Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)
Кафедра Інформатики
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти другий (магістерський)

ДОСЛІДЖЕННЯ ТА РЕАЛІЗАЦІЯ МЕТОДУ ЦИФРОВОЇ ІДЕНТИФІКАЦІЇ З ВИКОРИСТАННЯМ БЛОКЧЕЙНУ
(тема)

Виконав:
студент 2 курсу, групи ІНФм-22-2
Шахматено Д.В.
(прізвище, ініціали)

Спеціальності 122 Комп'ютерні науки
(код і повна назва спеціальності)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

Керівник доц. Кобилін О.А.
(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри _____ Кобилін О.А.
(підпис) (прізвище, ініціали)

2024 р.

Харківський національний університет радіоелектроніки

Факультет Інформаційно-аналітичних технологій та менеджменту
(повна назва)

Кафедра Інформатики
(повна назва)

Рівень вищої освіти другий (магістерський)

Спеціальність 122 Комп'ютерні науки
(код і повна назва)

Тип програми освітньо-професійна

Освітня програма Інформатика
(повна назва освітньої програми)

ЗАТВЕРДЖУЮ:
Зав. кафедри _____
(підпис)
«____»_____ 2024 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові Шахматенку Дмитру Володимировичу
(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження та реалізація методу цифрової ідентифікації з використанням блокчейну

затверджена наказом по університету від 3 листопада 2023 року № 1280Ст

2. Термін подання студентом роботи до екзаменаційної комісії 30 грудня 2023 р.

3. Вихідні дані до роботи блокчейн технології та сфери їх застосування, математичні моделі використання газу та алгоритмів циклу, програмні застосунки екосистеми Ethereum, застосування для цифрової ідентифікації.

4. Перелік питань, що потрібно опрацювати в роботі
1. Аналіз технології блокчейн.
2. Смарт-контракти в мережі ethereum та технології її реалізації.
3. Розгортання локального блокчейну та інтеграція з застосунком.

5. Перелік графічного матеріалу із зазначенням креслеників, схем, плакатів, комп'ютерних ілюстрацій (п.5 включається до завдання за рішенням випускової кафедри) актуальність теми, застосування блокчейн технологій в різних сферах, постановка задачі, тестові зображення.

6. Консультанти розділів роботи (п.6 включається до завдання 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

№ з/п	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	03.11.2023	
2	Аналіз завдання, підбір літератури	03.11.23-04.11.23	
3	Аналіз літератури з досліджуваної проблеми	05.11.23-06.11.23	
4	Аналіз технічних засобів	06.11.23-09.11.23	
5	Розробка методу	09.11.23-11.11.23	
6	Програмна реалізація	11.11.23-16.11.23	
7	Оформлення пояснювальної записки	16.11.23-20.11.23	
8	Перевірка на плагіат	28.11.2023	
9	Рецензування	02.12.2023	
10	Підготовка презентації та доповіді	15.12.2023	
11	Занесення роботи в електронний архів	09.01.2024	
12	Попередній захист кваліфікаційної роботи	09.01.2024	

Дата видачі завдання 3 листопада 2023 р.

Студент _____
(підпис)

Керівник роботи _____ доц. Кобилін О.А.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ/ABSTRACT

Пояснювальна записка до кваліфікаційної роботи: 65 с., 2 табл., 30 рис., 49 джерел.

ІДЕНТИФІКАЦІЯ, BLOCKHAIN, СМАРТ-КОНТРАКТ, ETHEREUM, GANACHE, TRUFFLE, JAVASCRIPT, ВЕБЗАСТОСУНОК, ETHEREUM.

Об’єктом дослідження є цифрова ідентифікація на основі блокчейн технологій.

Метою дослідження є реалізація та дослідження вебінтерфейсу для цифрової ідентифікації за допомогою блокчейн технологій Ethereum, та локальної тестової блокчейн мережі Ganache.

Інтегрувано методи розробки DApp з технологією блокчейн, щоб підвищити безпеку та надійність цифрової верифікації ідентичності. Впровадивши смарт-контракти та використавши платформу Ethereum разом з локальним блокчейном Ganache, було створено систему, яка не лише безпечно зберігає ідентифікаційні дані, але й дозволяє здійснювати ефективні транзакції з можливістю їх перевірки..

В результаті дослідження була успішно реалізована програмна реалізація системи цифрової ідентифікації особи з використанням технології блокчейн.

IDENTIFICATION, BLOCKCHAIN, SMART CONTRACT, ETHEREUM, GANACHE, TRUFFLE, JAVASCRIPT, WEB APPLICATION, ETHEREUM.

The object of research is digital identification based on blockchain technologies.

The purpose of the research is to implement and study the web interface for digital identification using Ethereum blockchain technology and the local test blockchain network Ganache.

DApp development methods are integrated with blockchain technology to improve the security and reliability of digital identity verification. By implementing smart contracts and using the Ethereum platform together with the local Ganache blockchain, a system was created that not only securely stores identity data but also allows for efficient and verifiable transactions.

As a result of the research, a software implementation of a digital identity system using blockchain technology was successfully implemented.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

KBA – Knowledge-Based Authentication (аутентифікація на основі знань)

EVM – Ethereum Virtual Machine (віртуальна машина Ethereum)

DApp – Decentralized Application (децентралізований застосунок)

PoW – Proof of Work (доказ виконання робот)

PoS – Proof of Stake (доказ частки)

DeFi – Decentralized Finance (децентралізовані фінансові сервіси)

2Fa – Two-Factor Authentication (двофакторна аутентифікація)

JSON – JavaScript Object Notation (нотація об’єктів)

RPC – Remote Procedure Call (віддалений виклик процедури)

API – Application Programming Interface (програмний інтерфейс)

HTML – HyperText Markup Language (мова гіпертекстової розмітки)

CSS – Cascading Style Sheets (каскадна таблиця стилів)

ВСТУП

У сучасному цифровому ландшафті тема безпечної цифрової ідентифікації є однією із найбільш актуальних та важливих задач. Персональні дані стали безцінним активом, що підвищує важливість безпеки та конфіденційності даних на тлі зростаючих загроз. У цьому контексті технологія блокчейн постає як трансформаційний інструмент з потенціалом революціонізувати методології цифрової ідентифікації. Такий розвиток подій обумовлює необхідність подальших досліджень і співпраці в цій галузі.

Блокчейн, як технологія, спочатку був задуманий як фундаментальна інфраструктура для криптовалют. Проте з часом він розширив своє застосування та охопив різні галузі. Фінанси, охорона здоров’я, управління ланцюгами поставок та державні послуги активно розглядають можливості використання блокчейну для підвищення прозорості, безпеки та довіри. Ця технологія надає унікальний шлях для створення децентралізованих систем ідентифікації, де дані користувачів захищені та контрольовані користувачами, забезпечуючи високий рівень безпеки та конфіденційності. Розгорнуті дослідження та розвиток цього підходу важливі для подальшого забезпечення цифрової безпеки та приватності [1–7].

В останні роки технологія блокчейн стала революційною інновацією з трансформаційним потенціалом у різних сферах, вона являє собою революційну зміну парадигми зберігання даних – децентралізовану базу даних, що складається з безперервного ланцюжка взаємопов’язаних блоків. Її відмінними рисами є надійність, незмінність і безпека. З часом вона знайшла застосування в цифровому мережевому зберіганні даних, верифікації особистості, захисті авторських прав, системах голосування та інших сферах [8, 9].

Актуальність дослідження полягає у всебічному дослідженні та подальшому впровадженні методології цифрової ідентифікації, використовуючи можливості технології блокчейн. Головна мета полягає в

розробці інноваційної системи, яка має високий рівень безпеки та конфіденційності для користувачів, але й надає їм більшу автономію в контролі над своїми даними [10].

В рамках дослідження буде проведено поглиблене вивчення технології блокчейн, розробки смарт-контрактів, що мають велике значення для полегшення процесу ідентифікації, та їх безперешкодної інтеграції у вебзастосунку. Крім того, дослідження включатимуть оцінку безпеки та продуктивності системи, що розробляється, а також оцінку її ефективності та застосовності.

Результатом даної роботи є система цифрової ідентифікації, яка пропонує альтернативу традиційним методам. Мета підвищити безпеку даних, зменшити ризики шахрайства та надати користувачам більший контроль над своєю особистою інформацією.

Дане дослідження підкреслює важливість розвитку методів цифрової ідентифікації в сучасному технологічно розвиненому світі, наголошуючи на зростаючій актуальності технології блокчейн у різних сферах, окрім її початкового застосування у криптовалютах.

1 ОГЛЯД ТА АНАЛІЗ ТЕХНОЛОГІЇ БЛОКЧЕЙН

1.1 Визначення та історія технології блокчейн

Технологія блокчейн – це децентралізована і розподілена система реєстрів, призначена для безпечного запису транзакцій через мережу комп’ютерів. Вона базується на принципах прозорості, незмінності та безпеки даних. У блокчейні інформація згрупована в блоки, кожен з яких містить кілька транзакцій. Ці блоки пов’язані між собою в хронологічному порядку, утворюючи ланцюжок блоків. Транзакції в блокчейні захищені за допомогою криптографічного хешування, що забезпечує цілісність даних [11-15].

Децентралізація, притаманна блокчейну, усуває потребу в посередниках, таких як банки або центральні органи влади, для перевірки транзакцій. Натомість вона покладається на мережу вузлів (комп’ютерів), які досягають консенсусу щодо підтвердження та запису транзакцій. Така децентралізація і прозорість роблять технологію блокчейн особливо привабливою для застосування за межами криптовалют [16-20].

Історія технології блокчейн бере свій початок з концепції децентралізованої цифрової валюти, відомої як біткойн. У 2008 році особа або група осіб під псевдонімом «Сатоші Накамото» опублікували революційний документ під назвою «Біткоїн: пірингова система електронних грошей». Цей документ представив концепцію блокчейну як базової технології, що лежить в основі біткоїна.

Перший блокчейн біткоїна з’явився в січні 2009 року, ознаменувавши народження першої криптовалюти. Спочатку її основною метою було полегшити обмін цифровими активами між користувачами в обхід традиційних фінансових посередників.

Основна інновація блокчейну полягала в його здатності створювати децентралізовану бухгалтерську книгу, вирішуючи давню проблему подвійних витрат у цифрових валютах. Записуючи кожну транзакцію у

захищений від підробки, прозорий і хронологічний спосіб, технологія блокчейн забезпечила революційне рішення для ненадійних однорангових цифрових обмінів [21, 22].

З часом технологія блокчейн отримала визнання завдяки своїм потенційним можливостям застосування за межами криптовалют. Інноватори та розробники почали досліджувати її корисність у різних галузях, що призвело до створення альтернативних блокчейнів, таких як Ethereum. Ethereum запровадив концепцію смарт-контрактів, уможлививши програмовані та самодостатні угоди на блокчейні.

Здатність блокчейну забезпечувати безпечне, прозоре та захищене від несанкціонованого доступу зберігання даних призвела до його застосування в управлінні ланцюгами поставок, охороні здоров’я, фінансах та багатьох інших галузях. Оскільки технологія блокчейн продовжує розвиватися, вона обіцяє стати трансформаційною силою у сфері технологій та бізнесу [23, 24].

За своєю суттю, блокчейн – це розподілений реєстр, який складається з ланцюжка блоків, кожен з яких містить набір транзакцій. Ці блоки пов'язані між собою криптографічно, створюючи безперервний і незмінний ланцюжок даних. Щоб зрозуміти основи блокчейну, важливо розуміти наступні ключові компоненти:

– децентралізація, на відміну від традиційних централізованих систем, де центральний орган контролює дані та транзакції, блокчейн працює в децентралізованій мережі вузлів (комп’ютерів). Кожен вузол має копію всього блокчейну, а транзакції підтверджуються за допомогою механізму консенсусу;

– криптографія, криптографічні методи відіграють фундаментальну роль у захисті даних в блокчейні. Транзакції записуються таким чином, щоб забезпечити їх автентичність і цілісність. Хеш-функції та цифрові підписи є одними з криптографічних інструментів, що використовуються в технології блокчейн;

– незмінність, після того, як дані записані в блокчейні, їх стає надзвичайно складно змінити або видалити. Кожен блок містить посилання на попередній блок через криптографічний хеш, створюючи ланцюжок, де зміна одного блоку вимагатиме зміни всіх наступних блоків – непрактичне завдання через розподілену природу мережі;

– механізми консенсусу, мережі блокчейн використовують механізми консенсусу для підтвердження транзакцій і підтримки цілісності реєстру. Прикладами таких механізмів є Proof of Work (PoW) і Proof of Stake (PoS), кожен з яких має власний підхід до досягнення консенсусу;

– прозорість, транзакції в публічному блокчейні видно всім учасникам мережі. Така прозорість сприяє підвищенню підзвітності та довіри між користувачами.

1.2 Принципи технології блокчейн

Одним з основних принципів блокчейну є прозорість. Кожна транзакція, яка записується в блокчейні, доступна для перегляду всіма учасниками мережі. Ця прозорість сприяє відкритості та відповідальності, знижуючи можливості для шахрайства та маніпуляцій [25–28].

В основі трансформаційного потенціалу блокчейну лежать кілька фундаментальних принципів, які забезпечують його функціональність і безпеку.

Незмінність тісно пов’язана з прозорістю. Як тільки дані внесені в блокчейн, їх стає майже неможливо змінити або видалити. Кожен блок містить посилання на попередній блок через криптографічний хеш, створюючи хронологічний і незмінний ланцюжок даних. Будь-яка спроба змінити один блок вимагатиме зміни всіх наступних блоків, що є практично неможливим з обчислювальної точки зору через децентралізовану та розподілену природу мережі (рис. 1.1).

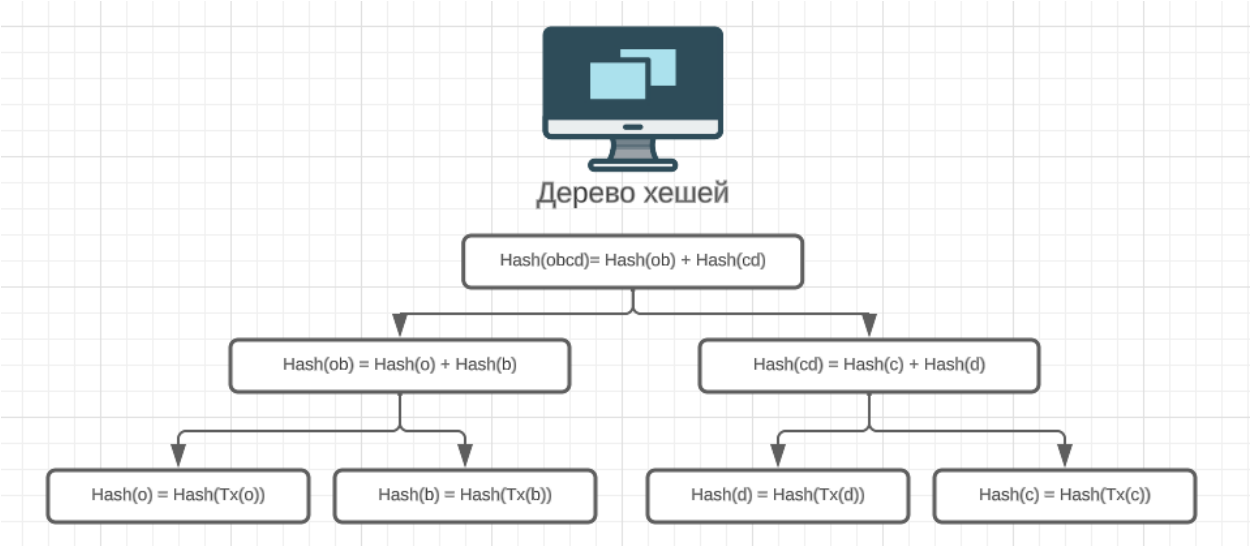

Рисунок 1.1 – Візуалізація хешування в блокчейні

Децентралізація є основним принципом технології блокчейн. Замість того, щоб покладатися на центральний орган або посередника для підтвердження транзакцій, блокчейн працює на мережі вузлів (комп'ютерів), які досягають консенсусу за допомогою алгоритмів. Така децентралізована природа зменшує ризик єдиної точки відмови і підвищує безпеку (рис. 1.2).

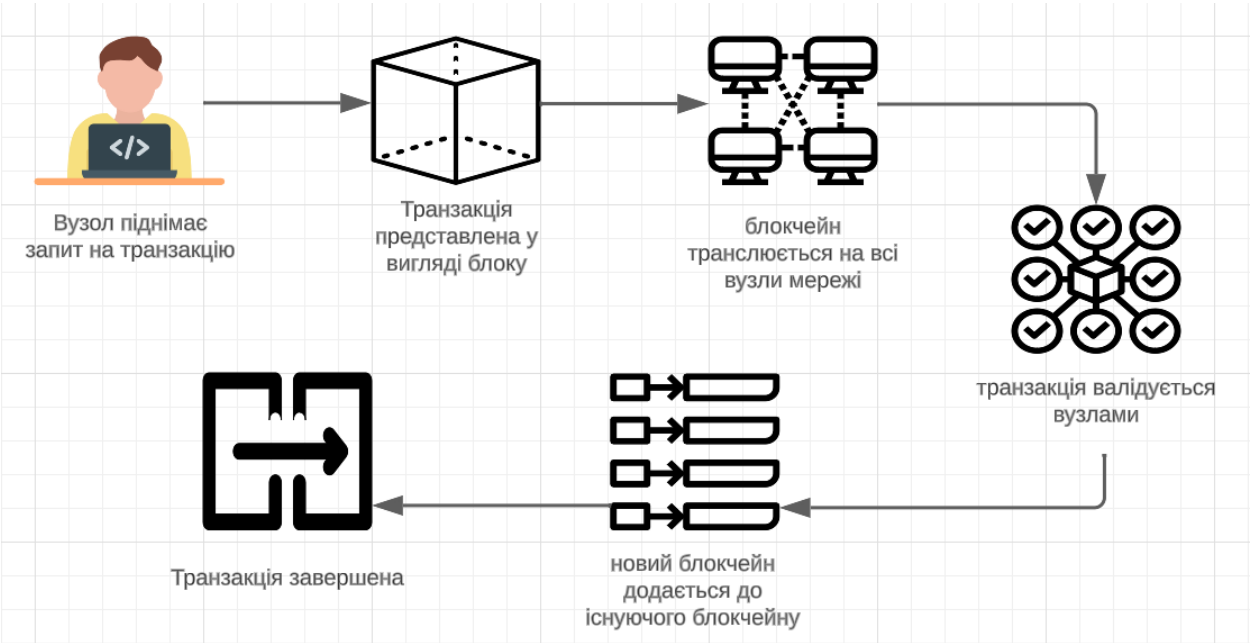

Рисунок 1.2 – Візуалізація децентралізованої мережі вузлів, що підтверджують транзакції

Транзакції в блокчейні захищені за допомогою криптографічного хешування. Кожна транзакція хешується і пов’язується з попередньою транзакцією, створюючи ланцюжок захищеної інформації [29]. Крім того, механізми консенсусу, такі як Proof of Work (PoW) або Proof of Stake (PoS), забезпечують додаткову безпеку, вимагаючи від учасників розв’язання складних математичних задач або внесення криптовалюти в якості застави для підтвердження транзакцій (рис. 1.3).

