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

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

«На правах рукопису»
УДК _004.056.5:004.891.3_

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
«___»_____2024 р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою
«Інтегровані інформаційні системи»

зі спеціальності 126 «Інформаційні системи та технології»

на тему: «Система захищеної комунікації на основі технології блокчейн»

Виконав:
студент 2 курсу, групи ІА-31мп
Кавун Святослав Ігорович _____

Керівник:
доцент каф. ІСТ, к.т.н., доцент
Писаренко Андрій Володимирович _____

Рецензент:
доцент каф. ІПІ, к.т.н., доцент
Лісовиченко Олег Іванович _____

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

Київ – 2024 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформаційних систем та технологій

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інтегровані інформаційні системи»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Олександр РОЛІК
«___»_____2024 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Кавуну Святославу Ігоровичу

1. Тема дисертації «Система захищеної комунікації на основі технології блокчейн», науковий керівник дисертації Писаренко Андрій Володимирович, к.т.н., доцент, затверджені наказом по університету від «08» 11 2024 р. № 5016-с

2. Термін подання студентом дисертації «09» 12 2024 р.

3. Об’єкт дослідження: система захищеної комунікації.

4. Вихідні дані: відсутність метаданих про комунікації, відсутність передачі персональних даних у відкритому вигляді, використання наскрізного шифрування, зберігання даних у децентралізованій мережі, використання смарт-контрактів.

5. Перелік завдань, які потрібно розробити: дослідження технології блокчейн, огляд існуючих систем захищеної комунікації, проєктування системи захищеної комунікації на основі блокчейну, реалізація та розгортання системи, тестування та аналіз результатів, розроблення стартап-проєкту.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: структурна схема системи захищеної комунікації на основі блокчейну, діаграма послідовності передачі даних в системі на прикладі відправки

повідомлення, схема життєвого циклу транзакції в системі, діаграма залежностей API контексту і компоненту Chat, діаграма прецедентів системи захищеної комунікації, схема трансформації коду Solidity в машинний код і назад, схема процесу шифрування та дешифрування повідомлень в системі, схема мережевої архітектури системи захищеної комунікації на основі технології блокчейн.

7. Дата видачі завдання 02.09.2024 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Дослідження технології блокчейн.	02.09.2024 – 15.09.2024	
2.	Огляд існуючих систем захищеної комунікації та ознайомлення з традиційними підходами до захищеної комунікації.	16.09.2024 – 29.09.2024	
3.	Проєктування системи захищеної комунікації на основі блокчейну.	30.09.2024 – 13.10.2024	
4.	Реалізація та розгортання спроєктованої інформаційної системи.	14.10.2024 – 03.11.2024	
5.	Тестування системи й аналіз ефективності захисту даних.	04.11.2024 – 17.11.2024	
6.	Розроблення стартап-проєкту «Система захищеної комунікації на основі технології блокчейн».	18.11.2024 – 24.11.2024	
7.	Оформлення магістерської дисертації.	25.11.2024 – 06.12.2024	
8.	Попередній захист.	09.12.2024	

Студент Святослав КАВУН

Науковий керівник Андрій ПИСАРЕНКО

РЕФЕРАТ

Система захищеної комунікації на основі технології блокчейн: 105 с., 24 табл., 37 рис., 9 дод., 32 джерел.

БЛОКЧЕЙН, ДЕЦЕНТРАЛІЗАЦІЯ, ШИФРУВАННЯ, БЕЗПЕКА КОМУНІКАЦІЇ, СМАРТ-КОНТРАКТ, КРИПТОГРАФІЯ, РОЗПОДІЛЕНИЙ РЕЄСТР, КОНФІДЕНЦІЙНІСТЬ.

Актуальність. Зростання кіберзагроз і вразливостей сучасних комунікаційних систем вимагає нових підходів до захисту інформації. Технологія блокчейн забезпечує децентралізовану та криптографічно захищену систему передачі даних, що робить її більш стійкою до втручання та атак. Це рішення є особливо важливим у контексті цифровізації бізнесу, розвитку криптовалют і зростання віддаленої роботи. Блокчейн забезпечує прозорість, надійність і безпеку комунікацій, тому його дослідження є важливим для безпечного функціонування сучасного цифрового суспільства.

