здатністю в один і той самий час обробляти велику кількість користувачів в мережі блокчейн.

Solidity був сворений задля написання смарт-контрактів на основі Ефіріума. На початку він був створений командою розробників Ethereum, дозволяючи користувачам писати високорівневий, контрактно-орієнтований код, який потім можна було б перекласти та виконувати на мовах програмування нижчого рівня. Solidity був створений, щоб спростити використання і технології, що лежать в основі блокчейн, і, хоча він є новим,

він продовжує набувати популярність серед спільноти розробників.

Мова, популярність якої зросла в суспілстві розробників блокчейн, - це Go. Go, створений у 2007 році, та є мовою програмування, створеною розробниками Google. У галузі блокчейна Go використовується у більшості випадків задля створення децентралізованих систем. Go відомий своєю простотою в маштабуванні додатків та вікористанні, допомагаючи знаходити рішення проблеми завдяки своїй простоті у використанні.

Було прийнято рішення використовувати мову Solidity для створення смарт-контракту. Візуальне відображення було реалізовано з вікорістнням бібліотеки React js на мові програмування java script, та дляреализації взаемоії з смарт контрактом було використано бібліотеку web3 написану для java script. На рисунку (рисунок 3.1) можно побачити схему додатку.

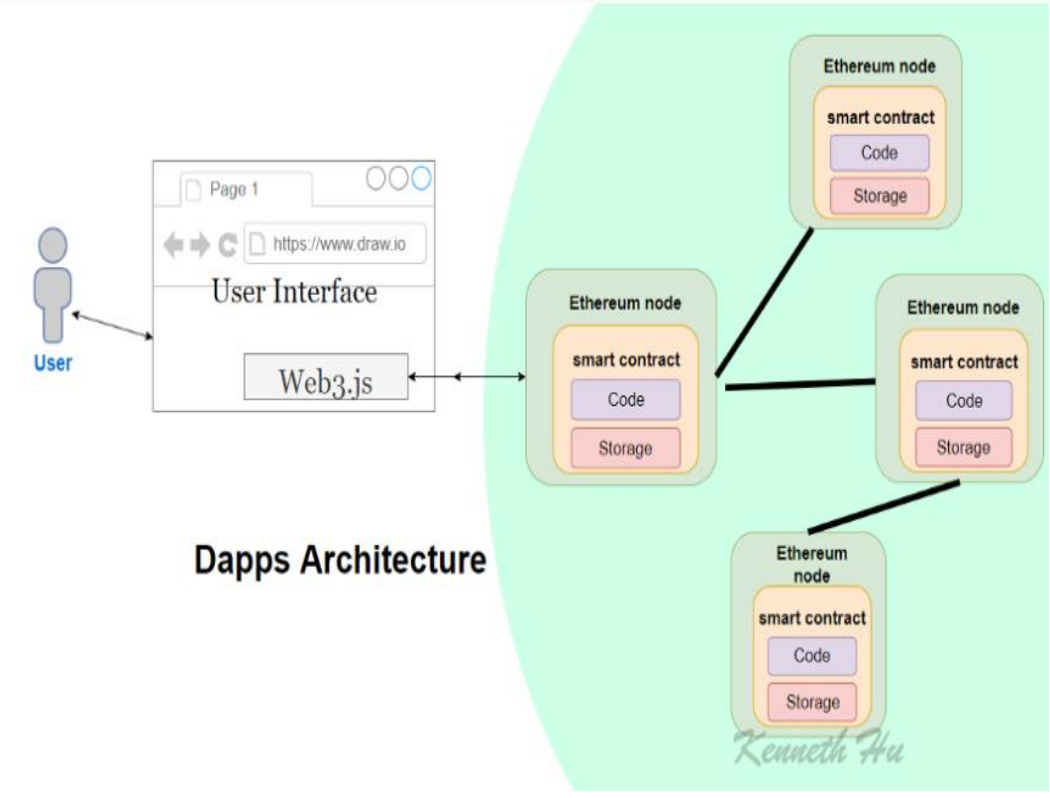


Рисунок 3.1 – Архитектура додатку

3.2 Програмна реалізація

Для створення додатку необхідно попередньо виконати налаштування середовища розробки та додаткового програмного забезпечення. Перш за все встановлюється node js і бібліотеки для звоемодії з блоченом, а також розширення для браузеру Metamask (рисунок 3.1).