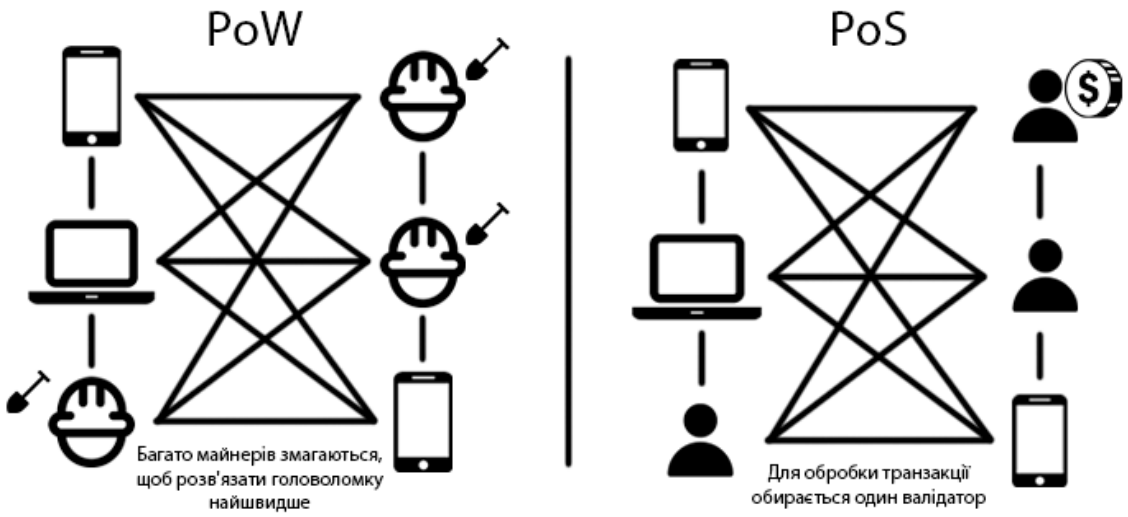

Рисунок 1.3 – Візуалізація PoW і PoS принципів захисту мережевих систем

Технологія блокчейн за своєю суттю забезпечує цілісність даних. Після того, як транзакція підтверджена і додана до блокчейну, вона не може бути змінена без консенсусу більшості учасників мережі. Така незмінність сприяє довірі між користувачами, оскільки вони можуть покладатися на цілісність записаних даних.

Ключовою концепцією блокчейну є бездовіра, що означає, що користувачам не потрібно довіряти центральному органу [30, 31]. Довіра замінюється криптографічними доказами і механізмами консенсусу, які забезпечують математичну гарантію дійсності транзакцій. Таке середовище без довіри особливо цінне в застосунках, де довіра традиційно викликає занепокоєння.

Масштабованість – це здатність блокчейну обробляти все більшу кількість транзакцій без шкоди для продуктивності. Масштабованість є предметом активних досліджень і розробок, а такі рішення, як шардінг і протоколи другого рівня, спрямовані на підвищення продуктивності блокчейну [32–35].

Технологія блокчейн є універсальною і може бути інтегрована в різні системи та платформи, що сприяє їхній інтероперабельності. Така адаптивність дає змогу застосовувати її не лише у фінансовій сфері, а й в інших сферах бізнесу та життя людини.

1.3 Типи блокчейнів та їх застосування

Публічні блокчейни – це відкриті мережі, доступні будь-кому, що дозволяють брати участь і перевіряти транзакції децентралізованою спільнотою вузлів. Яскравими прикладами є Bitcoin та Ethereum. Біткоїн та Ефіріум є яскравими прикладами публічних блокчейнів. Публічні блокчейни в першу чергу відомі завдяки криптовалютним транзакціям, які полегшують однорангові цифрові обміни. Крім того, вони відіграють ключову роль у децентралізованих фінансах (DeFi), де користувачі можуть брати участь у кредитуванні, запозиченнях і торгівлі активами без посередників. Публічні блокчейни також сприяють створенню децентралізованих застосунків (DApps) у безлічі секторів, таких як ігри, управління ланцюжками поставок і соціальні мережі. Прозорість і незмінність публічних блокчейнів роблять їх придатними для застосувань, де довіра, безпека і децентралізація мають першорядне значення [36, 37]. Найвідомішим прикладом використання є криптовалюти, які дозволяють здійснювати однорангові цифрові транзакції без посередників на кшталт банків (рис. 1.4).

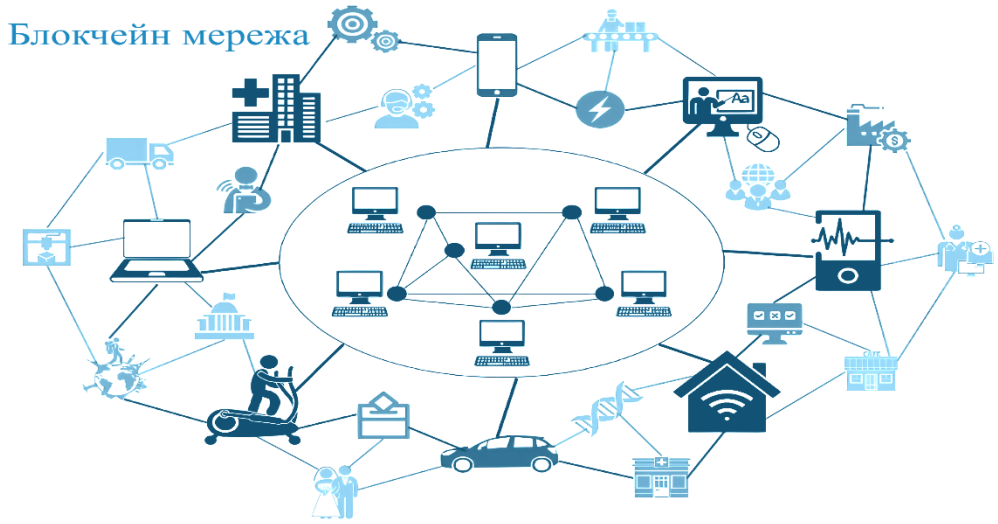

Рисунок 1.4 – Діаграма децентралізованої природи публічних блокчейнів

Приватні блокчейни – це мережі з обмеженим доступом, які зазвичай використовуються організаціями, консорціумами або окремими групами з певними правами доступу. Вони забезпечують підвищений рівень контролю над учасниками та доступом до даних порівняно з публічними блокчейнами. Приватні блокчейни користуються популярністю в галузях, що вимагають суворої конфіденційності, таких як охорона здоров'я та фінанси, також вони знаходять застосування в таких галузях, як управління ланцюгами поставок, де консорціум компаній може захотіти безпечно обмінюватися даними, не виставляючи їх на загальний огляд. Ці мережі можуть підтримувати переваги блокчейну, такі як прозорість і цілісність даних, зберігаючи при цьому контроль над учасниками (рис. 1.5).

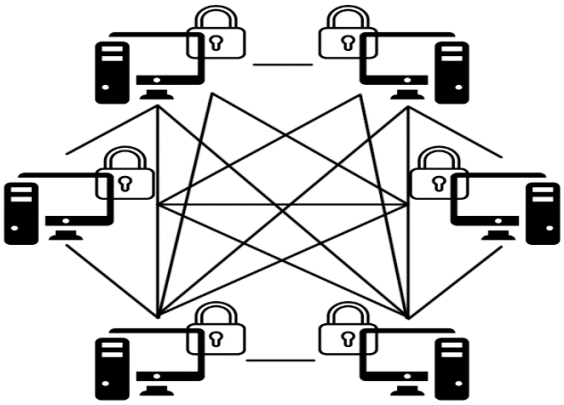

Рисунок 1.5 – Представлення приватної мережі блокчейн

Консорціумні блокчейни – це гібрид між публічними і приватними блокчейнами, який управляється консорціумом або групою організацій, що спільно керують мережею. Ці блокчейни поєднують елементи децентралізації з дозволеним доступом. Консорціумна модель блокчейну добре підходить для галузей, де багато зацікавлених сторін повинні співпрацювати, зберігаючи при цьому певний рівень контролю. Приклади включають фінансові консорціуми, які досліджують блокчейн для транскордонних платежів і торгового фінансування, також наприклад, у сфері охорони здоров'я консорціумний блокчейн може забезпечити безпечне зберігання даних про пацієнтів і доступ до них лише уповноваженим медичним працівникам, зберігаючи при цьому публічний доступ до певних дослідницьких даних (рис. 1.6).

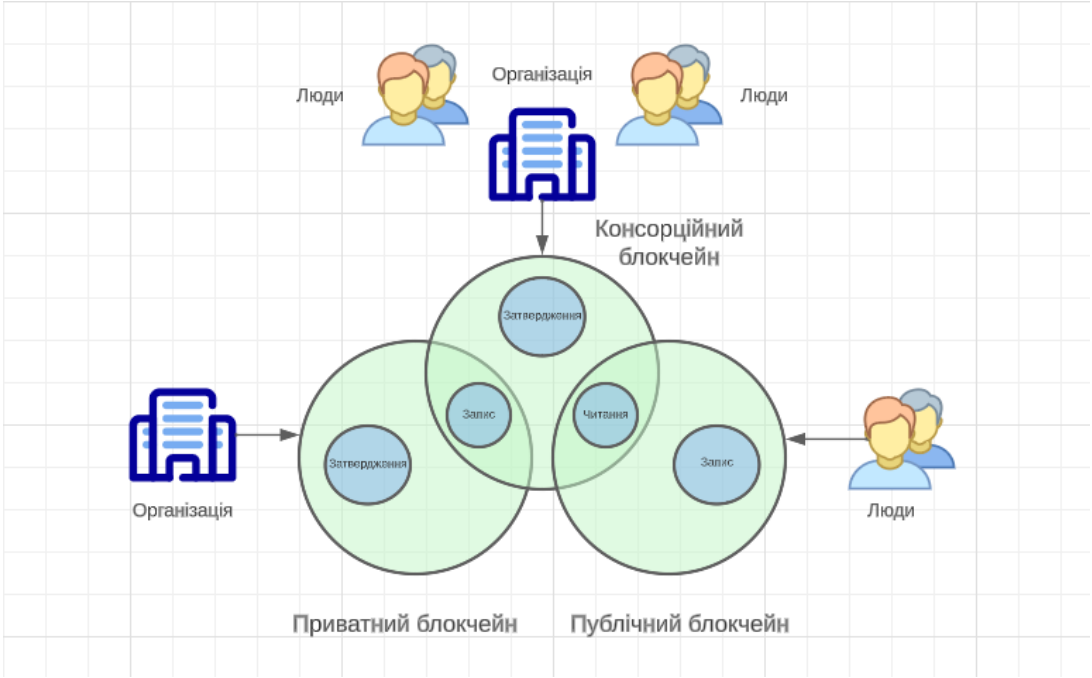

Рисунок 1.6 – Представлення консорційної мережі

Технологія кросчейн – це новий тренд, який дозволяє різним блокчейн-мережам безпечно спілкуватися та обмінюватися даними. Вона спрямована на подолання ізольованості окремих блокчейнів і відкриття нових можливостей для взаємодії. Крос-ланцюгові рішення мають потенціал у різних секторах, від децентралізованих фінансів (DeFi) до управління ланцюгами поставок,

дозволяючи безперешкодно переміщувати дані та активи між різними блокчейн-екосистемами. Основні цілі технології крос-ланцюгів включають в себе Підвищення функціональної сумісності, сприяння передачі активів, забезпечення безпеки, сприяння співпраці. Технологія міжмережевої взаємодії заохочує співпрацю між блокчейн-проектами та спільнотами. Уможливлюючи обмін даними та активами, вона прокладає шлях до інноваційних міжмережевих застосунків і сервісів (рис. 1.7).

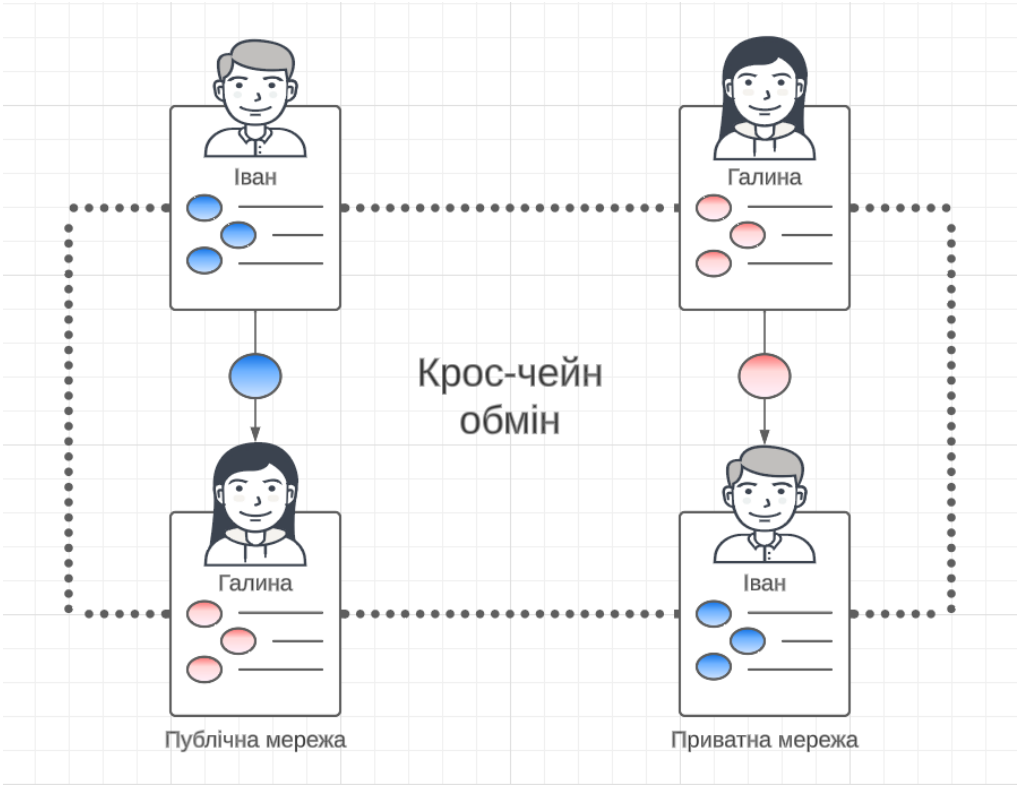

Рисунок 1.7 – Представлення кросчейн технології

Глобально блокчейни можна класифікувати як дозволені (з обмеженим доступом) або недозволені (з відкритим доступом). Дозволені блокчейни підходять для ситуацій, де довіра та перевірка особи є першочерговими, тоді як бездозволені блокчейни ставлять на перше місце децентралізацію та інклюзивність.

Майбутнє технології блокчейн полягає в досягненні більшої функціональної сумісності між різними мережами блокчейн. Інтероперабельні блокчейни можуть підвищити ефективність та співпрацю в різних галузях,

уможливлюючи передачу даних та активів без необхідності використання громіздких посередників.

Оскільки технологія блокчейн продовжує розвиватися, очікується, що вона знайде застосування в нових галузях, таких як Інтернет речей (IoT), де безпечний і децентралізований обмін даними є надзвичайно важливим. Крім того, блокчейн відіграватиме ключову роль у перевірці цифрової ідентичності, забезпечуючи безпечну та орієнтовану на користувача автентифікацію у все більш цифровому світі.

Розуміння різних типів блокчейнів та їхніх застосувань має вирішальне значення для вибору найбільш підходящого фреймворку блокчейну для конкретних випадків використання. Універсальність блокчейну охоплює численні галузі, пропонуючи індивідуальні рішення для широкого спектру потреб.

1.4 Сучасні методи онлайн ідентифікації та їх ключові виклики

Аутентифікація за допомогою пароля – найпоширеніший метод підтвердження особи в Інтернеті. Користувачі вводять комбінацію символів , щоб отримати доступ до свого облікового запису. Основними проблемами парольної автентифікації є слабкий вибір паролів, складність запам’ятовування декількох паролів для різних облікових записів і ризик витоку даних, який може призвести до розголошення збережених паролів (рис. 1.8). Однак цей метод має ряд обмежень:

– вразливості безпеки, паролі можуть бути зламані, оскільки їх можна вгадати, викрасти або зламати різними способами, включаючи атаки брут форс та фішинг;

– навантаження на користувача, керування численними складними паролями для кількох облікових записів може бути складним завданням і часто призводить до слабкої парольної практики;

– скидання паролів, забуті паролі призводять до незручностей і часто вимагають важких процедур відновлення.