Метою магістерської дисертації є підвищення безпеки обміну інформацією за допомогою створення системи захищеної комунікації на основі технології блокчейн.

Для досягнення мети були поставлені і вирішені такі завдання:

– дослідити технологію блокчейн, визначити її переваги та недоліки;

– провести огляд існуючих систем захищеної комунікації, їх проблеми та обмеження та ознайомитись з традиційними підходами до захищеної комунікації;

– спроєктувати систему захищеної комунікації на основі блокчейн-мережі;

– реалізувати спроєктовану систему;

– провести тестування системи й аналіз ефективності захисту даних, можливих атак та їх впливу.

Об'єкт дослідження: системи захищеної комунікації.

Предмет дослідження: технологія блокчейн для захищеної комунікаційної системи.

ABSTRACT

Secure communication system based on blockchain technology: 105 p., 24 tab., 37 draw., 9 app., 32 sources.

BLOCKCHAIN, DECENTRALIZATION, ENCRYPTION, COMMUNICATION SECURITY, SMART CONTRACT, CRYPTOGRAPHY, DISTRIBUTED LEDGER, CONFIDENTIALITY.

Relevance. Growing cyber threats and vulnerabilities of modern communication systems require new approaches to information security. Blockchain technology provides a decentralised and cryptographically secure data transmission system, making it more resistant to interference and attacks. This solution is especially important in the context of business digitalisation, the development of cryptocurrencies and the growth of remote work. Blockchain ensures transparency, reliability and security of communications, so its study is important for the safe functioning of the modern digital society.

The purpose of the master's thesis is to improve the security of information exchange by creating a secure communication system based on blockchain technology.

To achieve this goal, the following tasks were set and solved:

– to study blockchain technology, identify its advantages and disadvantages;

– to review existing secure communication systems, their problems and limitations, and to get acquainted with traditional approaches to secure communication;

– to design a secure communication system based on the blockchain network;

– to implement the designed system;

– to test the system and analyse the effectiveness of data protection, possible attacks and their impact.

Object of research: secure communication systems.

Subject of research: blockchain technology for a secure communication system.

ЗМІСТ

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАЧЕНЬ

ABI – Application Binary Interface – інтерфейс, який визначає, як програми взаємодіють із операційною системою або іншими програмами на рівні двійкового коду

API – Application Programming Interface – інтерфейс, який дозволяє програмам обмінюватися даними та функціями між собою

dApp – Decentralized Application – додаток, який працює на децентралізованій мережі, зазвичай на основі блокчейн-технології, без необхідності центрального управління

E2E – End-to-End – підхід, який охоплює повний процес або функціональність системи від початку до кінця, забезпечуючи інтеграцію та перевірку всіх етапів

ETH – Ethereum, Ether – криптовалюта, яка використовується в блокчейн-мережі Ethereum для виконання смарт-контрактів і забезпечення транзакцій

IoT – Internet of Things – мережа фізичних пристроїв, підключених до інтернету, які збирають, обмінюються та обробляють дані для автоматизації та взаємодії

P2P – Peer-to-Peer – модель мережі, в якій учасники взаємодіють безпосередньо один з одним, обмінюючись даними чи ресурсами без посередників

PKI – Public Key Infrastructure – система для управління цифровими сертифікатами та криптографічними ключами, яка забезпечує безпеку обміну інформацією в мережах

SaaS – Software as a Service – модель надання програмного забезпечення через інтернет, в якій користувачі отримують доступ до програм без потреби їх встановлювати, а оплата зазвичай здійснюється за підпискою

Web3 – концепція нового етапу розвитку Інтернету, заснованого на блокчейн-технологіях, децентралізації та використанні смарт-контрактів для забезпечення більших можливостей для користувачів

XSS – Cross-Site Scripting – вразливість безпеки веб-додатків, яка дозволяє зловмиснику вставляти шкідливий код на веб-сторінки, що можуть бути виконані іншими користувачами

ВСТУП

Блокчейн – це одна з провідних технологій нашого століття, яка вже давно вийшла за межі світу криптовалют і знаходить застосування в багатьох різних сферах. Це ланцюжок блоків, які з’єднані між собою, але при цьому не впливають один на одного. Кожен блок зберігає певну інформацію і розміщений у децентралізованій мережі. Записи в блоках містять дані про транзакції, події або інші відомості, які неможливо видалити та навіть змінити. Саме завдяки цьому блокчейн є настільки надійним і прозорим інструментом. У спрощеному вигляді блокчейн можна уявити як базу даних, де інформація має найвищий рівень захищеності, оскільки переписати блок фізично неможливо [1].