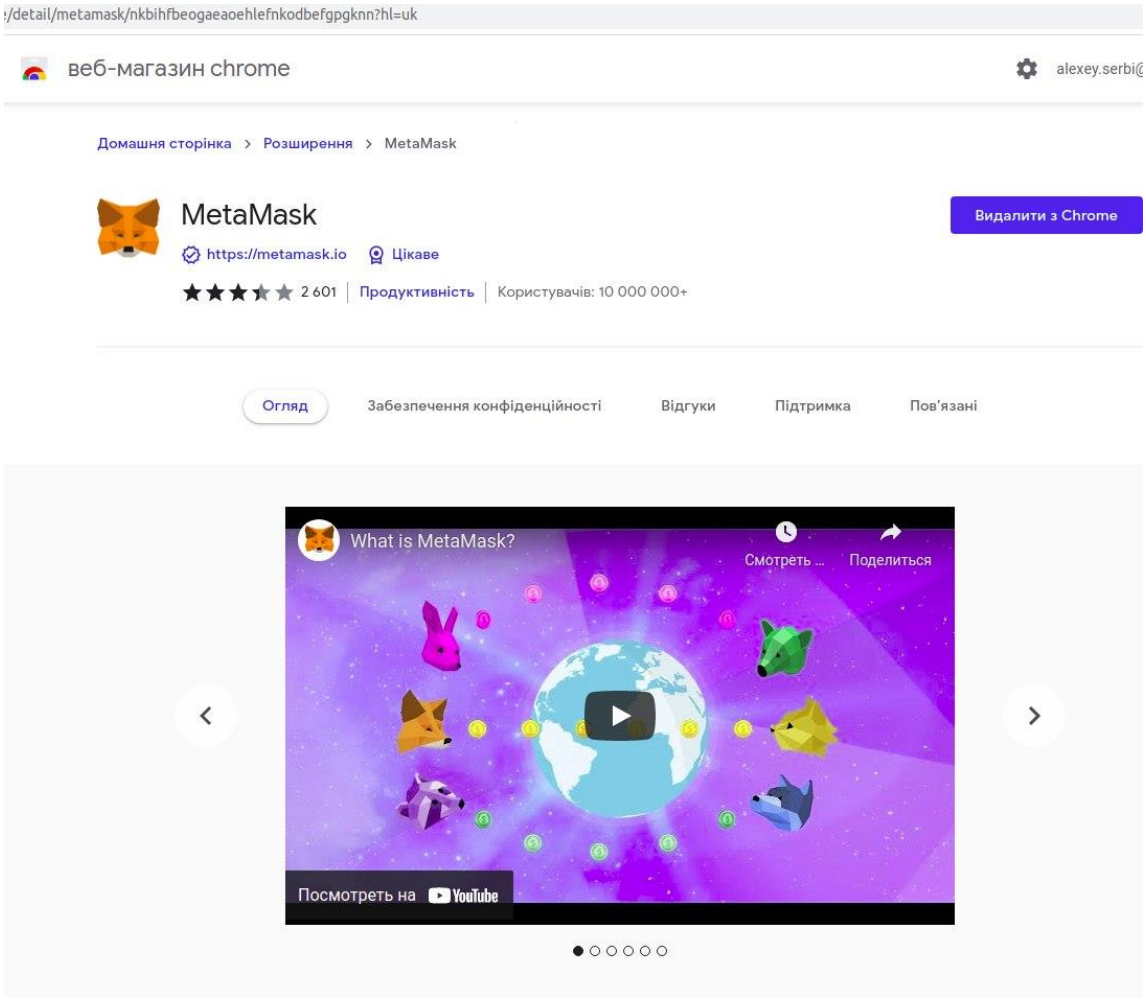


Рисунок 3.2 – Розширення Metamask

Спершу треба розробити смарт контракт для розгортання його у нашій мережі блокчейн. Створемо метод для збереження тексту у структурі з інформацією о користувачі (Лістинг 3.1).