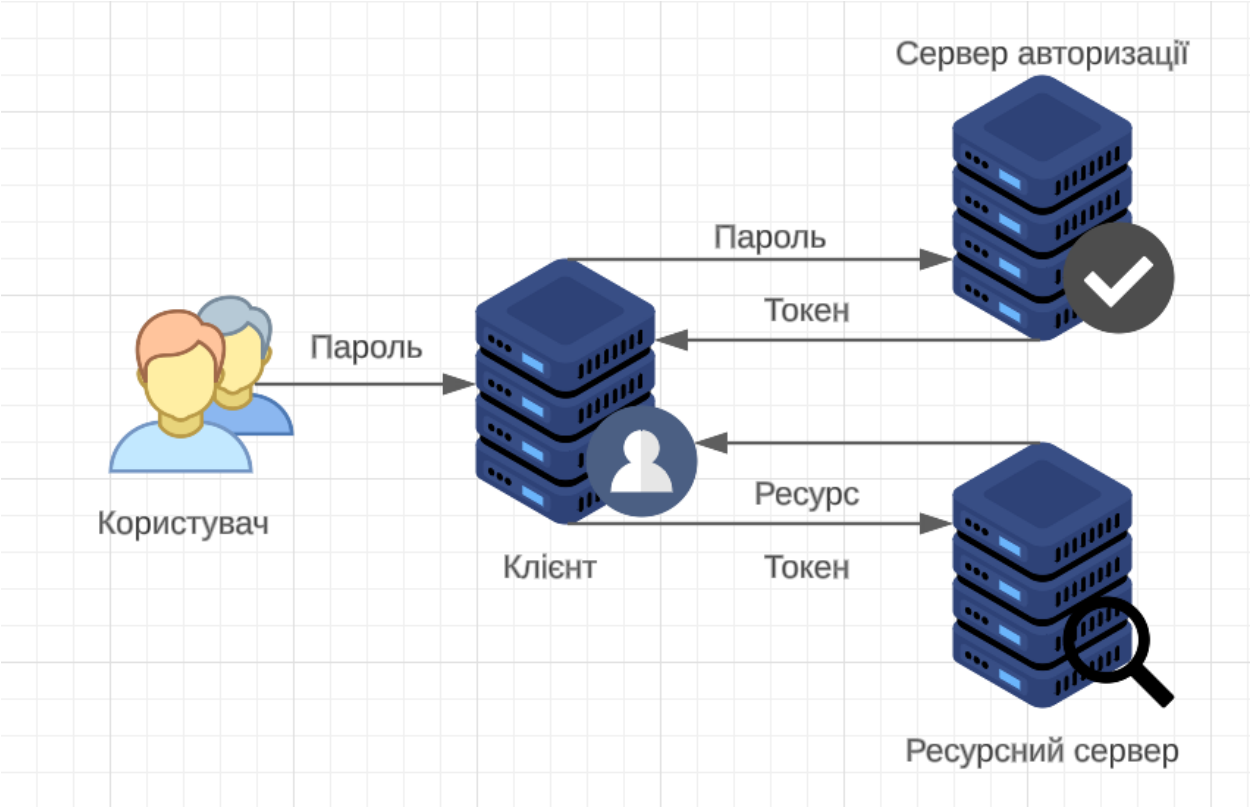

Рисунок 1.8 – Візуалізація що ілюструє автентифікацію на основі пароля

Двофакторна аутентифікація підвищує безпеку, вимагаючи від користувачів надання двох різних типів підтвердження. Зазвичай це включає щось, що користувач знає (пароль), і щось, що він має (наприклад, одноразовий код, надісланий на мобільний пристрій). Хоча 2FA є більш безпечною, ніж однофакторна аутентифікація, вона все ще може бути вразливою до підміни SIM-карт або фішингових атак. 2FA може створювати проблеми з юзабіліті, особливо для користувачів, які не звикли до додаткового етапу верифікації. 2FA на основі SMS може бути вразливим до викрадення SIM-картки. Також проблемою є залежність від пристрою: 2FA часто покладається на певний пристрій або застосунків, що робить його незручним для користувачів, які можуть не мати доступу до свого методу вторинної автентифікації (рис. 1.9).

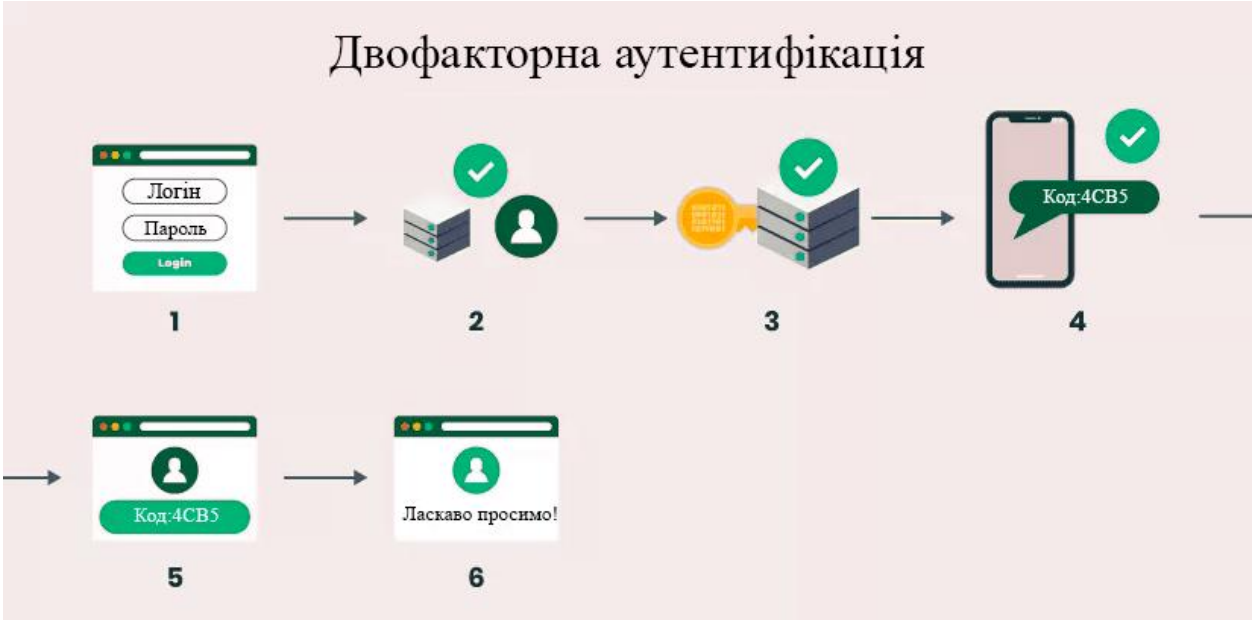

Рисунок 1.9 – Схема процесу двофакторної аутентифікації

Біометрична аутентифікація використовує унікальні фізичні або поведінкові атрибути людей для ідентифікації. Сюди входять відбитки пальців, розпізнавання обличчя, сканування райдужної оболонки ока та розпізнавання голосу. Біометричні методи забезпечують високий рівень безпеки, але можуть викликати занепокоєння щодо конфіденційності та зберігання даних. Виклики включають ризик витоку біометричних даних, варіації точності та надійності між різними біометричними системами, а також потенційні етичні проблеми, пов’язані з конфіденційністю (рис. 1.10).

Рисунок 1.10 – Наявні методи біометричної аутентифікації

KBA – передбачає відповіді на запитання, що ґрунтуються на особистій інформації. Обмеження включають:

– доступність даних, загальнодоступна особиста інформація робить KBA вразливим до атак з боку осіб, які мають доступ до персональних даних користувачів;

– забута інформація, користувачі можуть забути відповіді на свої секретні питання, що призводить до труднощів у відновленні облікового запису.

Верифікація на основі блокчейну використовує незмінність і безпеку технології блокчейн, щоб забезпечити децентралізовану систему управління ідентифікаційними даними, стійку до несанкціонованого втручання. Користувачі можуть контролювати доступ до своїх ідентифікаційних даних, підвищуючи рівень конфіденційності та безпеки. Виклики включають масштабованість блокчейн-мереж для перевірки особистих даних, необхідність створення зручних інтерфейсів та забезпечення сумісності з існуючими системами ідентифікації особи (рис. 1.11).

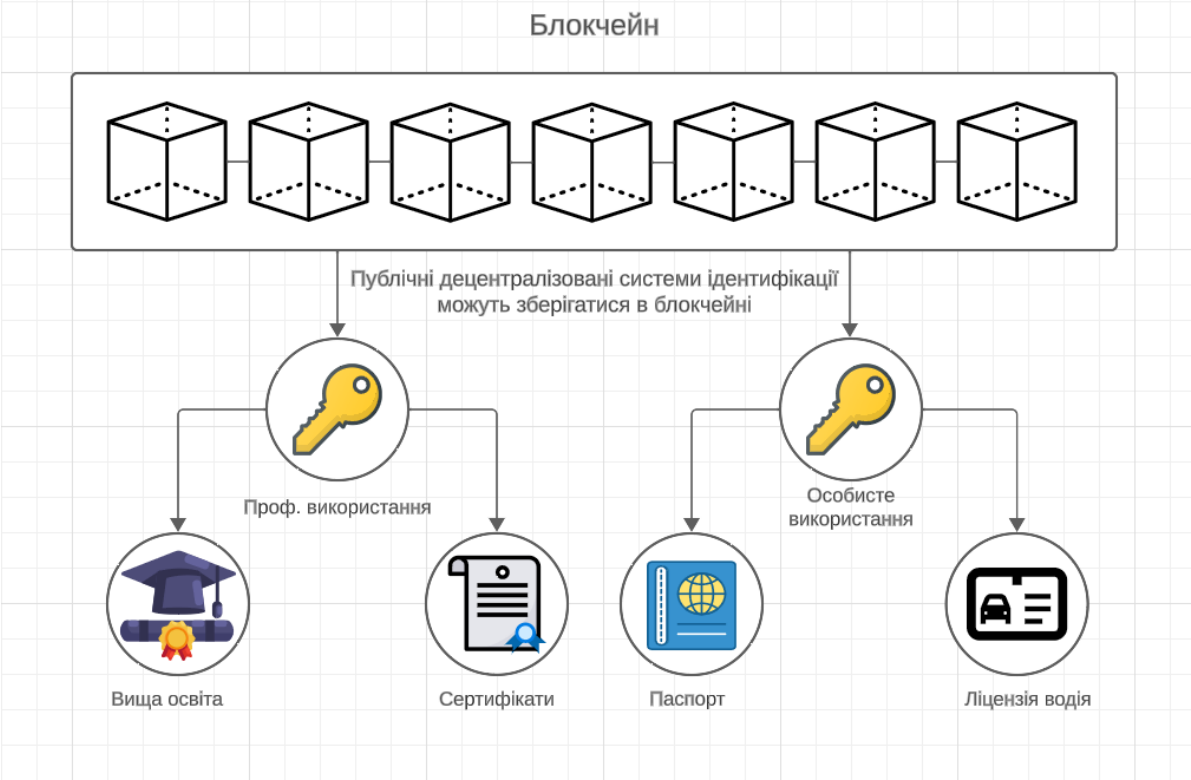

Рисунок 1.11 – Перевірка особи на основі блокчейну

Хоча ці методи онлайн-ідентифікації мають свої переваги, вони також мають суттєві недоліки, включаючи вразливість до витоку даних, проблеми з конфіденційністю та зручністю використання. Оскільки наша цифрова взаємодія продовжує розширюватися, виникає нагальна потреба в інноваційних та надійних рішеннях, які можуть усунути ці обмеження, надаючи пріоритет безпеці, контролю та зручності користувачів.

Дослідження цифрової верифікації ідентичності на основі блокчейну, є багатообіцяючим шляхом до подолання цих обмежень. Використовуючи децентралізовані, безпечні та орієнтовані на користувача характеристики блокчейну, ми прагнемо докорінно змінити спосіб підтвердження особистих даних в Інтернеті, забезпечуючи більшу довіру та безпеку в цифровій сфері.

1.5 Постановка задачі дослідження

Існуючі методи верифікації та автентифікації особистості в Інтернеті стикаються з серйозними проблемами з точки зору безпеки, конфіденційності та зручності використання. Традиційні системи, що базуються на паролях, вразливі до зламу, що призводить до витоку даних і незручностей для користувачів. Нові біометричні рішення, хоча і є більш безпечними, викликають занепокоєння щодо конфіденційності та захисту даних. Потреба в безпечній, конфіденційній і зручній для користувача системі цифрової верифікації особистості очевидна в нашому світі, який стає дедалі більш оцифрованим. Крім того, централізація даних користувачів в існуючих системах ідентифікації створює значні ризики для конфіденційності та безпеки. Технологія блокчейн дає можливість вирішити ці проблеми, пропонуючи децентралізоване і захищене від несанкціонованого втручання управління ідентифікаційними даними. Однак успішне впровадження вимагає подолання технічних і юзабіліті-викликів.

Об’єктом дослідження є цифрова ідентифікація на основі блокчейн технологій.

Метою дослідження є реалізація та дослідження вебінтерфейсу для цифрової ідентифікації за допомогою блокчейн технологій Ethereum, та локальної тестової блокчейн мережі Ganache.

Для досягнення мети необхідно вирішити такі завдання:

– проаналізувати існуючі рішення для ідентифікації на основі блокчейну;

– розробити прототип системи перевірки особи на основі блокчейну;

– дослідити та запропонувати стратегії для вирішення проблем масштабування рішень для ідентифікації особи на основі блокчейну.

2 СМАРТ-КОНТРАКТИ В МЕРЕЖІ ETHEREUM ТА ТЕХНОЛОГІЇ ЇХ РЕАЛІЗАЦІЇ

2.1 Основи смарт-контрактів в Ethereum та їх роль

Смарт-контракти приносять зміну парадигми договірних зобов'язань. Замість того, щоб покладатися на традиційні правові рамки, ці самодостатні контракти переводять угоди в програмний код, забезпечуючи автоматичне виконання заздалегідь визначених умов [38, 39].

Лідером децентралізованих платформ є Ethereum, ця екосистема стала революційною платформою, що пропонує простір не лише для криптовалютних транзакцій, але й для децентралізованих застосунків (DApps), заснованих на смарт-контрактах. Унікальність цієї платформи полягає в тому, що вона дозволяє здійснювати складні програмовані транзакції з використанням власної валюти – Ефіру.

Хоча для розробки смарт-контрактів Ethereum спочатку з'явилися різні мови, такі як Serpent, Solidity незабаром затьмарила їх. Синтаксис Solidity, схожий на JavaScript, в поєднанні з надійною функціональністю, адаптованою для блокчейну, зробили її кращим вибором для розробників. Сумісність з віртуальною машиною Ethereum (EVM) – середовищем виконання смарт-контрактів на Ethereum – ще більше посилює її важливість.

Розуміння нюансів смарт-контрактів Ethereum вимагає поглибленого вивчення їх життєвого циклу. Життєвий цикл смарт-контракту охоплює всі етапи, які проходить контракт, від його початкової розробки до остаточного виконання або припинення в блокчейні. Він надає безцінну інформацію про те, як формулюються, перевіряються, взаємодіють і, зрештою, укладаються контракти, відображаючи код, який перетворюється на непорушну угоду.

Життєвий цикл смарт-контракту:

– створення, перш ніж контракт буде реалізовано, він проходить численні ітерації, сеанси налагодження та ретельний аудит. Цей етап є життєво

важливим для того, щоб переконатися, що контракт не містить вразливостей, які в історії призводили до значних порушень, таких як атака DAO;

– розгортання, після ретельної розробки та тестування контракти розміщуються в мережі Ethereum. За розгортання стягується «газова» комісія, яка сплачується в Ефірі і залежить від складності контракту та перевантаженості мережі;

– виклик методу, після того, як контракт створений і активний, він очікує на взаємодію. Ці взаємодії можуть відбуватися через автоматичні тригери, встановлені іншими контрактами, або через дії, ініційовані користувачем. Кожна взаємодія з контрактом реєструється як транзакція в блокчейні і, як правило, потребує газу для виконання (рис. 2.1).

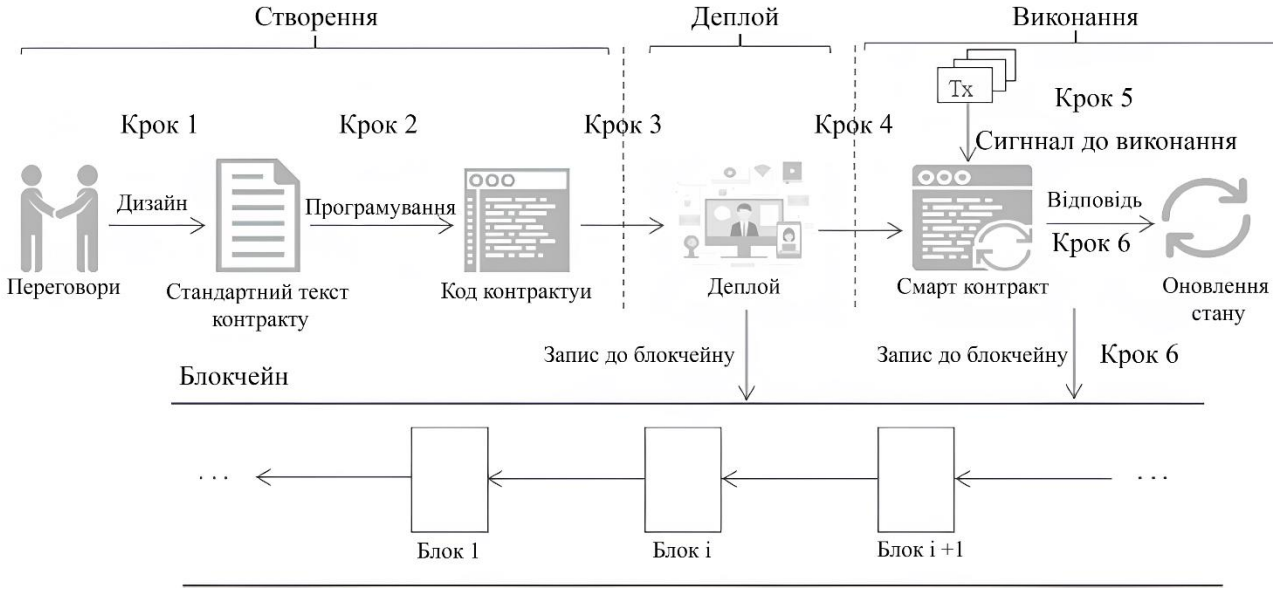

Рисунок 2.1 – Діаграма візуалізації життєвого циклу смарт-контракту

Ethereum, з його динамічною архітектурою, працює на децентралізованій мережі вузлів, кожен з яких містить копію всього блокчейну. Така структура забезпечує надлишковість даних і підвищену безпеку. Оскільки транзакції за участю смарт-контрактів транслюються через цю мережу, вони проходять ретельну перевірку за допомогою спеціальних механізмів консенсусу. Спочатку, використовуючи підхід Proof of Work (PoW), учасники, яких називали «майнерами», повинні були розв’язувати

складні криптографічні головоломки, щоб підтвердити і включити нові транзакції в блокчейн. Незважаючи на високий рівень безпеки, цей метод вимагає значних обчислювальних затрат за якості графічних чіпів відеокарт, що в майбутньому призвело до так званого «кризису відеокарт», і величезних енергетичних ресурсів. Складність цих головоломок диктується формулою, яка відображає складність майнінгу:

$$Difficulty = \frac{target_{max}}{CurrentTarget}, \tag{2.1}$$

де $Difficulty$ – відображає поточну складність видобутку;

$target_{max}$ вказує на максимальне потенційне значення цільової функції;

$CurrentTarget$ – її поточне значення.