Сучасні технологічні досягнення значно спростили спілкування між людьми. Тепер комунікація здійснюється через різні канали, такі як чати, електронна пошта, голосові та відеодзвінки. Проте нові цифрові методи взаємодії також підняли на поверхню важливі питання безпеки комунікації та захисту приватності даних усіх учасників. Водночас для вирішення цих проблем можна використовувати інші передові технології [2]. Структура і характеристики технології блокчейн роблять її перспективною для створення безпечних комунікаційних систем.

Актуальність. Зростання кіберзагроз і вразливостей сучасних комунікаційних систем вимагає нових підходів до захисту інформації. Технологія блокчейн забезпечує децентралізовану та криптографічно захищену систему передачі даних, що робить її більш стійкою до втручання та атак. Це рішення є особливо важливим у контексті цифровізації бізнесу, розвитку криптовалют і зростання віддаленої роботи. Блокчейн забезпечує прозорість, надійність і безпеку комунікацій, тому його дослідження є важливим для безпечного функціонування сучасного цифрового суспільства.

Метою магістерської дисертації є підвищення безпеки обміну інформацією за допомогою створення системи захищеної комунікації на основі технології блокчейн.

Для досягнення мети були поставлені і вирішені такі завдання:

– дослідити технологію блокчейн, визначити її переваги та недоліки;

– провести огляд існуючих систем захищеної комунікації, їх проблеми та обмеження та ознайомитись з традиційними підходами до захищеної комунікації;

– спроєктувати систему захищеної комунікації на основі блокчейн-мережі;

– реалізувати спроєктовану систему;

– провести тестування системи й аналіз ефективності захисту даних, можливих атак та їх впливу.

Об'єкт дослідження: системи захищеної комунікації.

Предмет дослідження: технологія блокчейн для захищеної комунікаційної системи.

Система, що є результатом дослідження в рамках даної роботи, може використовуватись у середовищах із високими вимогами до безпеки, таких як телекомунікації, військова сфера чи міжнародні відносини. Її можна інтегрувати в існуючі інфраструктури комунікацій з мінімальними затратами шляхом використання децентралізованих платформ із відкритим кодом.

Магістерська робота складається з наступних розділів: вступ, основні розділи, висновки, перелік використаних джерел із 32 найменувань, 9 додатків.

1 ТЕХНОЛОГІЯ БЛОКЧЕЙН

1.1 Основні принципи роботи блокчейну

Суть технології блокчейн можна розкрити через поняття «реєстр». Реєстр – це система впорядкування та фіксації будь-якої інформації. Отже, реєстр у своєму первісному вигляді був основою комерційної діяльності з давніх часів і використовувався для запису та зберігання різних даних, переважно про гроші чи власність. Спочатку для запису слугували глиняні таблички, потім – папірус, пергамент і папір. Однак із появою комп'ютерних технологій, які спочатку використовувалися для переведення інформації з паперу в цифровий код, почалася нова ера [3].

Сьогодні завдяки алгоритмам стало можливим створення цифрових розподілених реєстрів, що мають функціональні можливості та властивості, які значно перевершують традиційні паперові та електронні реєстри [3].

Розподілений реєстр являє собою базу даних, що зберігається на кількох вузлах мережі (нодах), де кожен вузол отримує дані від інших і зберігає повну копію реєстру. Такі вузли працюють автономно, оновлюючись незалежно один від одного. Важливою особливістю розподіленого реєстру є його децентралізація, тобто відсутність єдиного центру для зберігання та реєстрації даних. Усі дані у вузлах розподіленого реєстру повинні бути достовірними та актуальними, що досягається через узгодження між усіма вузлами системи. Кожен вузол самостійно створює та записує оновлення до реєстру. Щоб впевнитись, що більшість вузлів погоджується з остаточною версією, вони «голосують» за оновлення. Процес досягнення єдиної версії даних у копії реєстру називається консенсусом. Він автоматично виконується за допомогою спеціального алгоритму консенсусу. Після досягнення консенсусу розподілений реєстр оновлюється, і кожен вузол зберігає фінальну узгоджену версію реєстру [3]. Приклад загальної структури розподіленого реєстру наведено на рисунку 1.1.