Лістинг 3.1 – Реалізацфя смарт-сонтракту (файл Storage.sol)

```
pragma solidity >=0.4.22 <0.9.0;

contract Storage {
    struct message{
        string text;
        uint256 createdAt;
        address sender;
    }

    message[] store;

    function set(string calldata _text) public payable {
        message memory messageStruct = message({text: _text,
createdAt: block.timestamp, sender: msg.sender});

        store.push(messageStruct);
    }
}
```

Для більше зручного розгортання та компіляції коду написаного мовою Solidity було встановленно бібліотеку truffle. Для його налаштування вікорустовуеться конфігураційний файл (Лістинг 3.2).

Лістинг 3.2 – Конфігурація truffle (файл truffle-config.js)

```
module.exports = {
  /**
   * Networks define how you connect to your ethereum client and
let you set the
   * defaults web3 uses to send transactions. If you don't
specify one truffle
   * will spin up a development blockchain for you on port 9545
when you
   * run `develop` or `test`. You can ask a truffle command to
use a specific
   * network from the command line, e.g
   *
   * $ truffle test --network <network-name>
   */

  networks: {
    // Useful for testing. The `development` name is special -
```

```
truffle uses it by default
    // if it's defined here and no other network is specified at
the command line.
    // You should run a client (like ganache-cli, geth or
parity) in a separate terminal
    // tab if you use this network and you must also set the
`host`, `port` and `network_id`
    // options below to some value.
    //
    development: {
     host: "127.0.0.1",     // Localhost (default: none)
     port: 8545,            // Standard Ethereum port (default:
none)
     network_id: "*",       // Any network (default: none)
    },
    // Another network with more advanced options...
    // advanced: {
    // port: 8777,             // Custom port
    // network_id: 1342,       // Custom network
    // gas: 8500000,           // Gas sent with each transaction
(default: ~6700000)
    // gasPrice: 20000000000,  // 20 gwei (in wei) (default: 100
gwei)
    // from: <address>,        // Account to send txs from
(default: accounts[0])
    // websocket: true        // Enable EventEmitter interface
for web3 (default: false)
    // },
    // Useful for deploying to a public network.
    // NB: It's important to wrap the provider as a function.
    // ropsten: {
    // provider: () => new HDWalletProvider(mnemonic,
`https://ropsten.infura.io/v3/YOUR-PROJECT-ID`),
    // network_id: 3,       // Ropsten's id
    // gas: 5500000,        // Ropsten has a lower block limit
than mainnet
    // confirmations: 2,    // # of confs to wait between
deployments. (default: 0)
    // timeoutBlocks: 200,  // # of blocks before a deployment
times out  (minimum/default: 50)
    // skipDryRun: true     // Skip dry run before migrations?
(default: false for public nets )
    // },
    // Useful for private networks
    // private: {
    // provider: () => new HDWalletProvider(mnemonic,
`https://network.io`),
    // network_id: 2111,   // This network is yours, in the
cloud.
    // production: true    // Treats this network as if it was a
public net. (default: false)
    // }
  },
```

```
  // Set default mocha options here, use special reporters etc.
  mocha: {
    // timeout: 100000
  },

  // Configure your compilers
  compilers: {
    solc: {
      version: "0.8.12",    // Fetch exact version from solc-bin
(default: truffle's version)
      // docker: true,        // Use "0.5.1" you've installed
locally with docker (default: false)
      // settings: {          // See the solidity docs for
advice about optimization and evmVersion
      //  optimizer: {
      //    enabled: false,
      //    runs: 200
      //  },
      //  evmVersion: "byzantium"
      // }
    }
  },

  // Truffle DB is currently disabled by default; to enable it,
change enabled:
  // false to enabled: true. The default storage location can
also be
  // overridden by specifying the adapter settings, as shown in
the commented code below.
  //
  // NOTE: It is not possible to migrate your contracts to
truffle DB and you should
  // make a backup of your artifacts to a safe location before
enabling this feature.
  //
  // After you backed up your artifacts you can utilize db by
running migrate as follows:
  // $ truffle migrate --reset --compile-all
  //
  // db: {
    // enabled: false,
    // host: "127.0.0.1",
    // adapter: {
    //   name: "sqlite",
    //   settings: {
    //     directory: ".db"
    //   }
    // }
  // }
};
```

Наступним кроком нам необхідно створити додаток для взаємодії з нашим смарт-контрактом. У нашому додатку мі вікористовуемо бібліотеку React js для відображення, а для підьеднання до мережі блокчейн бібліотеку web3 js (Лістинг 3.3).