Хоча PoW надійно забезпечує безпеку мережі, його обчислювальна інтенсивність і величезне енергоспоживання викликають занепокоєння. Усвідомлюючи ці недоліки, Ethereum перейшов на фреймворк Proof of Stake (PoS) з оновленням 2.0. Тут майнерів замінюють валідатори, які «заставляють» свою криптовалюту, щоб підтвердити свої повноваження валідатора. Ймовірність того, що валідатор буде обраний, щоб запропонувати наступний блок, пропорційна його частці:

$$P(validatior) = \frac{Stake_{validator}}{TotalStake}, \tag{2.2}$$

де $P(validator)$ – позначає ймовірність вибору валідатора;

$Stake_{validtor}$ – розмір ставки конкретного валідатора;

$TotalStake$ означає загальну суму стейків в мережі.

Цей перехід не тільки прокладає шлях до більшої енергоефективності, але також передбачає більш швидкі транзакції, сигналізуючи про більш масштабовану і стійку траєкторію розвитку Ethereum.

Невід'ємною частиною мережі Ethereum є інтероперабельність та оновлення. В умовах постійного розвитку технологічного ландшафту, Ethereum часто випускає оновлення, які називаються «хардфорками». Вони забезпечують підвищену безпеку, масштабованість і функціональність. Відомі форки, такі як Istanbul і Constantinople, впровадили оптимізацію, яка покращила розгортання і виконання смарт-контрактів (рис. 2.2).

Рисунок 2.2 – Основні етапи оновлення мережі Ethereum

2.2 Інструменти та технологічна інфраструктура для розроблення смарт-контрактів в Ethereum

Розвиток середовища смарт-контрактів Ethereum нерозривно пов'язаний з потужними технологіями та інструментами, які допомагають розробникам створювати складні децентралізовані застосунки. Віртуальна машина Ethereum (EVM) є життєво важливим компонентом екосистеми Ethereum і слугує децентралізованим виконавчим середовищем, що відповідає за рівномірну обробку смарт-контрактів у всій мережі Ethereum. Відзначений

Тьюрінг-повнотою, EVM має здатність виконувати будь-який алгоритм, незалежно від його складності. Кожна дія, від простих переказів Ефіру до більш складних функцій смарт-контракту, проходить перевірку кожним вузлом мережі, забезпечуючи консенсус щодо стану контракту і будь-яких подальших змін. Для полегшення цих операцій EVM використовує «газову» систему, яка не лише монетизує обчислення, але й забезпечує ефективне кодування та винагороджує прихильників мережі. Надаючи пріоритет безпеці, EVM ізолює кожен смарт-контракт, таким чином захищаючи мережу в цілому від потенційних вразливостей в окремих контрактах. Більше того, коли розробники використовують такі мови, як Solidity або Vyper, для створення застосунків Ethereum, вони генерують байт-код – серію відкритих кодів, які інтерпретуються EVM. В рамках EVM контракти розрізняють сховище, яке зберігає дані між викликами, і пам’ять, яка є ефемерною і скидається після кожної функції. Розуміння цих відмінностей, особливо з точки зору витрат на газ, є вкрай важливим, оскільки операції, керовані сховищем, зазвичай споживають більше газу, ніж ті, що пов’язані з пам’яттю.

У мережі Ethereum газ відіграє ключову роль у забезпеченні належного функціонування транзакцій і смарт-контрактів. Газ є одиницею виміру обчислювальних ресурсів, необхідних для виконання певних дій в мережі, таких як виконання функцій смарт-контрактів:

– газ, як одиниця виміру обчислювальних зусиль, необхідних для виконання операцій або смарт-контрактів. Концепція газу в Ethereum не тільки є фундаментальною для її функціональності, але й слугує захисним механізмом від зловживань у мережі. Кожна операція в Ethereum, від простої транзакції до виконання складного смарт-контракту, вимагає певної кількості обчислювальних ресурсів. Газ кількісно оцінює ці обчислювальні зусилля, гарантуючи, що кожна дія має вартість, пропорційну споживаним ресурсам. Вартість газу не є статичною, а коливається залежно від попиту на обчислювальні потужності та пропускної здатності мережі (табл. 2.1);

Таблиця 2.1 – Вартість газу для типових операцій в мережі Ethereum

Операція	Вартість газу
Базова транзакція	21,000
Складні контрактні операції	Від 30,000
Запис до сховища стану	20,000
SSTORE якщо значення = 0	5,000
Операція SLOAD	800
Операції ADD/SUB/MULT/DIV	Від 3 до 5
Операція LOG	Від 375 до 8,750

– Wei і Gwei, хоча газ є одиницею обчислення, платежі за ці обчислення здійснюються у валютних одиницях Ethereum. Wei – це найменший номінал валюти Ethereum. Gwei – це просто зручне позначення Wei, що дорівнює 10^9 Wei.

Користувачі встановлюють «ціну газу» в Gwei, коли надсилають транзакцію, представляючи кількість Wei, яку вони готові заплатити за кожну одиницю газу. Таким чином, загальна вартість транзакції в Wei розраховується шляхом множення ціни газу на кількість газу, спожитого в ході транзакції (табл. 2.2).

Таблиця 2.2 – Перерахунку одиниць Ethereum

Номінал	Значення у Wei
Wei	1
Gwei	10^9
Ether	10^{18}

Занурюючись в екосистему смарт-контрактів Ethereum, Remix стає важливою точкою дотику для розробників. Цей вебзастосунок з відкритим вихідним кодом пропонує середовище, пристосоване для розробки та розгортання смарт-контрактів. Такі ключові функції, як інтегрований

відладчик, дозволяють розробникам ретельно аналізувати кожну операцію EVM, гарантуючи, що контракти виконуються за призначенням. Крім того, інструмент «статичного аналізу» перевіряє контракти на наявність вразливостей, висвітлюючи потенційну неефективність газу і направляючи розробників до оптимізованого коду. В Ethereum кожна операція тягне за собою плату, виражену в «газі», щоб компенсувати обчислювальні зусилля вузлів мережі:

$$Total\ Gas = (Base\ Gas + Operational\ Gas) * Gas\ Price, \quad (2.3)$$

де $Base\ Gas$ – являє собою постійну плату за операцію;

$Operational\ Gas$ – коливається в залежності від складності обчислень.

Окрім простої розробки, Truffle пропонує набір інструментів для ретельного тестування, міграції та управління смарт-контрактами Ethereum [38]. Його можливості поширюються на компіляцію смарт-контрактів, лінкування та управління бінарними файлами. Особливої уваги заслуговує синергія Truffle з Ganache, персональним блокчейном для розробки Ethereum. Ganache імітує повну поведінку клієнта, роблячи перехід від розробки до реального розгортання (рис. 2.3).

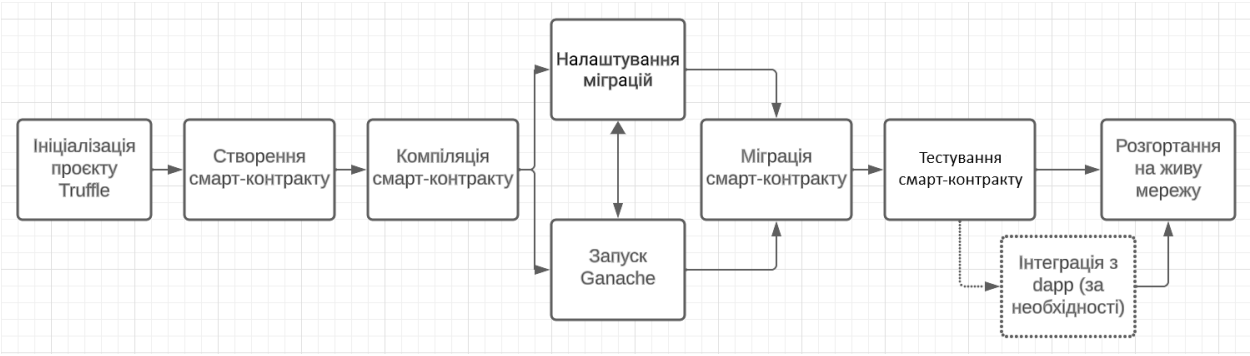

Рисунок 2.3 – Блок-схема, що описує процес розробки Truffle від створення, тестування і до розгортання з використанням Ganache

MetaMask, в першу чергу відомий як гаманець Ethereum, трансформував взаємодію користувачів з блокчейном Ethereum. Для розробників вбудовані інструменти , такі як dapp-браузер, що є безцінними для розробників. Цей інструмент спрощує взаємодію між децентралізованими застосунками та мережею Ethereum за допомогою складного алгоритму оцінки газу, що забезпечує ефективність транзакцій (рис. 2.4).

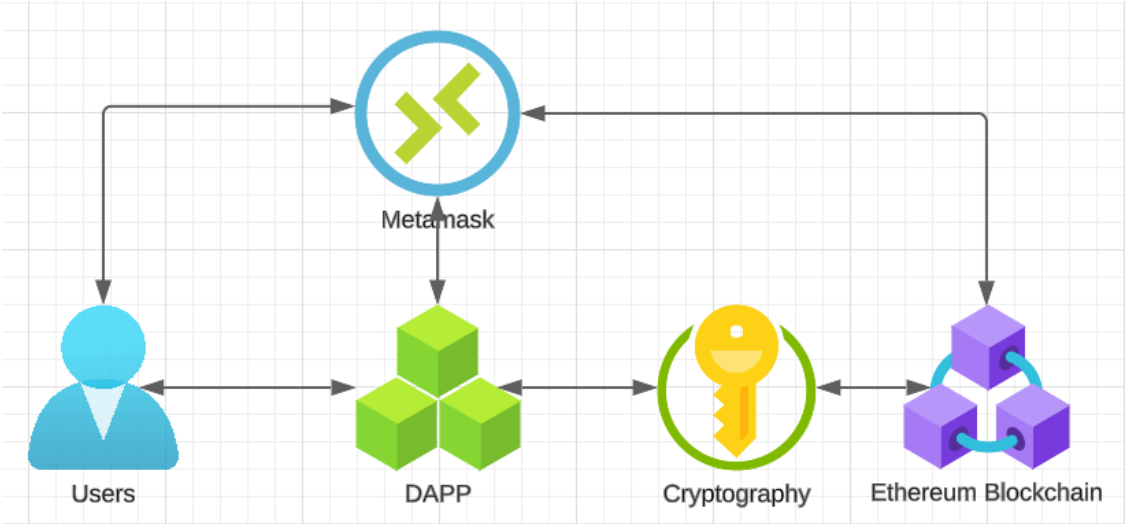

Рисунок 2.4 – Взаємодія Metamask з мережою Ethereum

Переходячи від традиційної веброзробки до блокчейну, розробники знаходять варіанти в Web3.js – мості, що полегшує цей перехід. Як набір бібліотек, Web3.js дозволяє розробникам взаємодіяти як з локальними, так і з віддаленими вузлами Ethereum за допомогою різних протоколів зв'язку. Однією з особливостей є можливість прямого виклику методів зі смарт-контрактів Ethereum завдяки Ethereum JSON RPC API. Крім того, Web3.js полегшує математичну задачу перетворення значень між Ефіром та його меншим номіналом, Wei:

$$Wei\ value = Ether\ value * 10^{18}. \quad (2.4)$$

Такі функції корисності є дуже важливими, оскільки EVM оперує значеннями переважно у Wei, забезпечуючи точність у всіх контрактних взаємодіях.

2.3 Безпека та оптимізація смарт-контрактів

У складному ландшафті технології блокчейн безпека та ефективність смарт-контрактів мають першочергове значення. Оскільки ці контракти діють автономно, один недогляд може призвести до значних вразливостей. Необхідно заглибитись в нюанси забезпечення того, щоб ці контракти не тільки працювали за призначенням, але й робили це оптимально [40].

Перш ніж занурюватися в механіку оптимізації смарт-контрактів, важливо зрозуміти потенційні загрози, з якими вони стикаються. Активна боротьба з цими загрозами може запобігти безлічі потенційних пасток (рис. 2.5):

– реінтеграційна атака, ця уразливість виникає, коли функція дозволяє зовнішні виклики до того, як встановиться її внутрішній стан, що потенційно може призвести до багаторазового виведення даних;

Лістинг 2.1 Оновлення стану:

```
function withdraw(uint amount) external {
require(balances[msg.sender] >= amount);
balances[msg.sender] -= amount;
(bool success, ) = msg.sender.call{value: amount}("");
require(success);}
```

– переповнення та недоповнення: Цілі числа без знаку в Solidity мають межі. Коли значення перевищує ці межі, воно або обнуляється, або перескакує до максимального значення (у випадку недоповнення). Забезпечення безпеки

математичних операцій часто передбачає використання бібліотек на кшталт SafeMath, які генерують помилки, коли виникають такі проблеми, замість того, щоб дозволяти числам згортатися.

Рисунок 2.5 – Ризики безпеки смарт-контрактів

Дуже важливо притримуватись принципів безпечного кодування. Впровадження надійних практик кодування може захистити контракти від потенційних загроз і забезпечити їхню довгострокову життєздатність.

Одним із таких прикладів є незмінні величини, це потужний інструмент проти зловмисників. Після встановлення ці значення або адреси не можуть бути змінені, таким чином гарантуючи статичну точку відліку або конфігурацію. Також одним із інструментів є Логування подій, в Ethereum транзакції є абсолютими, вони або повністю успішні, або повністю провалюються. Події надають можливість реєструвати важливі дії або зміни в контракті, надаючи розробникам і користувачам чітку історію і сліди того, що відбулося.

Газ є джерелом життєдіяльності мережі Ethereum, полегшуючи виконання транзакцій. Оптимізація споживання газу смарт-контрактами гарантує не лише економічну ефективність транзакцій, але й те, що вони не зупиняться через перевищення лімітів газу. Газовий механізм Ethereum слугує кільком цілям. Перш за все, він гарантує, що операції в мережі мають вартість, запобігаючи спаму або атакам на відмову в обслуговуванні. Крім того, він прив'язує обчислювальну роботу операцій до реальної вартості, забезпечуючи ефективність коду. Забезпечуючи безпеку, розробники часто стикаються з проблемою управління та оптимізації вартості газу своїх контрактів (рис. 2.6).

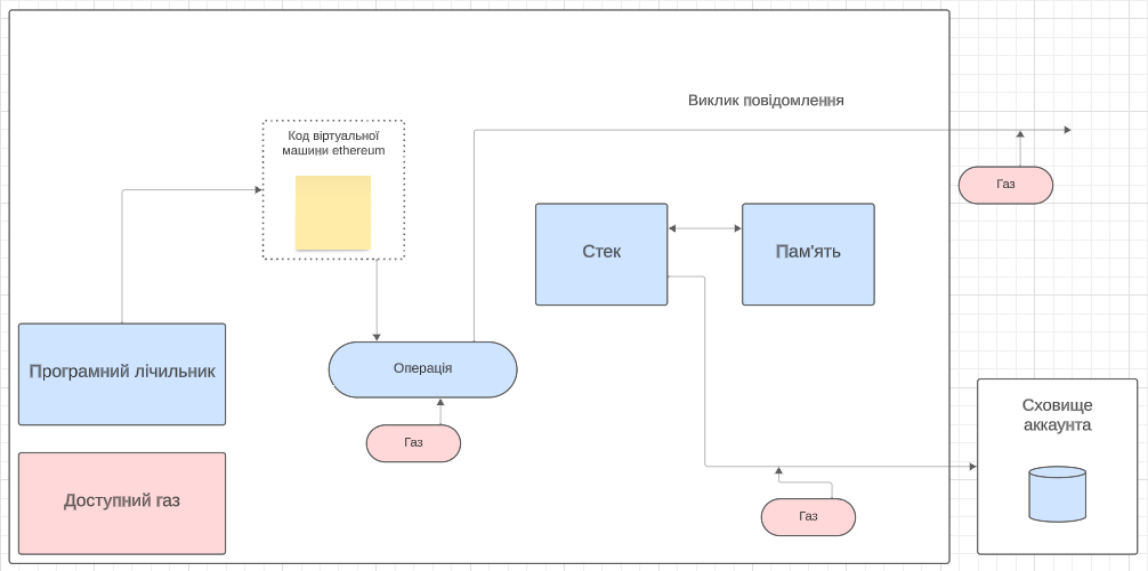

Рисунок 2.6 – Діаграма структури споживання газу EVM

Цикли, особливо необмежені, можуть швидко споживати велику кількість газу, особливо якщо вони пов'язані зі зберіганням або складними обчисленнями. Розбиття завдань, уникнення велик их циклів або використання позаланцюгових обчислень може суттєво оптимізувати контракт. Алгоритм варості циклу можна вирахувати за формули алгоритму циклічної вартості:

$$Gas_{total} = \sum_{i=1}^{n} Gas_i . \qquad (2.5)$$

Зберігання є однією з найдорожчих операцій в перерахунку на газ в Ethereum. Використання відповідних структур даних, мінімізація операцій зберігання та видалення непотрібних даних може значно зменшити витрати. Solidity зберігає змінні у слотах шириною 32 байти. Коли змінні упаковуються у структуру, Solidity намагається мінімізувати кількість використовуваних слотів, що потенційно економить газ. Також потрібно знати про Оптимізацію модифікаторів, код розміщується у модифікованій функції, а код модифікатора копіюється у всіх випадках його використання. Це призведе до збільшення розміру байткоду та використання газу.

Після етапу розробки, регулярний аудит і перевірка смарт-контрактів є життєво важливими. Ці процеси забезпечують додатковий рівень впевненості в тому, що контракт працює безпечно і за призначенням. Світ смарт-контрактів постійно розвивається, і в міру розвитку екосистеми можуть з’являтися нові вразливості. Регулярний аудит коду, як вручну, так і за допомогою автоматизованих інструментів, має вирішальне значення. Це не тільки забезпечує безпеку коштів та операцій, але й будує довіру з користувачами та зацікавленими сторонами. Крім того, після розробки контракти мають бути перевірені на таких платформах, як Etherscan, що робить вихідний код доступним для публічного перегляду та перевірки. Така прозорість гарантує, що користувачі можуть довіряти тому, що операції за контрактом відповідають його призначенню [41-43].

Безпека і оптимізація в смарт-контрактах є двома стовпами, що забезпечують надійність і ефективність децентралізованих застосунків. У міру того, як простір розвивається, розуміння і вирішення цих аспектів стає ще більш важливим, тим більше, при рості застосунку та його глобальності не в рамках закритого блокчейну, а при деплою його в публічні блокчейни, ці аспекти значно збільшуються, адже прогавини в безпеці та оптимізації можуть призвести до великих втрат інформації користувачів.

2.4 Застосування смарт-контрактів для цифрової ідентифікації

Блокчейн, як технологія, що постійно розвивається, розширив сферу свого застосування, вийшовши за рамки фінансових транзакцій. Цифрова ідентифікація за допомогою блокчейну обіцяє підвищену безпеку, надійність і контроль користувачів. У цьому сегменті висвітлюються можливості та тонкощі використання смарт-контрактів для цифрової трансформації.

Розгортання технології блокчейн у цифровій ідентифікації має кілька різних аспектів:

– децентралізований контроль, Традиційні системи ідентифікації управляються централізовано, але блокчейн пропонує децентралізований механізм, повертаючи користувачам контроль над їхніми персональними даними;

– незмінний запис, Після того, як особистість перевірена і збережена в блокчейні, вона не може бути змінена, що забезпечує захист від крадіжки особистих даних або шахрайства;

– глобальна доступність, Можна створити універсально доступну систему, яка дозволить користувачам отримати доступ до своєї ідентифікації з будь-якого місця [30].

Лістинг 2.2 Приклад контракту реєстрації особи користувача:

```
pragma solidity ^0.8.0;
contract Identity {
struct Person {string name; uint age; bool isVerified;}
mapping(address => Person) identities;
function addIdentity(string memory _name, uint _age) public {
identities[msg.sender] = Person(_name, _age, false);}
function verifyIdentity(address _user) public {
identities[_user].isVerified = true;}
```

Цей контракт демонструє можливість користувача зареєструвати свою особу та метод верифікації. Він підкреслює незмінну природу блокчейну після перевірки особи користувача.

Однак використання блокчейну для ідентифікації також пов’язане з певними проблемами:

– конфіденційність даних, публічні реєстри потребують вдосконалених криптографічних заходів для забезпечення конфіденційності персональних даних;

– інтероперабельність і масштабованість, сумісність між різними блокчейн-платформами та паралельна обробка численних перевірок ідентичності є обов’язковою умовою;

– масштабованість, блокчейн повинен ефективно обробляти величезну кількість процесів перевірки особи одночасно без вузьких місць.

Створення системи ідентифікації на основі блокчейну вимагає ретельного планування, архітектурного фреймворку та ітеративного вдосконалення. Процес полягає не лише в розробці системи, але й у забезпеченні її інтеграції з різними інтерфейсами застосунків, масштабованості для користувачів і, що найважливіше, стійкості до загроз безпеці.

Як базовий рівень системи, смарт-контракти не лише зберігають ідентифікаційні дані, але й підтримують логіку маніпуляцій з даними - додавання, верифікацію, пошук тощо. При розробці цих контрактів важливо передбачити майбутні зміни, тому виникає потреба в контрактах, які можна модернізувати. Цей рівень виконує функції як сховища даних, так і механізму управління ними. Модульний підхід дозволяє розділити різні компоненти логіки ідентифікації на окремі контракти, що покращує читабельність, можливість модернізації та безпеку. Ефективне структурування даних всередині контракту забезпечує мінімальне використання газу. Для цього часто використовують картографічні структури завдяки їхній властивості постійного доступу в часі.

Лістинг 2.3 Приклад картографічної структури:

```
mapping(address => UserIdentity) public identities;
```

Доступ на основі ролей гарантує, що лише авторизовані суб’єкти можуть читати або змінювати дані. Наприклад, лише певні ролі можуть мати право підтверджувати особу користувача або змінювати дані.

Лістинг 2.4 Контроль доступу до даних:

```
using Roles for Roles.Role;
Roles.Role private admins;
function verifyIdentity(address user) public {
require(admins.has(msg.sender), "Not an admin");
identities[user].isVerified = true;
}
```

Для підвищення прозорості та полегшення відстеження поза ланцюжком, корисно публікувати події після значних змін стану в контракті.

Лістинг 2.5 Реєстрація подій:

```
event IdentityVerified(address indexed user, string name, string
dateOfBirth);
function verifyIdentity(address user) public {
require(admins.has(msg.sender), "Not an admin");
identities[user].isVerified = true;
emit IdentityVerified(user, identities[user].name,
identities[user].dateOfBirth);
}
```

Передбачення та створення механізмів для потенційних майбутніх змін є важливим фактором. Це гарантує, що в міру розвитку сфери цифрової ідентифікації система залишатиметься адаптивною без шкоди для цілісності даних.

На етапі приєднання користувачів поєднуються зручність користування та надійна безпека. Основна увага приділяється автентичності особи під час створення ідентифікатора. Тут можуть бути використані такі механізми запобігання атакам Сивілл, як біометрія, апаратні токени та атестація третьою стороною.

$$U=hash(data+salt), \tag{2.6}$$

де U – унікальний хешований ідентифікатор користувача;

$data$ – дані користувач;

$salt$ – додатковий секрет для безпеки хешування.

Лістинг 2.6 Приєднання користувача:

```
function createUser(string memory data, string memory salt) public returns(bytes32) {
bytes32 hashedIdentity = keccak256(abi.encodePacked(data, salt));
return hashedIdentity;
}
```

3 РОЗГОРТАННЯ ЛОКАЛЬНОГО БЛОКЧЕЙНУ ТА ІНТЕГРАЦІЯ З ЗАСТОСУНКОМ

3.1 Створення тестової мережі MetaMask

У рамках кваліфікаційної роботи було розроблено протопим системи цифрової ідентифікації на базі блокчейну. Для реалізації даного застосунку був обраний ряд інструментів які дозволяють реалізовувати DApp, одним з таких інструментів є MetaMask.

В умовах розвитку блокчейну роль криптогаманців виходить за рамки простого зберігання цифрових активів. Вони є одним із ключових інструментів для взаємодії з децентралізованим Інтернетом, web3 та блокчейном виступаючи в якості каналу для управління адресами, аутентифікації транзакцій та взаємодії з застосунками. MetaMask виділяється серед всіх існуючих варіантів застосунку завдяки безшовній інтеграції з браузером, забезпечуючи інтуїтивно зрозумілий інтерфейс для транзакцій на основі Ethereum.

На початку 2020 року у MetaMask було трохи менше 5 мільйонів користувачів. До травня 2022 року їхня кількість наближалася до 30 мільйонів. Це робить MetaMask найпопулярнішим криптовалютним гаманцем у світі.

Хоча MetaMask не є єдиним браузерним гаманцем Ethereum, він пропонує одну з найкращих точок доступу для dApps. MetaMask також вдосконалюється, надаючи рішення для гаманців для установ і великих організацій.

Після встановлення MetaMask користувачі проходять простий шлях налаштування – створюють новий гаманець, після чого користувачу потрібно створити новий унікальний пароль який буде вторично захищати гаманець, також після цього користувачу стає доступна секретна фраза – mnemonic phrase, яка є унікальним ідентифікатором гаманця та дає доступ в під’єднанні його до інших пристроїв та також ця фраза діє як інструмент відновлення

доступу до гаманця. Спочатку MetaMask підключає користувачів до основної мережі Ethereum, але в цілях розробки фокус зміщується в бік тестової мережі яку потрібно під’єднати вручну та потім вибрати як основну для взаємодії з DApp (рис. 3.1).

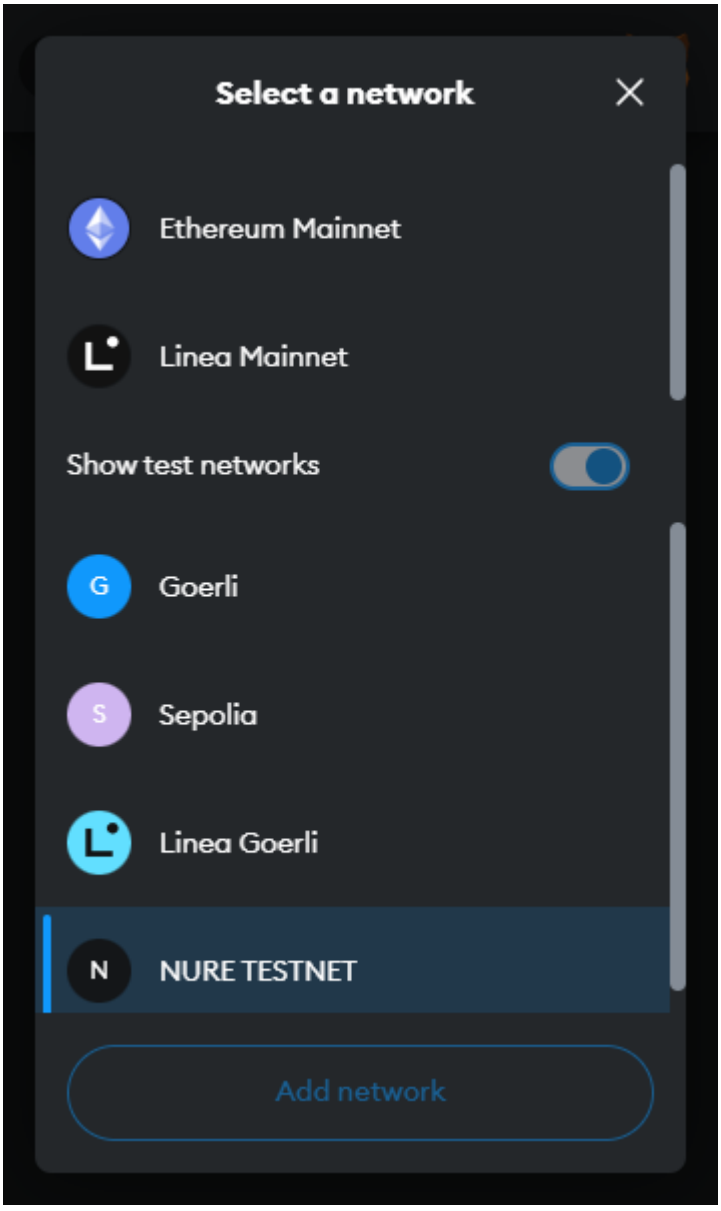

Рисунок 3.1 – Список мереж MetaMask

Обґрунтування вибору MetaMask для тестування полягає в його зручному інтерфейсі, що полегшує підключення до тестових мереж, таких як Ropsten , Rinkeby або як в нашому випадку локальної мережі Ganache. Такі мережі є безцінними для розробників, забезпечуючи реалістичне середовище

для розгортання та взаємодії зі смарт-контрактами без фінансових наслідків, пов’язаних з основною мережею Ethereum.

Налаштування MetaMask для тестової мережі передбачає кастомізацію налаштувань для імітації різних сценаріїв роботи блокчейну. Ця гнучкість дозволяє створити тестову мережу Localhost або підключитися до публічних тестових мереж, пропонуючи автентичний досвід для тестування застосунків. Доступ до тестового Ефіру з кранів – сервісів, які надають безкоштовний Ефір для тестування, ще більше покращує це симуляційне середовище.

Ще одним важливим кроком є включення в гаманець кастомних токенів, що відповідають стандартам ERC-20 або іншим типам токенів. Він імітує реальні операції з різноманітними цифровими активами, забезпечуючи комплексне тестування.

У деяких випадках розробникам може знадобитися вручну додати тестові мережі до MetaMask, щоб адаптувати середовище тестування до конкретних вимог або взаємодіяти з локальним екземпляром блокчейну, наприклад, Ganache. Таке ручне налаштування має вирішальне значення, коли тестові мережі за замовчуванням не відповідають потребам проекту або коли для тестування потрібне більш контрольоване та ізольоване середовище.

Щоб додати власну тестову мережу вручну, потрібно відкрити меню «Мережі» в MetaMask і вибрати «Додати мережу». Тут розробнику буде запропоновано ввести такі дані, як назва мережі, нова URL-адреса RPC, ідентифікатор ланцюжка, символ і URL-адреса провідника блоків. Ці параметри визначають, як MetaMask буде зєднуватися з користувацькою мережею і відображати її. Наприклад, при інтеграції з локальним екземпляром Ganache потрібно ввести адресу локального HTTP RPC-сервера, зазвичай наприклад http://127.0.0.1:7545, і ідентифікатор ланцюжка, який надає Ganache (рис. 3.2).

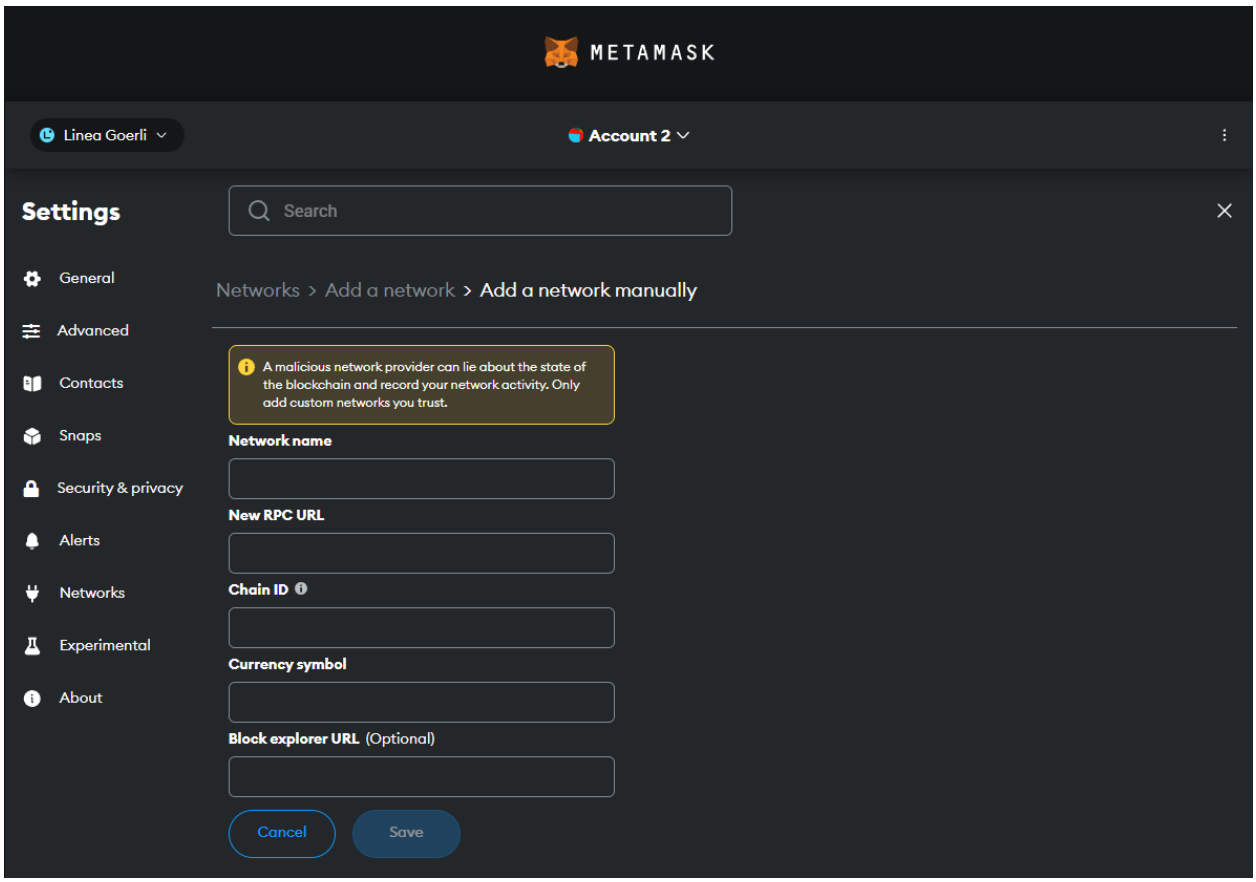

Рисунок 3.2 – Інтерфейс створення власної кастомної мережі MetaMask

Таке ретельне налаштування забезпечує безперешкодну інтеграцію з локальним блокчейном, дозволяючи розробнику розгортати смарт-контракти, ініціювати транзакції та взаємодіяти з блокчейном так, ніби він є основною мережею Ethereum, при цьому маючи повний контроль над поведінкою та станом мережі. Така точність гарантує, що будь-які проведені тести є ретельними і відображають умови, з якими зіткнеться застосунок після розгортання в основній мережі.

Отже, створення тестової мережі в рамках MetaMask є невід'ємним етапом розробки блокчейн-застосунків. Вона забезпечує безпечну і безкоштовну платформу для розробників для вдосконалення смарт-контрактів, перевірки функціональності і забезпечення бездоганного досвіду кінцевого користувача.

3.2 Підключення смарт контракту до вебінтерфейсу

Злиття смарт-контрактів з інтерфейсом – це делікатний процес, який поєднує надійну функціональність блокчейну зі зручним для користувача інтерфейсом. Щоб досягти такої безшовної взаємодії, було створено інтуїтивно зрозумілий інтерфейс, вибір дизайну якого значною мірою зумовлений потребою в простоті та ефективності взаємодії з користувачем.

Спочатку було інкапсулювано основні функції нашого смарт-контракту в зручний інтерфейс. Використовуючи HTML5 і CSS, було розроблено адаптивний макет, який підлаштовується під різні пристрої, забезпечуючи доступність і простоту використання. Інтерфейс представляє користувачам інтерактивні елементи, які запускають операції смарт-контракту, не перевантажуючи їх складнощами технології блокчейн, що лежить в основі транзакцій.