Лістинг 3.3 – Реалізацфя React js додатку (файл App.js)

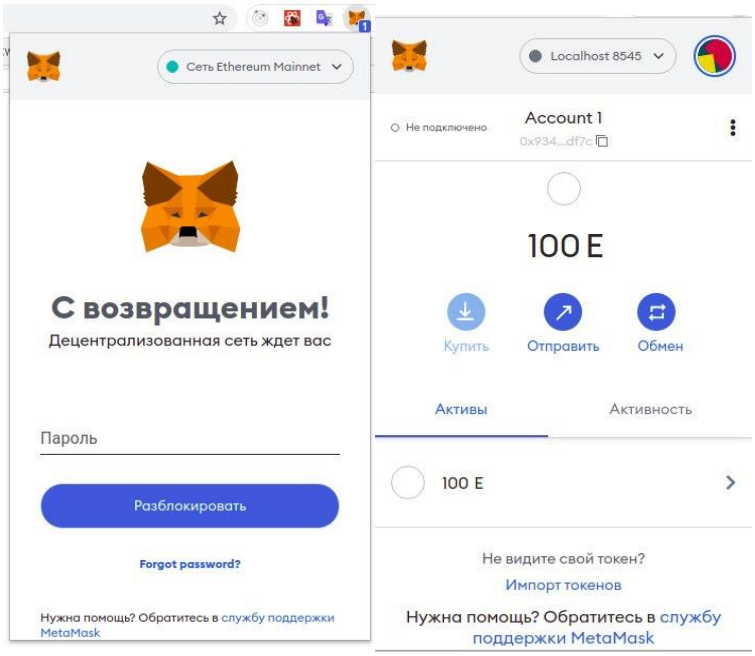
```
import './App.css';
import Web3 from 'web3/dist/web3.min.js';
import Storage from 'contracts/Storage.json';
import { useEffect, useState } from 'react';

const initWeb3 = async () => {
  if (window.ethereum) {

    return new Web3(window.ethereum);
  }
};

const getAccount = async () => {
  if (window.ethereum) {

    const accounts = await window.ethereum.request({method:
'eth_requestAccounts'});

    return accounts[0];
  }
};

const send = async (web3, text, account) => {
  const id = await web3.eth.net.getId();

  const storageContract = new web3.eth.Contract(
    Storage.abi,
    Storage.networks[id].address
  );

  console.log(await
storageContract.methods.set(text).send(({from: account})));
};


function App() {
  const [account, setAccount] = useState()
  const [web3, setWeb3] = useState()
  const [text, setText] = useState('');

  useEffect(async () => {
```

```
    setWeb3(await initWeb3());
  }, []);

  useEffect(() => {
    getAccount().then((result) => {
      setAccount(result);
    });
  }, []);

  const onChange = (e) => {
    setText(e.target.value);
  };

  const onSend = () => {
    send(web3, text, account);
  }

  return (
    <div className="App">
      <div className="fex">
        MY CURRENT ACCOUNT: {account}
        <input
          value={text}
          onChange={onChange}
        />
        <button
          onClick={onSend}
        >
          SEND
        </button>
      </div>
    </div>
  );
}

export default App;
```

Для більшь зручної взаемодії коду смарт-котратку та бібліотеки React js було додато у залежності папка bulid (рисунок 3.3) із результатом зборки смарт-котратку.

Рисунок 3.3 – Залежності додатку

Після налаштування, та разгортання можна авторизуватися у нашь додаток через Metamask (рисунок 3.4).



а) б)

Рисунок 3.4 – Авторизація у Metamask: а) сторінка авторизації; б) домашня сторінка

Задля відображення поля вводу на сторінці користувачу було

реалізовано компонент для відображення та імплементації логіки роботи поля вводу. Код реалізації можна розглянути у роботі (Лістинг 3.4). Зручний інтерфейс користувача надзвичайно важливий для користувачів.