Неможливо переоцінити ключову роль JavaScript (JS) у цій інтеграції. Він слугує каналом, через який наш фронтенд взаємодіє з блокчейном Ethereum. Ми використали web3.js – набір бібліотек, які дозволяють нашому вебзастосунку взаємодіяти з локальним або віддаленим вузлом Ethereum за допомогою HTTP, IPC або WebSocket.

Web3.js – це важлива бібліотека JavaScript, яка дозволяє вебінтерфейсу взаємодіяти з блокчейном Ethereum. Дана бібліотека JavaScript відіграє ключову роль фасилітатора, посередника, який дозволяє інтерфейсу безперешкодно взаємодіяти з мережею Ethereum. За допомогою web3.js можна надсилати та отримувати дані з блокчейну, викликати методи смарт-контрактів та прослуховувати події, і все це в режимі реального часу та з мінімальною аппаратною затримкою.

Використовуючи web3.js, застосунок встановлює з мережею Ethereum через вузол Ethereum або шлюзовий сервіс, такий як Infura. Це зєднання схоже на міст, який дозволяє двом окремим сутностям – браузеру користувача і децентралізованому світу Ethereum – легко обмінюватися інформацією. Після

підключення web3.js надає інструменти для ініціювання транзакцій, які є основою будь-якої взаємодії в блокчейні. Ці транзакції дозволяють користувачам виконувати функції смарт-контракту, будь то переказ криптовалюти або запуск будь-якої зміни стану в рамках логіки контракту.

Крім того, web3.js відіграє важливу роль у прослуховуванні та обробці подій, що генеруються смарт-контрактами. Ця можливість обробки подій гарантує, що наш застосунок залишається чуйним та інформативним, надаючи користувачеві зворотній завязок в режимі реального часу про результати його транзакцій або будь-які зміни в стані контракту.

Лістинг 3.1 Перевірка користувача на наявність MetaMask за допомогою Web3.js:

```
if (typeof window.ethereum !== 'undefined') {
window.web3 = new Web3(window.ethereum);
try {
await window.ethereum.request({ method: 'eth_requestAccounts' });
} catch (error) {
console.error("User denied account access");
document.getElementById('status-message').innerText = "User denied
account access";
return; }
} else {
console.error("Please install MetaMask!");
document.getElementById('status-message').innerText = "Please install
MetaMask!";
return; }
```

Для розробки та розгортання смарт-контракту було використовано Truffle Suite – середовище, яке надає набір інструментів для написання, тестування та розгортання смарт-контрактів. Інтерфейс командного рядка

Truffle пропонує простий спосіб компіляції та розгортання смарт-контрактів, а його конфігураційний файл дозволяє нам визначати такі деталі, як налаштування мережі та конфігурацію компілятора. Truffle дозволяє організувати код смарт-контрактів. Кожен смарт-контракт міститься у власному файлі Solidity, який знаходиться в каталозі контрактів. Truffle розпізнає ці файли і компілює їх в ABI (Application Binary Interface) і байт-код, які необхідні для розгортання контрактів в блокчейні. Розгортання смарт-контрактів в мережі Ethereum управляється за допомогою скриптів міграції, які є файлами JavaScript в каталозі міграцій. Ці скрипти дають нам точний контроль над процесом розгортання, наприклад, над тим, які контракти розгортати, в якому порядку і як надавати аргументи конструктора [39].

Лістинг 3.2 Код міграції:

```
const ConvertLib = artifacts.require("ConvertLib");
const MetaCoin = artifacts.require("identitySystem");
module.exports = function(deployer) {
 deployer.deploy(ConvertLib);
deployer.link(ConvertLib, MetaCoin);
deployer.deploy(MetaCoin);
};
```

Конфігурація Truffle визначається в файлі truffle-config.js. Тут ми вказуємо мережеві налаштування для розгортання, такі як мережа Ethereum (локальна, тестова або основна) і обліковий запис, з якого будуть розгортатися контракти (на якому повинно бути достатньо Ефіру для покриття витрат на розгортання). Ми також можемо встановити додаткові параметри, такі як ліміт газу і ціну газу, в контексті даної роботи ці параметри можна скопіювати з локального блокчейну Ganache.

Лістинг 3.3 Конфігурація truffle config:

```
development:
{
host: "127.0.0.1",
port: 7545,
network_id: "5777",}
```

Truffle також може працювати з Truffle Boxes – шаблонними наборами для різних типів застосунків на основі Ethereum. Вони можуть включати інтерфейсні бібліотеки та компоненти, такі як React або Angular, для взаємодії зі смарт-контрактами. Truffle компілює контракти і вставляє необхідні екземпляри web3 у інтерфейсний код, полегшуючи процес інтеграції [37].

Ретельна увага до деталей на етапі інтеграції гарантувала, що кожна транзакція, чи то розгортання контракту, чи то передача токенів ERC-20, буде виконана з точністю. Тісний зв’язок між інтерфейсом і смарт-контрактом був спроектований таким чином, щоб забезпечити надійний і водночас інтуїтивно зрозумілий користувацький інтерфейс.

3.3 Розгортання локального блокчейна

Локальні блокчейни слугують важливими пісочницями для розробників, забезпечуючи середовище, в якому вони можуть безпечно тестувати та налагоджувати застосунки перед розгортанням у публічних мережах. Вони імітують поведінку публічного блокчейну, пропонуючи таку ж функціональність, але при цьому є ізольованими і контрольованими.

Ganache – це персональний блокчейн для розробки Ethereum, який можна використовувати для розгортання контрактів, розробки застосунків і запуску тестів. Він доступний як у вигляді десктопного застосунку, так і у

вигляді інструменту командного рядка (раніше відомий як TestRPC). Ganache є частиною набору інструментів Truffle Suite, створенного для того щоб розробка на Ethereum стала простішою та ефективнішою.

Ця платформа імітує реальне блокчейн-середовище з функціями, спрямованими на полегшення розробки. Наприклад, вона дає розробникам можливість робити форк з будь-якого існуючого блокчейну, включаючи основну мережу Ethereum (Mainnet). Це означає, що можна імітувати стан основної мережі Ethereum, включаючи всі транзакції до певного моменту, щоб протестувати свої контракти з реальними даними і умовами.

Ganache надає ряд стандартних облікових записів, попередньо профінансованих фальшивим Ефіром, що усуває необхідність видобувати або шукати тестовий Ефір. Цей аспект спрощує тестування і розробку, оскільки ви можете здійснювати транзакції миттєво, не чекаючи на підтвердження, що було б характерно для реальної мережі (рис. 3.3).

ADDRESS	BALANCE	TX COUNT	INDEX	
0×7f94a8dFaf3e1b18D893CB5e890d47Fb0f5104C7	99.96 ETH	39	0	
0×a02c85C43A1cFc8bb018ba6ae7B149868D7eCCd7	100.00 ETH	2	1	
0×1CeD021A331A3c440eCA16b09ac7b8585dD507D6	100.00 ETH	3	2	
0×a5d58F18278c3609E10D8E1d81F89E3aB90BB999	100.00 ETH	5	3	
0×164De7F2b0986Aa2f2be1907aB4509c52160B3AB	100.00 ETH	6	4	
0×efE126B469be2EF658de0Fd2de897EB9330B4f9b	100.00 ETH	0	5	
0×6bCDb60a3BB1Aa66e98984180fbacD2C2BAced51	100.00 ETH	3	6	

Рисунок 3.3 – Список облікових записів Ganache

Кожен раз, коли Ganache запускається, він створює абсолютно новий блокчейн в пам’яті. Це зручно для тестування, оскільки щоразу процес починається з чистого аркуша і не має артефактів попередніх тестів. Однак, для

поточних проектів це також дозволяє підтримувати стан, запускаючи його з базою даних.

Ключовою особливістю Ganache є можливість контролювати час роботи блокчейну, що дозволяє вам імітувати такі умови, як прострочені транзакції або бачити, як поводяться ваші контракти в різний час. Він також включає в себе розширений контроль майнінгу, який дозволяє встановлювати час блоків, призупиняти майнінг і відновлювати його, що є критично важливим для тестування функцій, які залежать від конкретних умов блоків.

При першому створенні проекту в Ganache потрібно створити новий проект та зв’язати його з файлом конфігурації truffle в директорії застосунку, після чого запустити проект (рис. 3.4).

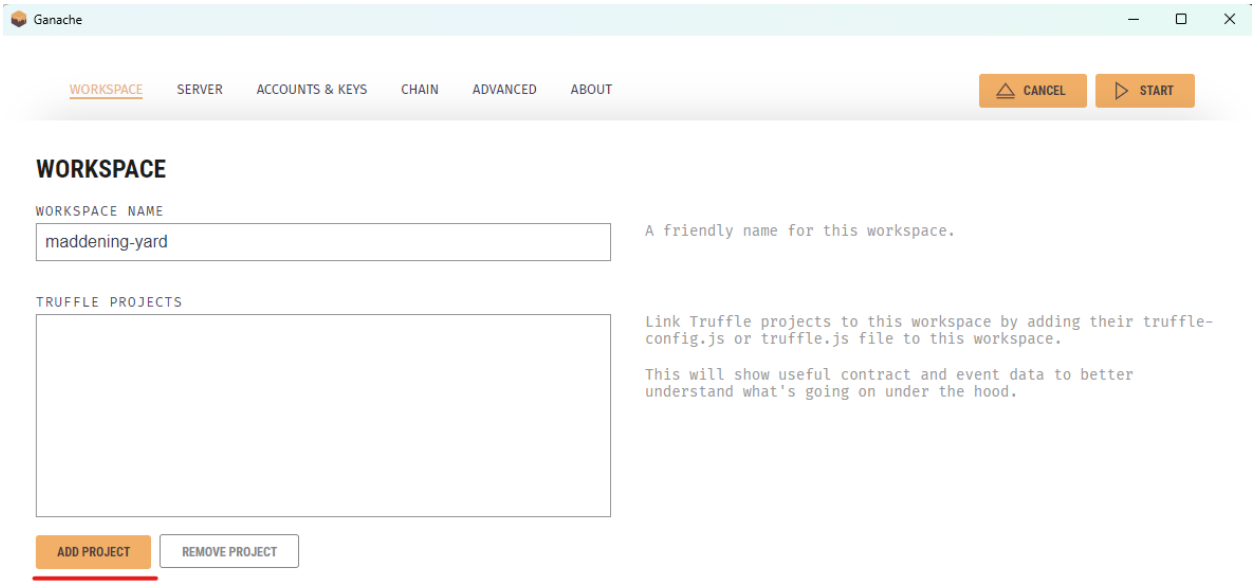

Рисунок 3.4 – Інтерфейс створення нового проекту в Ganache

Налаштування Ganache дуже просте. Після запуску програма генерує блокчейн, який запускається локально на комп’ютері. На головному екрані є акаунти з попередньо завантаженим тестовим ефіром, номер поточного блоку, ліміт газу та іншу важливу інформацію (рис. 3.5).

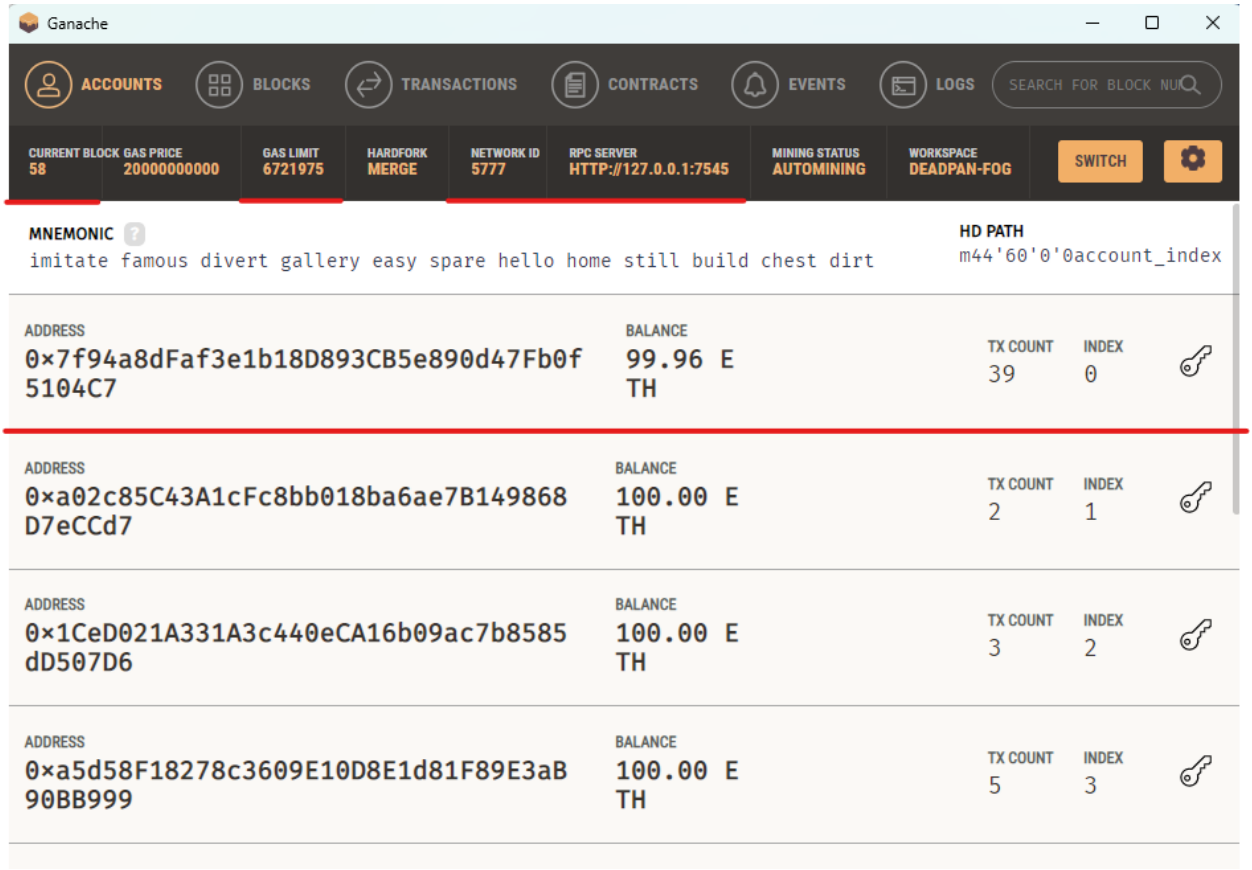

Рисунок 3.5 – Інтерфейс Ganache з виділеними основними елементами локальної мережі

Розділ налаштувань дозволяє вам налаштувати різні аспекти, такі як налаштування сервера, включаючи ім’я хоста, номер порту і мережевий ідентифікатор, процеси автоматизації та оповіщень встановлені нативно в систему розгортання локального блокчейну, а також облікові записи і ключі, де є можливість встановити кількість облікових записів для генерації і баланс Ефіру для кожного з них (рис. 3.6).

Інтеграція застосунку з Ganache починається з налаштування файлу truffle-config.js для підключення до локального блокчейну, створеного Ganache. Потрібно визначити налаштування мережі розробки, вказавши на локальний екземпляр Ganache.

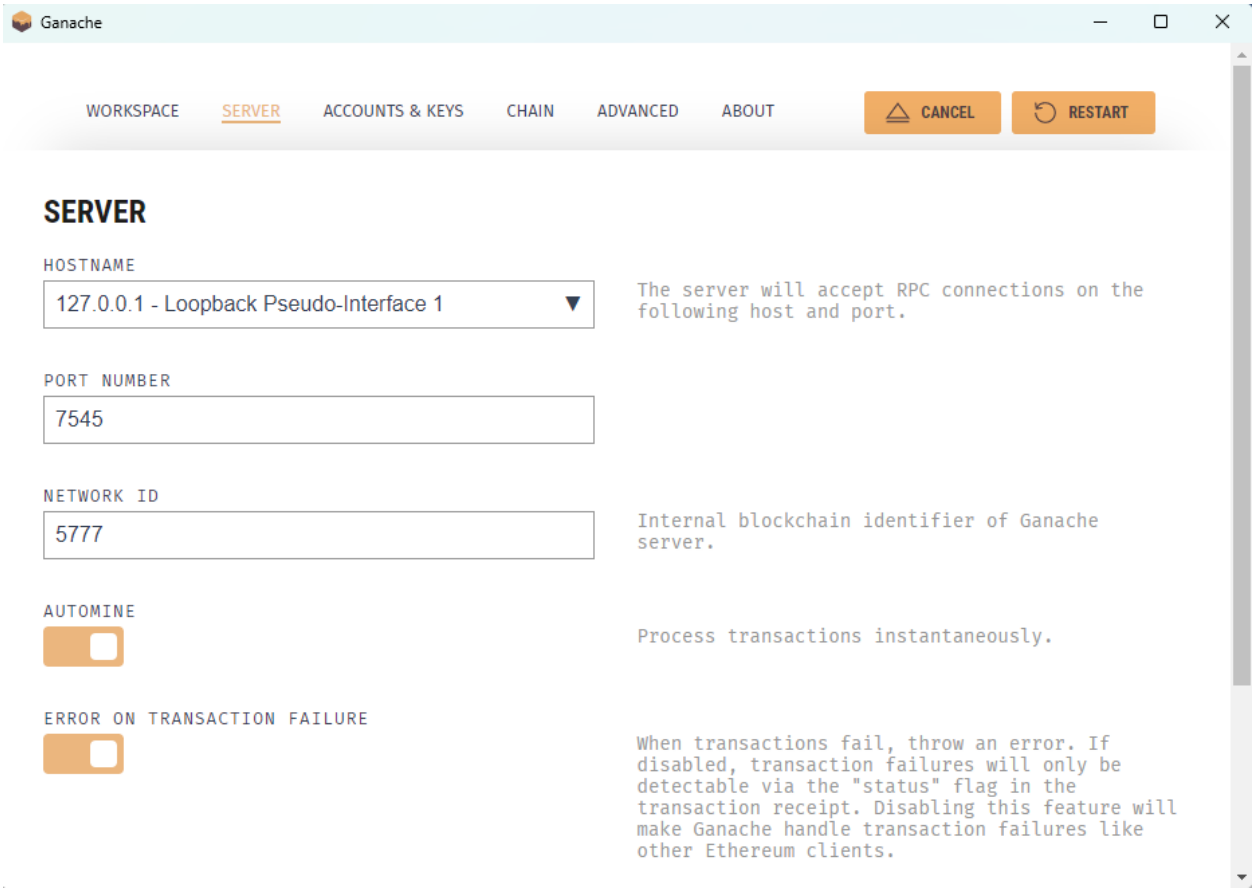

Рисунок 3.6 – Інтерфейс налаштування локального сервера блокчейну Ganache

Далі потрібно використувати Truffle консоль, або нативний термінал windows – PowerShell для компіляції смарт-контрактів використовубчи команду *truffle compile*, потім потрібно мігрувати контракти в блокчейн Ganache за допомогою команди *truffle migrate --network development*. Це розгортає наші контракти в локальному блокчейні, де можна взаємодіяти з ними через консоль Truffle або безпосередньо через інтерфейс Ganache.

Тепер інтерфейс програми може взаємодіяти з смарт-контрактами, розгорнутими на локальному блокчейні Ganache, через web3.js, що забезпечує безперервний цикл розробки та тестування.

Після проведених маніпуляцій можна запускати і тестувати застосунок в контрольованому середовищі. Будь-які транзакції та взаємодія з смарт-контрактами відбуваються миттєво і відображаються в інтерфейсі Ganache,

забезпечуючи негайний зворотній зв’язок і чітке уявлення про поведінку системи (рис. 3.7).

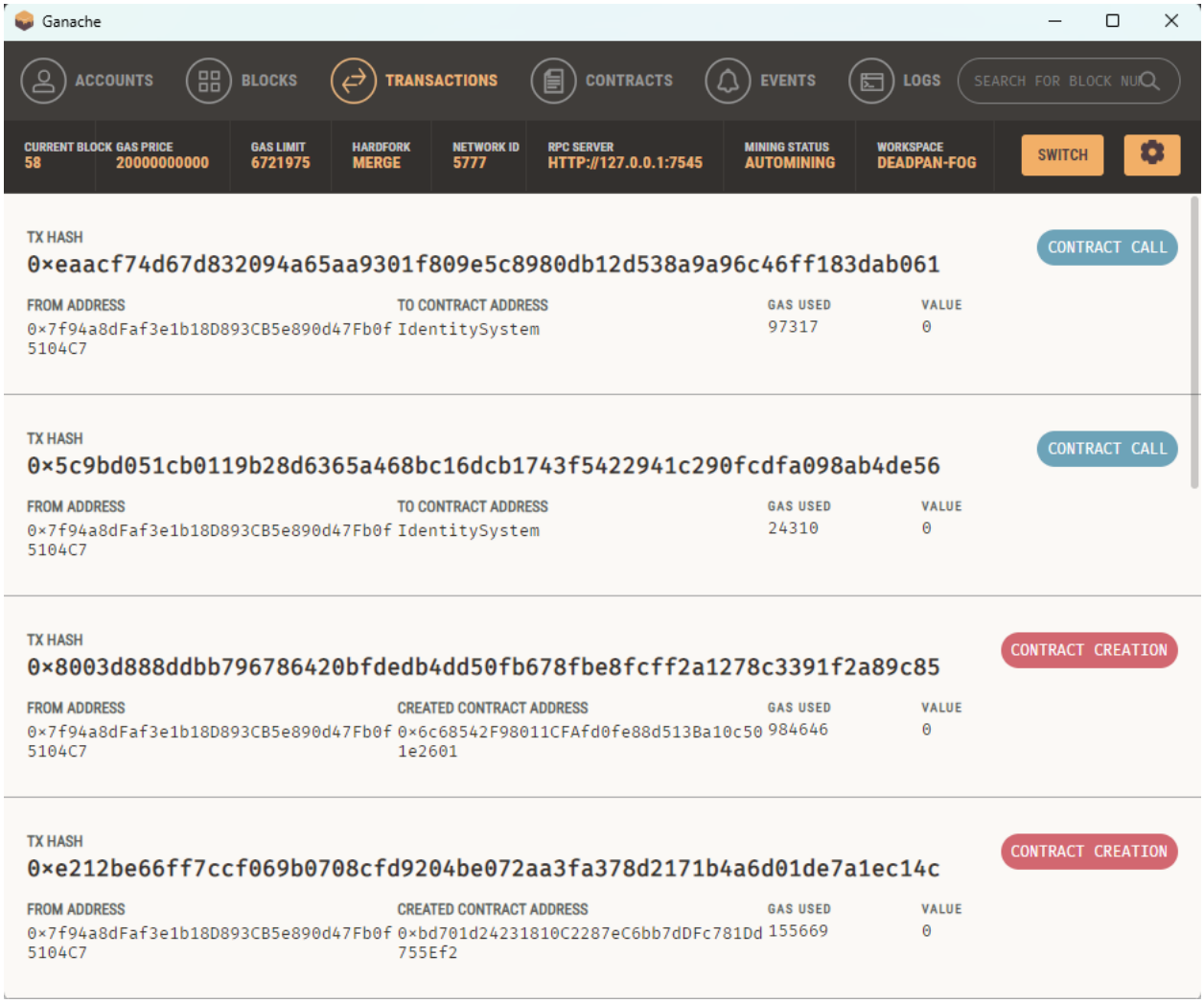

Рисунок 3.7 – Взаємодія контрактів Ganache що відображена у формі транзакцій

Таке налаштування є важливим для забезпечення правильної роботи всіх компонентів програми перед її розгортанням у реальних умовах. Це також гарантує, що будь-які проблеми можуть бути виявлені за допомогою вкладки з логами та виправлені з мінімальним ризиком.

3.4 Запуск та тести

Після налаштування локального блокчейн-середовища та успішної інтеграції смарт-контракту у фронт-енд, наступний етап включав запуск застосунку та проведення комплексних тестів для забезпечення функціональності, продуктивності та безпеки. При запускі застосунку через VS Code плагін – live server, на початковому екрані користувача зустрічає нативний MetaMask нотифікатор який пропонує юзеру ввести пароль до його облікового гаманця (рис. 3.8).

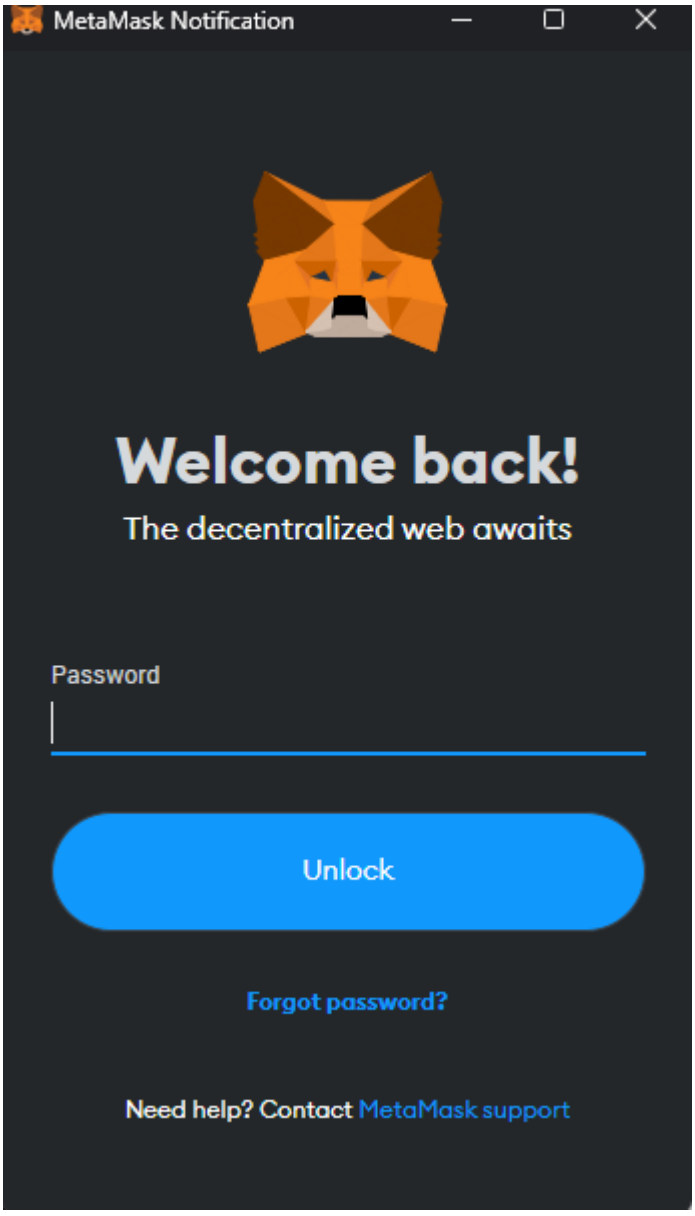

Рисунок 3.8 – Інтерфейс поля вводу пароля до гаманця MetaMask

Першим кроком було ініціювати застосунок, щоб перевірити, що він працює безперебійно в локальному налаштуванні блокчейну. Це було досягнуто шляхом запуску скриптів запуску програми та забезпечення правильного завантаження інтерфейсу, який безперешкодно підключався до блокчейну. Після вдалого старту в ітерфейсі Ganache можно відслідкувати смарт-контракти які були мігровані до вебзастосунку, та які готові до виконання або виконуються (рис. 3.9).

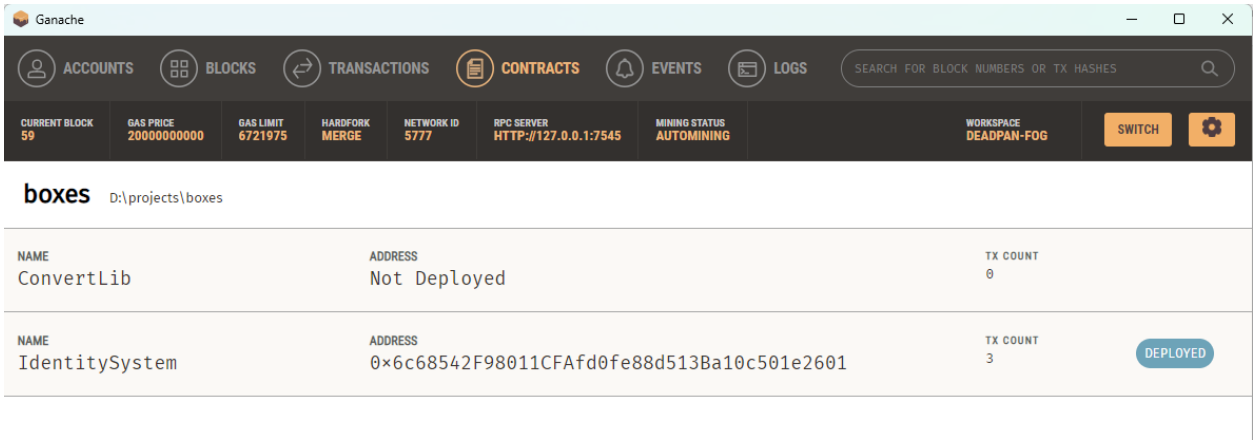

Рисунок 3.9 – Список мігрованих смарт-контрактів до локальної блокчейн мережі Ganache

Наступним кроком стала взаємодія зі смарт-контрактом через інтерфейс користувача. Такі дії, як ініціювання транзакцій, запити до блокчейну на отримання даних та імітація взаємодії з користувачем, здійснювалися для того, щоб переконатися, що смарт-контракт викликається коректно. Користувач ввода дані в поле «Create your identity», після чого для виклику дії смарт-контракту потрібно натиснути кнопку «Create identity», яка напряму викликає смарт контракт, після чого на екрані прийде повідомлення від нативного нотифікатора MetaMask про успішне, або ні, виконня функції смарт контракта, але в любому випадку в блокчейні буде відображена транзакція яка свідчить про те що взаємодія між вебінтерфейсом та блокчейн мережою налагоджена та працює правильно (рис. 3.10).

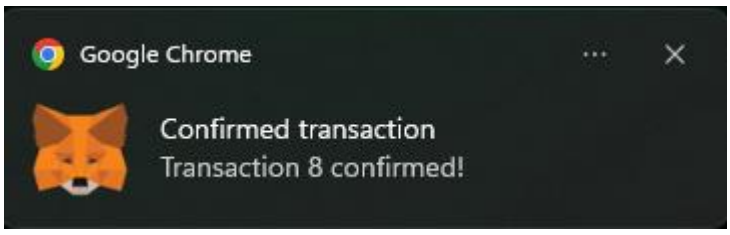

Рисунок 3.10 – Повідомлення про успішне виконання транзакції з смарт-контрактом

Якщо у смарт-контракті виникають проблеми, такі як помилки виконання або неочікувана поведінка, вони зазвичай відображаються в консолі розробки. Консоль реєструє помилки і надає трасування стеку, результати викликів функцій та інформацію про події, що допомагає в налагодженні. Наприклад, некоректне виконання функції може спричинити помилку повернення в консолі, а помилка перевищення ліміту газу вказує на те, що транзакція перевищила ліміт газу. Ці журнали мають значення при розробці, щоб знайти і виправити проблеми перед розгортанням смарт-контракту в основній мережі.

Для оцінки потоку транзакцій у застосунку були проведені комплексні тести. Це включало створення та підписання транзакцій, відправлення їх до локального блокчейну та спостереження за їх додаванням до блоків. Інтерфейс Ganache зіграв тут незамінну роль, відображаючи деталі транзакцій та інформацію про блоки для перевірки. При створенні нового користувача та отримання повідомлення про успішний виклик транзакції та роботи смарт-контракту, в Ganache в список транзакцій та вкладку Event, яка є індикатором успішності події, і пропускає лише події які успішно виконуються, бо транзакція навіть в випадку не успішного викання все одно відображається та є відкритою, що є однією з характеристик блокчейн технологій з її децентралізованістю та відкритістю що покладено в фундамент та є невід’ємною частиною даної технології (рис. 3.11).

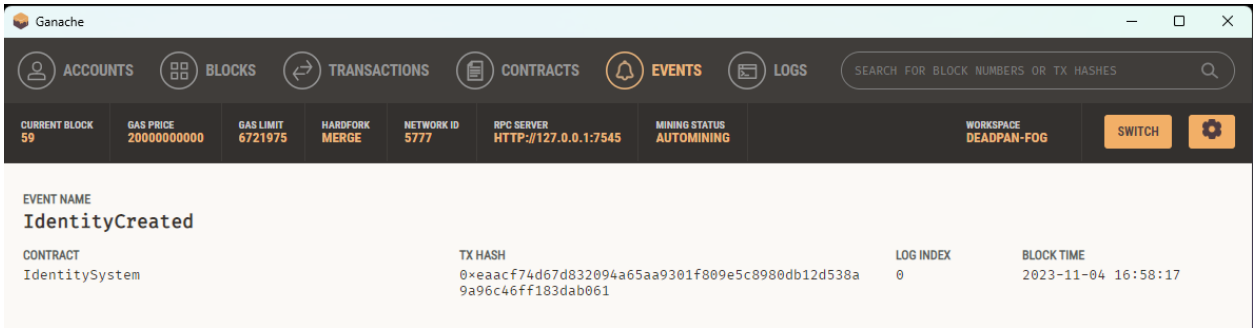

Рисунок 3.11 – Список подій після успішного виклику смарт-контракту через вебінтерфейс

При перегляді подій в Event, потрібно натиснути на поточну подію та переглянути правильність інформації переданої через вебінтерфейс (рис. 3.12).

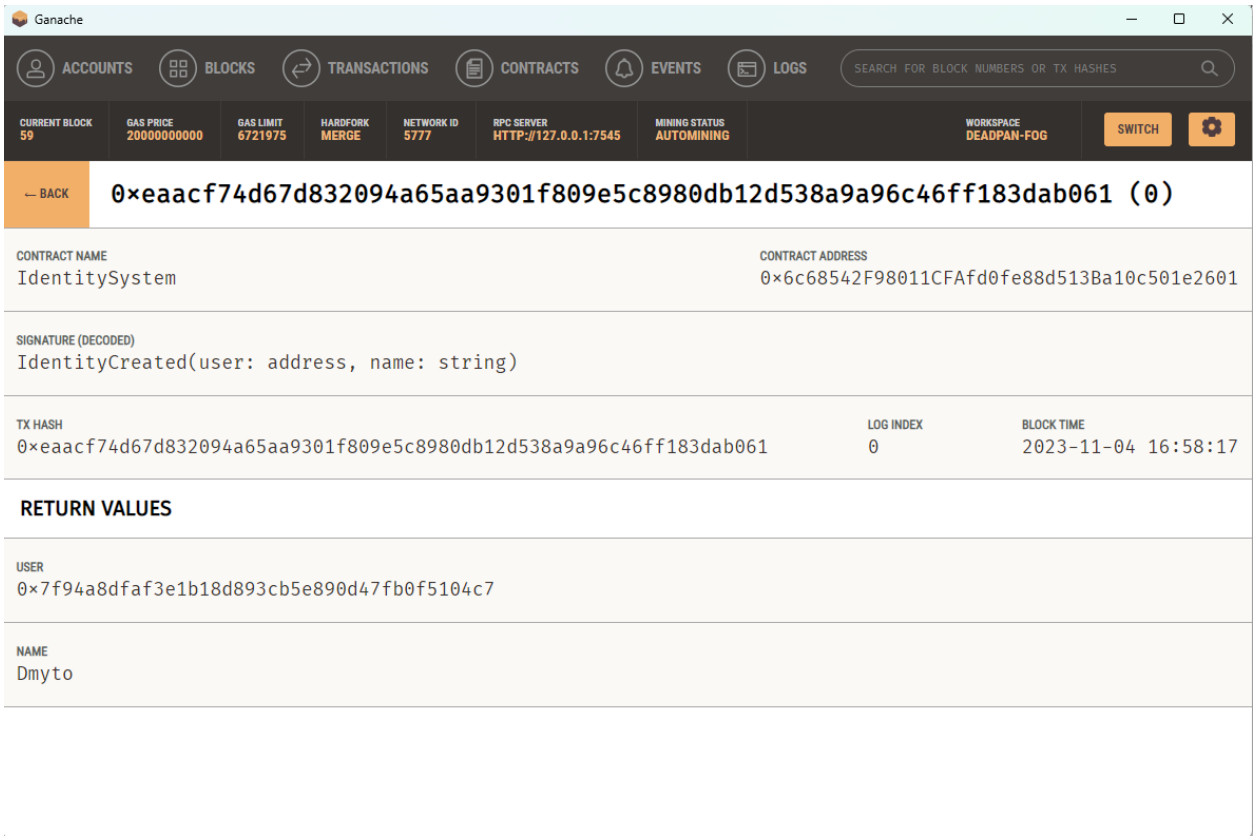

Рисунок 3.12 – Інформація про подію

Всередині події можно переглянути інформацію яка передалась від вебінтерфейсу до блокчейну. Тут є ім’я контракту, яка зазначає назва файлу в якому був записаний смарт-контракт та мігрований до тестової блокчейн

мережі, блокчейн адреса записана за допомогою хешування в строчці contract address, параметри які викликає смарт-контракт, в даному випадку це запис та передача string змінних таких як: ім’я користувача (_name), адреса (_adress) та дата народження (_dob), хеш транзакції, та дані користувача який викликав цей контракт, а саме, хеш його Web3 криптогаманця, та ім’я яке указане в «_name» передається за допомогою написаного коду, ця функція не є обов’язковою в даній технології, але в контексті даної роботи було прийняте рішення по реалізації даної можливості для полегшення користувацького досвіду. Більш детальна інформація з усіма даними переданими з вебінтерфейсу можна переглянути на вкладці transaction, де вибравши потрібний контракт можно детально ознайомитись з інформацією (рис. 3.13).