Лістинг 3.4 – Реалізація поля вводу (файл App.js)

```
import * as React from 'react';
import { withTranslation, WithTranslation } from 'react-
i18next';
import DatePicker from 'react-datepicker';
import moment from 'moment';

import FormattedMessage from
'legacy/components/FormattedMessage';
import { Analytics } from 'legacy/utils/Analytics';
import { InputFormStyled, InputStyled, FileLabel, FileBox, Error
} from './inputForm.styles';
import "react-datepicker/dist/react-datepicker.css";

let focusedTime: number;

interface InputProps extends WithTranslation {
  changeHandler?: (value: string) => void;
  value: string;
  placeholderMessageId?: string;
  type?: 'number' | 'text' | 'date' | 'tel' | 'file' |
'autocomplete' | 'email' | 'search';
  errorStatus?: boolean;
  errorMsgId?: string;
  name?: string;
  fullWidth?: boolean;
  textTransform?: string;
  isDisabled?: boolean;
  minValue?: string;
  maxValue?: string;
  IDName?: string;
  autocompleteSuggestions?: string[];
  openSuggestionsList?: boolean;
  chooseAutocompleteOptionHandler?: (value: string) => void;
  maxLength?: number;
  style?: any;
}

const InputForm: React.SFC<InputProps> = (props: InputProps) =>
{
  const {
    placeholderMessageId,
    changeHandler,
```

```
    t,
    value,
    type,
    errorMsgId,
    errorStatus,
    name,
    fullWidth,
    textTransform,
    isDisabled,
    minValue,
    maxValue,
    IDName,
    autocompleteSuggestions,
    openSuggestionsList,
    chooseAutocompleteOptionHandler,
    maxLength,
    style
  } = props;

  const selectedDate = value && type === 'date' ? moment(value,
'DD/MM/YYYY').toDate() : null;
  const minDate = minValue && type === 'date' ? new
Date(minValue) : null;
  const maxDate = maxValue && type === 'date' ? new
Date(maxValue) : null;

  const onInputFileChange = (files: any) => {
    changeHandler && changeHandler(files);
  };

  const onInputChange = ({ target }:
React.ChangeEvent<HTMLInputElement>) => {
    changeHandler && changeHandler(target.value);
  };

  const onOptionChange = (value: string) => {
    chooseAutocompleteOptionHandler &&
chooseAutocompleteOptionHandler(value);
  };

  const onDateChange = (value: Date): any => {
    if (!value) {
      return null;
    }

    const date = moment(value).format('DD/MM/YYYY');

    changeHandler && changeHandler(date);
  };

  const placeholder = placeholderMessageId ?
t(placeholderMessageId) : '';
```

```
  const onFocus = () => {
    focusedTime = new Date().getTime();
  };

  return (
    <InputFormStyled fullWidth={fullWidth}
disabled={isDisabled}>
      {
        type === 'file' &&
        <FileBox>
          <FileLabel>
            <div>Upload</div>
            <InputStyled
              type={type}
              id={IDName}
              onChange={onInputFileChange}
              placeholder={placeholder}
              value={value}
              onFocus={onFocus}
              disabled={isDisabled}
              textTransform={textTransform}
            />
          </FileLabel>
        </FileBox>
      }
      { type !== 'date' && type !== 'file' && type !==
'autocomplete' && <InputStyled
        style={style}
        type={type}
        onChange={onInputChange}
        placeholder={placeholder}
        value={value}
        onFocus={onFocus}
        disabled={isDisabled}
        id={IDName}
        textTransform={textTransform}
        maxLength={maxLength}
      />}
      { type === 'date' &&
        <DatePicker
          id={IDName}
          selected={selectedDate}
          onChange={onDateChange}
          placeholderText={placeholder}
          disabled={isDisabled}
          minDate={minDate}
          maxDate={maxDate}
          showYearDropdown
          yearDropdownItemNumber={15}
          scrollableYearDropdown
          showMonthDropdown
          dateFormat={["dd/MM/yyyy", "dd-MM-yyyy",
"dd.MM.yyyy"]}
```

```
          className="custom-date-input"
        />
      }
      { type === 'autocomplete' && <>
        <InputStyled
          type="text"
          onChange={onInputChange}
          placeholder={placeholder}
          value={value}
          onFocus={onFocus}
          disabled={isDisabled}
          id={IDName}
          textTransform={textTransform}
          maxLength={maxLength}