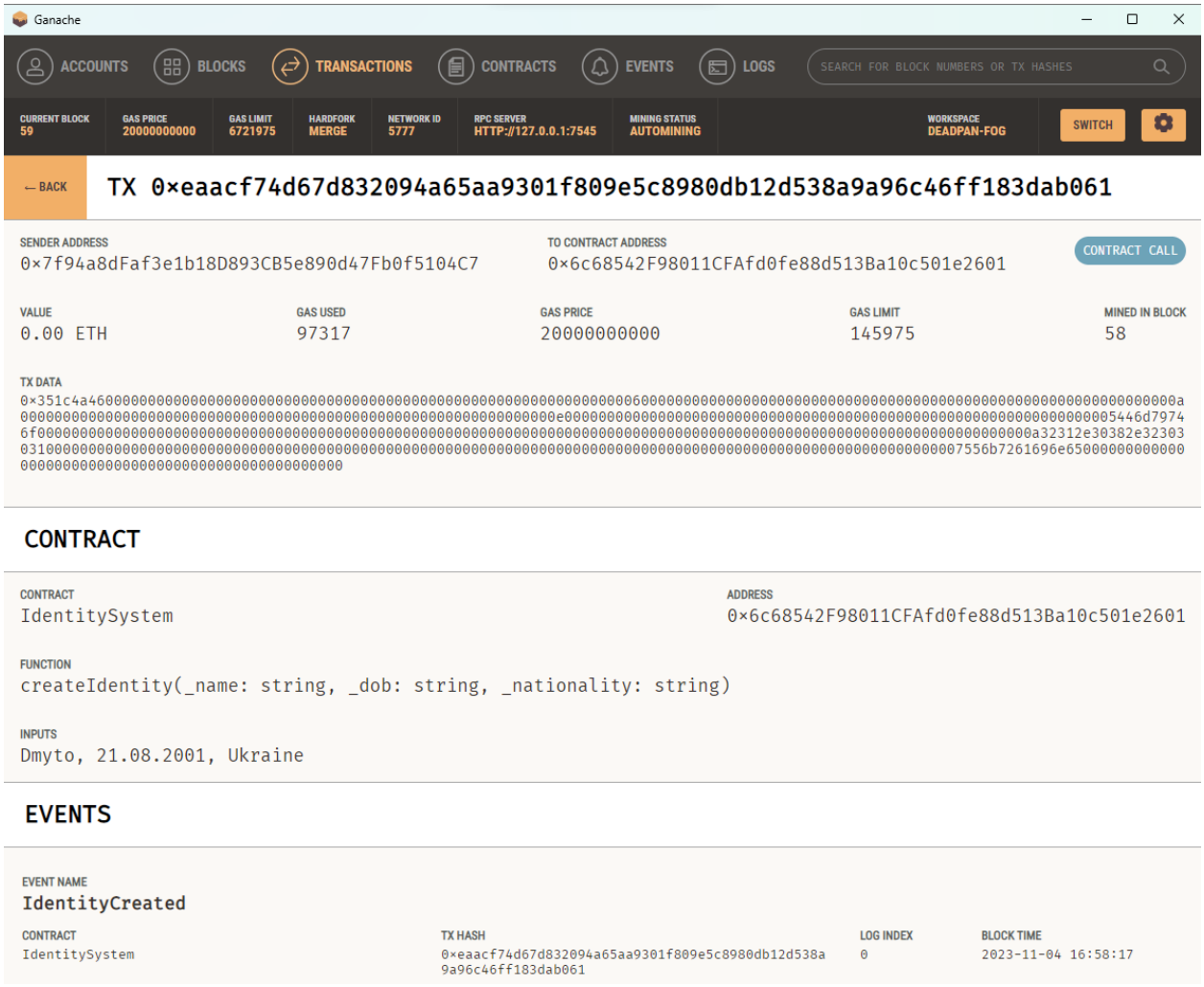

Рисунок 3.13 – Дані транзакції передані з вебвінтерфейсу

Щоб забезпечити надійність, обробка помилок була протестована шляхом моделювання різних сценаріїв, включаючи невірні вхідні дані, невдалі транзакції та переривання зв’язку. Поведінка програми в цих умовах відстежувалася для підтвердження відповідних повідомлень про помилки які виводились в консоль браузера.

Лістинг 3.4 Код для виклику помилки невказаного веріфаєра:

```
contract.methods.registerVerifier(verifierAddress).send({ from: account })
then(function(receipt) {
console.log("Verifier has been registered:", receipt);
})
catch(function(error) {
onsole.error("Error registering verifier:", error);
});
```

Іншим важливим аспектом була продуктивність, особливо час відгуку програми на дії користувача та швидкість підтвердження транзакцій у локальному блокчейні. Були зібрані показники, які допомогли оцінити будь-які потенційні вузькі місця або проблеми з продуктивністю.

Враховуючи важливість безпеки в блокчейн-застосунках, було проведено аудит безпеки. Це включало тестування на загальні вразливості та перевірку коду смарт-контрактів на наявність потенційних експлойтів.

Нарешті, було проведено користувацьке тестування з потенційними кінцевими користувачами, щоб зібрати відгуки про зручність та функціональність застосунків. На основі цих відгуків були внесені корективи, щоб переконатися, що кінцевий продукт відповідає очікуванням і вимогам користувачів.

ВИСНОВКИ

У рамках кваліфікаційної роботи був розроблений і реалізований прототип системи цифрової ідентифікації на основі блокчейн-технологій.

Вибір блокчейн-технологій грунтувався на його надійних механізмах безпеки та децентралізованій парадигмі, що є принципово важливими для застосунків, які вимагають незмінних записів даних та незалежної взаємодії між учасниками.

Була обрана платформа Ethereum та Ganache завдяки репутації як зрілої та універсальної блокчейн-платформи з широкими можливостями для смарт-контрактів. Широко використовуване середовище програмування, що підтримується сильною спільнотою, ідеально підходить для розробки децентралізованих застосунків. Ganache, що входить до складу Truffle, надає зручний локальний блокчейн для безпечної та ефективної розробки і тестування. Він дозволяє реалістично імітувати середовище Ethereum без витрат, пов’язаних з основною мережею.

Дослідницький сегмент роботи надав цінні знання про те, як блокчейн може вирішувати конкретні проблеми, пропонуючи нові можливості для розробників і користувачів, зокрема, у забезпеченні цілісності даних і вдосконаленні операційних процесів.

Комплексне тестування підтвердило функціональність та надійність системи. Ряд тестів моделював різноманітні сценарії, щоб гарантувати, що продуктивність системи відповідає очікуванням за різних умов експлуатації.

Дослідження дійшло висновку, що блокчейн є перспективним рішенням для цифрової ідентифікації, забезпечуючи безпечну, приватну та ефективну перевірку особи [44-48].

Результати роботи апробовано у вигляді тез доповідей під час національної конференції «NEW WAYS OF CREATING SCIENTIFIC IDEAS FOR IMPLEMENTATION» [49].

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Rabotiahov, A., Kobylin, O., Dudar, Z., & Lyashenko, V. (2018, February). Bionic image segmentation of cytology samples method. In *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)* (pp. 665-670). IEEE.

2. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень.

3. Lyashenko, V., Mohammad, A., & Kobylin, O. (2015). Experiments with Fusion of Images with Use of Wavelet Transformation in Problems of the Text Information Analysis.

4. Kobylin, O., Vyskrebentseva, S., & Petrova, R. (2019). Обробка даних, що містять пропуски в задачах кластеризації. *Системи управління, навігації та зв'язку. Збірник наукових праць*, *5*(57).

5. Oleg, K., Sergii, M., & Mykhailo, S. (2017, October). Video Clustering via Multidimensional Time-Series Analysis. In *Proceedings of the 9th International Conference on Information Management and Engineering* (pp. 60-63). ACM.

6. Mashtalir, S., Mashtalir, V., & Stolbovyi, M. (2018, August). Representative Based Clustering of Long Multivariate Sequences with Different Lengths. In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)* (pp. 545-548). IEEE.

7. Bodyanskiy, Y., Kobylin, I., Rashkevych, Y., Vynokurova, O., & Peleshko, D. (2018, February). Hybrid fuzzy-clustering algorithm of unevenly and asynchronously spaced time series in computer engineering. In *2018 14th International Conference on Advanced Trends in Radioelecrtronics, Telecommunications and Computer Engineering (TCSET)* (pp. 930-935). IEEE.

8. Bodyanskiy, Y., Vynokurova, O., Kobylin, I., & Kobylin, O. (2016). Adaptive fuzzy clustering of short time series with unevenly distributed observations

in Data Stream Mining tasks. *Information Technology and Management Science, 19*(1), 23-28.

9. Lyashenko V., Kobylin O., Selevko O. (2020) Wavelet Analysis and Contrast Modification in the Study of Cell Structures Images. International Journal of Advanced Trends in Computer Science and Engineering. 9(4). – 4701-4706.

10. Mashtalir, V., Ruban, I., & Levashenko, V. (Eds.). (2019). Advances in Spatio-Temporal Segmentation of Visual Data (Vol. 876). Springer Nature.

11. Kobylin, O., & Lyashenko, V. (2016). Contrast Modification as a Tool to Study the Structure of Blood Components.

12. Kobylin, O. A., Gorokhovatskyi, V. O., Tvoroshenko, I. S., & Peredrii, O. O. (2020). The application of non-parametric statistics methods in image classifiers based on structural description components. Telecommunications and Radio Engineering, 79(10).

13. Kobylin, O., & Lyashenko, V. (2014). Comparison of standard image edge detection techniques and of method based on wavelet transform.

14. Кобилін, О. А., & Творошенко, І. С. (2021). Методи цифрової обробки зображень.

15. Gorokhovatskiy, V. A., Kobylin, O. A., & Kulikov, Y. A. (2015). Application of Granulation of Feature Descriptions in Structural Image Recognition. *Telecommunications and Radio Engineering*, *74*(6).

16. Kuzminska, O., Mazorchuk, M., Morze, N., & Kobylin, O. (2019, June). Digital learning environment of ukrainian universities: The main components to influence the competence of students and teachers. In *International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications* (pp. 210-230). Springer, Cham.

17. Kinoshenko, D., Kobylin, O., Mashtalir, S., & Stolbovyi, M. (2019, March). Metric video retrieval speedup by irrelevant data elimination. In *Eleventh International Conference on Machine Vision (ICMV 2018)* (Vol. 11041, pp. 176-183). SPIE.

18. Lyashenko, V., Matarneh, R., Kobylin, O., & Putyatin, Y. (2016). Contour detection and allocation for cytological images using Wavelet analysis methodology.

19. Lyashenko, V., Kobylin, O., & Ahmad, M. A. (2014). General methodology for implementation of image normalization procedure using its wavelet transform.

20. Gorokhovatskyi V., Tvoroshenko I., Kobylin O., and Vlasenko N. (2023) Search for visual objects by request in the form of a cluster representation for the structural image description, Advances in Electrical and Electronic Engineering, 21(1), pp. 19-27.

21. Yakovleva, O., Kovtunenko, A., Liubchenko, V., Honcharenko, V., & Kobylin, O. (2023). Face Detection for Video Surveillance-based Security System (COLINS-2023). In *CEUR Workshop Proceedings* (Vol. 3403, pp. 69-86).

22. Tvoroshenko I., Gorokhovatskyi V., Kobylin O., and Tvoroshenko A. (2023) Application of deep learning methods for recognizing and classifying culinary dishes in images, International Journal of Academic and Applied Research, 7(9), pp. 57–70.

23. Takemiya, M., & Vanieiev, B. (2018, July). Sora identity: Secure, digital identity on the blockchain. In *2018 ieee 42nd annual computer software and applications conference (compsac)* (Vol. 2, pp. 582-587). IEEE.

24. Di Pierro, M. (2017). What is the blockchain?. *Computing in Science & Engineering*, *19*(5), 92-95.

25. Ammous, S. (2016). Blockchain technology: What is it good for?. *Available at SSRN 2832751*.

26. Rodeck, D., & Curry, B. (2022). What is blockchain. *Forbes. www. forbes. com/advisor/investing/cryptocurrency/what-is-blockchain/(22 May 2022). Search in*.

27. Masiero, S., & Bailur, S. (2021). Digital identity for development: The quest for justice and a research agenda. *Information Technology for Development*, *27*(1), 1-12.

28. Dunphy, P., & Petitcolas, F. A. (2018). A first look at identity management schemes on the blockchain. *IEEE security & privacy*, *16*(4), 20-29.

29. Li, D., Peng, W., Deng, W., & Gai, F. (2018, July). A blockchain-based authentication and security mechanism for IoT. In *2018 27th International Conference on Computer Communication and Networks (ICCCN)* (pp. 1-6). IEEE.

30. Gao, S., Su, Q., Zhang, R., Zhu, J., Sui, Z., & Wang, J. (2021). A privacy-preserving identity authentication scheme based on the blockchain. *Security and Communication Networks*, *2021*, 1-10.

31. Guo, H., & Yu, X. (2022). A survey on blockchain technology and its security. *Blockchain: research and applications*, *3*(2), 100067.

32. Xu, X., Weber, I., & Staples, M. (2019). *Architecture for blockchain applications* (pp. 1-307). Cham: Springer.

33. Maesa, D. D. F., & Mori, P. (2020). Blockchain 3.0 applications survey. *Journal of Parallel and Distributed Computing*, *138*, 99-114.

34. Khan, S. N., Loukil, F., Ghedira-Guegan, C., Benkhelifa, E., & Bani-Hani, A. (2021). Blockchain smart contracts: Applications, challenges, and future trends. *Peer-to-peer Networking and Applications*, *14*, 2901-2925.

35. Almakhour, M., Sliman, L., Samhat, A. E., & Mellouk, A. (2020). Verification of smart contracts: A survey. *Pervasive and Mobile Computing*, *67*, 101227.

36. Wang, Z., Jin, H., Dai, W., Choo, K. K. R., & Zou, D. (2021). Ethereum smart contract security research: survey and future research opportunities. *Frontiers of Computer Science*, *15*, 1-18.

37. Voshmgir, S. (2020). *Token Economy: How the Web3 reinvents the internet* (Vol. 2). Token Kitchen.

38. Ahamed, N. N., & Vignesh, R. (2023, March). A Build and Deploy Ethereum Smart Contract for Food Supply Chain Management in Truffle-Ganache Framework. In *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)* (Vol. 1, pp. 36-40). IEEE.

39. Verma, R., Dhanda, N., & Nagar, V. (2022, July). Application of truffle suite in a blockchain environment. In *Proceedings of Third International Conference on Computing, Communications, and Cyber-Security: IC4S 2021* (pp. 693-702). Singapore: Springer Nature Singapore.

40. Panda, S. K., & Satapathy, S. C. (2021). An investigation into smart contract deployment on Ethereum platform using Web3. js and solidity using blockchain. In *Data Engineering and Intelligent Computing: Proceedings of ICICC 2020* (pp. 549-561). Springer Singapore.

41. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., Gadetska S., and Al- Dhaifallah M. (2023) Statistical data analysis models for determining the relevance of structural image descriptions, *IEEE Access*, 11, pp. 126938-126949.

42. Daradkeh Y.I., Gorokhovatskyi V., Tvoroshenko I., and Zeghid M. (2022) Tools for fast metric data search in structural methods for image classification, *IEEE Access*, 10, pp. 124738-124746.

43. Гороховатський В.О., Творошенко І.С., Чмутов Ю.В. (2022) Застосування систем ортогональних функцій для формування простору ознак у методах класифікації зображень, *Сучасні інформаційні системи*, 6(3), С. 5-12.

44. Гороховатський В., Передрій О., Творошенко І., Марков Т. (2023) Матриця відстаней для множини компонентів структурного опису як інструмент для створення класифікатора зображень, *Сучасні інформаційні системи*, 7(1), С. 5-13.

45. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Development of an application for recognizing emotions using convolutional neural networks, *International Journal of Academic Information Systems Research*, 7(7), pp. 25-36.

46. Pomazan V., Tvoroshenko I., and Gorokhovatskyi V. (2023) Handwritten character recognition models based on convolutional neural networks, *International Journal of Academic Engineering Research*, 7(9), pp. 64-72.

47. Gorokhovatskyi V., Tvoroshenko I. (2023) Identification of visual objects by the search request. *International scientific symposium «INTELLIGENT SOLUTIONS-S». Computational intelligence (results, problems and perspectives). Decision making theory: proceedings of the international symposium*, September 28, 2023, Kyiv-Uzhorod, Ukraine, pp. 25-27.

48. Yakovleva O., Kovač M., Ardasov V. & Yeremenko I. (2023). Study on adding functionality to the Zoom online conference system for monitoring the participant activities, *Public Administration and Regional Development*, 19(1), pp. 158-184.

49. Шахматенко Д. (2023) Дослідження та реалізація методу цифрової ідентифікації з використанням блокчейну, *Abstracts of I International Scientific and Practical Conference «New ways of creating scientific ideas for implementation», (September 18 – 20, 2023). Varna, Bulgaria*, pp. 281-284.