        />
        { openSuggestionsList && <ul className="suggestions">
          {(autocompleteSuggestions || []).map((suggestion,
index) => {
            return (
              <li key={`${index}_${suggestion}`} onClick={() =>
onOptionChange(suggestion)}>
                {suggestion}
              </li>
            );
          })}
        </ul>
        }
      </>
      }
      {errorStatus && (
        <Error className="form-input-error-box">
          <FormattedMessage messageId={errorMsgId}
className="fs-14 red" />
        </Error>
      )}
    </InputFormStyled>
  );
};

InputForm.defaultProps = {
  changeHandler: () => null,
  placeholderMessageId: '',
  type: 'text',
  errorStatus: false,
  errorMsgId: '',
  name: '',
  fullWidth: false,
  textTransform: '',
  isDisabled: false,
  minValue: '',
  maxValue: '',
} as Partial<InputProps>;
```

```
export default withTranslation()(InputForm);

import styled from 'styled-components';

interface FormProps {
  fullWidth?: boolean;
  disabled?: boolean;
  textTransform?: string;
  style?: any;
}

export const InputFormStyled = styled('div')<FormProps>`
  position: relative;
  max-width: ${props => (props.fullWidth ? '100%' : '340px')}
  width: 100%;
  margin: 0 auto;
  margin-bottom: 20px;

  &:last-child {
    margin-bottom: 0;
  }

  .react-datepicker__input-container {
    width: inherit;
  }

  .react-datepicker-wrapper {
    width: 100%;
  }

  .custom-date-input {
    width: 100%;
    border: 1px solid #e1dfe9;
    background-color: ${props => (props.disabled ? '#e6e5e5' :
'#fff')}
    border-radius: 5px;
    font-size: 0.94em;
    height: 45px;
    padding: 0px 16px;
    color: #000000;

    &:focus {
      outline: none;
    }
  }

  .suggestions {
    position: absolute;
    z-index: 9;
    border: 1px solid #e1dfe9;
    border-top-width: 0;
    list-style: none;
    margin-top: 0;
```

```
    max-height: 143px;
    overflow-y: auto;
    padding-left: 0;
    width: 100%;
  }

  .suggestions li {
    background-color: #fff;
    padding: 12px;
  }

  .suggestion-active,
  .suggestions li:hover {
    background-color: #da291c;
    color: #fff;
    cursor: pointer;
    font-weight: 700;
  }
`;

export const InputStyled = styled('input')<FormProps>`
  width: 100%;
  border: 1px solid #e1dfe9;
  background-color: ${props => (props.disabled ? '#e6e5e5' :
'#fff')}
  border-radius: 5px;
  font-size: 0.94em;
  height: 45px;
  padding: 0px 16px;
  color: #000000;
  text-transform: ${props => (props.textTransform ?
props.textTransform : 'none')};

  &:focus {
    outline: none;
  }

  // remove X frome end of input
  &&[type='search']::-webkit-search-decoration,
  &&[type='search']::-webkit-search-cancel-button,
  &&[type='search']::-webkit-search-results-button,
  &&[type='search']::-webkit-search-results-decoration {
    display: none;
  }

  &&[type='file'] {
    display: none;
  }

  &&[type='date'] {
    &&:before {
      content: attr(placeholder);
      box-sizing: border-box;
```

```
      position: absolute;
      top: 0;
      left: 0;
      right: 0;
      bottom: 0;
      background: #fff;
      color: #787878;
      pointer-events: none;
      padding: 0px 16px;
      border: 1px solid #e1dfe9;
      border-radius: 5px;
      font-size: 0.94em;
      height: 45px;
      padding: 0px 16px;
      line-height: 3.7;
      appearance: none;

      @media (max-width: 1280px) {
        line-height: 3.7;
      }
    }

    &&:focus:before,
    &&:not([value='']):before {
      display: none;
    }

    &&::-webkit-inner-spin-button,
    &&::-webkit-outer-spin-button {
      display: none;
      -webkit-appearance: none;
    }
  }
`;

export const Error = styled('div')`
  white-space: normal;
  word-break: break-word;
  position: absolute;
  top: 45px;
  width: 100%;
  text-align: center;
`;

export const FileBox = styled('div')`
  display: flex;
  height: 40px;
  border: 1px solid #da291c;
  border-radius: 25px;
  background-color: #da291c;
  cursor: pointer;
  font-size: 1.1em;
  padding: 0 25px;
```

```
  color: #fff;
  overflow: hidden;
  align-items: center;
  justify-content: center;
`;

export const FileLabel = styled('label')`
  width: 100%;
  div {
    text-align: center;
    cursor: pointer;
  }
`;
```

Після авторизації можна перейти до створення нового блоку вікористовючи интерейс користувача додавши текст до поля вводу та натиснувши ктопку відправки (рисунок 3.5).

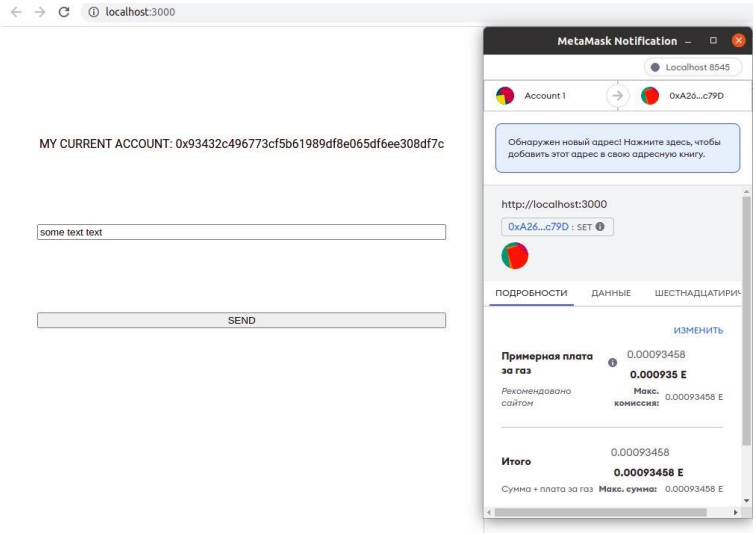


Рисунок 3.5 – Додавання нового блоку

Підтвердивши транзакцію у розширенні Metamask ми виконуемо транзакцію та знас будуть зняті кошти за проведення цієї операції (рисунок 3.6).

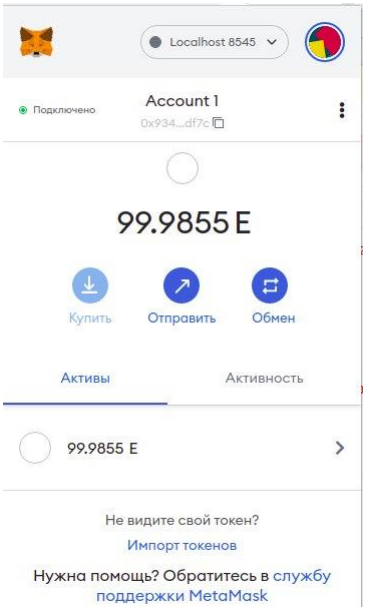


Рисунок 3.5 – Акаунт зі знятими коштами, після проведення транзакції

Та в решті решт, ми можемо побачити інформацію про виконану транзакцію у консолі (рисунок 3.6). Такі, як:

1) ціна транзакції;
2) номер блоку;
3) відправник;
4) отримувач;
5) хеш транзакції.

Рисунок 3.6 – Інформація про транзакцію

ВИСНОВКИ

Наявний розвиток цифрової економіки як результат доводить нас до фактичної реалізації цифрової трансформації всіх аспектів людського виду діяльності, включно як і виробничу, так і соціальну сферу людського життя.

Блокчейн - це один із способів розподіленого зберігання даних. Цю технологію можна використовувати для запису та відстеження будь-якого виду інформації: від медичних показників до проведення виборів.

На сьогоднішній день можна виділити три умовні сфери застосування технології блокчейн: блокчейн 1.0 - це валюта; блокчейн 2.0 - це контракти; блокчейн 3.0 - додатки, область яких не обмежана рамками фінансових транзакцій та ринків. До останньої категорії і належить використання технолгії блокчейн в освіті.

Головна відмінність блокчейна від стандартних баз даних - децентралізація. Тобто, по-перше, за цим процесом не слідкує жодний регулятор чи організацію. А по-друге, інформація не зосереджена, скажімо, на серверах в одному місці, а розподілена у величезній мережі комп'ютерів у всьому світі.

Перед попаданням блоку в ланцюжок має відбутися низка подій.

По-перше, угода має бути верифікована. На відміну від класичних транзакцій, що затверджуються банком або платіжною системою, у блокчейні операції підтверджуються мережею комп'ютерів. Як правило, мережі складаються з тисяч і навіть мільйонів машин у всьому світі.

По-друге, після того, як угода була верифікована, інформація надсилається до блоку. Там міститься дата, час, сума та цифровий підпис обох сторін.

І, нарешті, блок отримує унікальний ідентифікаційний зашифрований код, а також хеш попереднього пакета, доданого в ланцюжок. Після хешування він може бути доданий блокчейн.

Всі блоки з'являються в ланцюзі в хронологічному порядку. У ньому міститься інформація про дату та час угоди, власний зашифрований код та хеш попереднього блоку. Як тільки інформація потрапляє до блокчейну, її неможливо змінити або видалити.

І головне – прозорість мережі. Будь-яка людина може переглянути інформацію про блоки, що означає повну прозорість транзакцій. Так як у будь-який момент часу в мережі знаходяться мільйони комп'ютерів, то стає практично неможливим зламати систему і залишитися непоміченим.

У дипломній роботі описаний наявний процес отримання дипломів і, та його альтернатива, запропонована у ході роботи, інноваційний підхід із використанням цифрових токенів базуючись на технології блокчейн. Розглянут принцип роботи блокчейн-технології в освіті та механізм її використання.

У роботі пропонується декілька векторів розвитку освіти на основі технології блокчейн мереж такі, як: особова картка студента, підтвердження акредитації освітньої організації, підтвердження достовірності документів про освіту, інтелектуальна власність, ідентифікація студентів.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. The History of Blockchain Technology [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://101blockchains.com/history-of-blockchaintimeline

2. Cryptography Hash functions [Електронний ресурс]. Режим доступу до ресурсу: https://www.tutorialspoint.com/cryptography/cryptography_hash _functions.htm

3. Smart Contracts: The Ultimate Guide for the Beginners [Електронний ресурс]. – 2018. - Режим доступу до ресурсу: https://101blockchains.com /smartcontracts

4. Майнинг [Електронний ресурс]. Режим доступу до ресурсу: https://ru.bitcoinwiki.org/wiki/%D0%9C%D0%B0%D0%B9%D0%BD%D0%B8 %D0%BD%D0%B3

5. Что такое Алгоритм Консенсуса в Blockchain? [Електронний ресурс]. Режим доступу до ресурсу: https://academy.binance.com/ ru/blockchain/what-isa-blockchain-consensus-algorithm

6. . J. Golosova, A. Romanovs, “The Advantages and Disadvantages of the Blockchain Technology”, Riga, 2018

7. What’s a Peer-to-Peer (P2P) Network? [Електронний ресурс]. – 2002. - Режим доступу до ресурсу: https://www.computerworld.com/article/ 2588287/networking-peer-to-peernetwork.html

8. Blockchain Architecture Basics: Components, Structure, Benefits & Creation [Електронний ресурс]. – 2019. - Режим доступу до ресурсу: https://mlsdev.com/blog/156-how-to-build-your-own-blockchain-architecture

9. Что такое децентрализация биткоин и криптовалют [Електронний ресурс]. – 2019. - Режим доступу до ресурсу: https://prostocoin.com /blog/decentralization 66

10. BubbleTone - первая децентрализованная телекоммуникационная

экосистема. [Електронний ресурс]. – 2018. - Режим доступу до ресурсу: https://cyberway.golos.io/@ambicia/5blgxw-bubbletone-pervayadecentralizovannaya-telekommunikacionnaya-ekosistema

11. SCTelecom. IRBIS Network Decentralized telecommunications network [Електронний ресурс]. – 2019. - Режим доступу до ресурсу: https://safecalls.io/ieo/docs/SCTelecomWPv1.2.pdf?

12. Blockchain for telecom roaming, fraud user identification, and overage management. [Електронний ресурс]. – 2018. - Режим доступу до ресурсу: https://developer.ibm.com/technologies/blockchain/patterns/blockchain-fortelecom-roaming-fraud-and-overage-management/