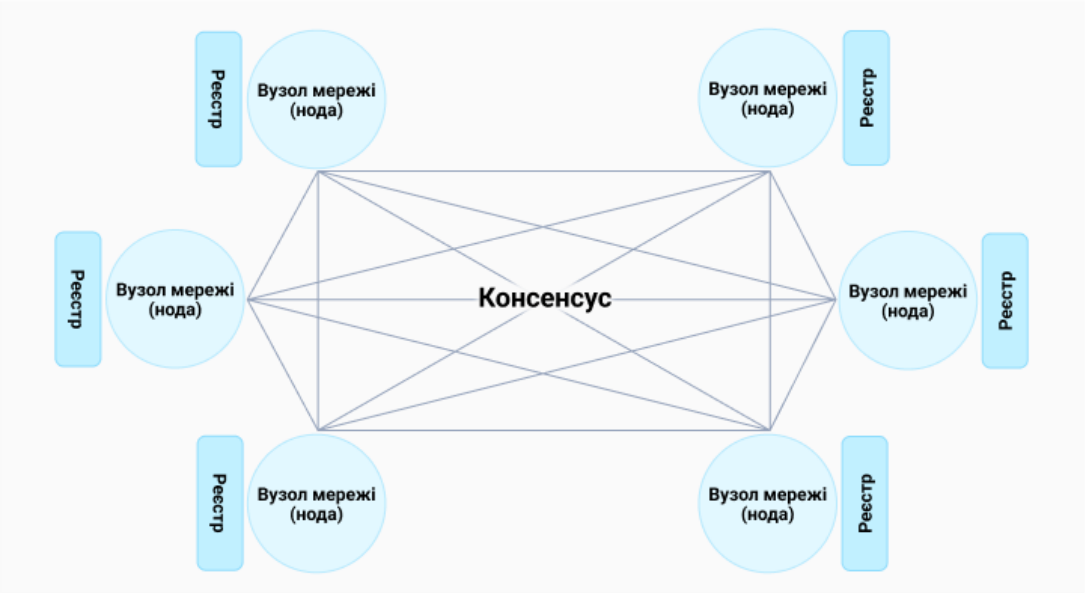

Рисунок 1.1 – Приклад загальної структури розподіленого реєстру [3]

Механізм консенсусу – це те, що забезпечує безпеку децентралізованих мереж. Вузли повинні узгодити поточний стан перед оновленням блокчейну. Цей автоматизований процес допомагає уникнути помилок і захищає мережу від загроз, таких як подвійні витрати, коли цифровий токен використовується більше одного разу – як у випадках навмисного шахрайства, так і через системні збої. Також, він захищає від атаки Sybil, коли зловмисники маніпулюють мережею за допомогою фальшивих вузлів. Розподілений консенсус повністю автоматизований, і він виконується так, як запрограмовано. Таким чином, користувачі покладаються на технологію, а не на третю сторону, для забезпечення цілісності. Як наслідок, логіка і реалізація механізму консенсусу повинні бути бездоганними [4].

Блокчейн-платформи постійно змінюють правила консенсусу, щоб знайти ідеальний баланс між децентралізацією, швидкістю роботи та безпекою. Розглянемо п'ять найпоширеніших практик консенсусу [5].

Практика перша. Доказ роботи (Proof of Work). Це механізм консенсусу, який став основою для всіх інших, залежить від групи валідаторів, що перевіряють транзакції, розв’язуючи складні математичні задачі заради винагороди у вигляді нового блоку. Процес є енергозатратним і передбачає використання спеціалізованих комп’ютерів для знаходження хешу – 64-значного

шістнадцяткового числа, яке формується за допомогою методів криптографії. Майнінг криптовалют, як описаний процес генерації блоків, є популярним способом застосування систем доказу роботи та може приносити значні прибутки у вигляді нових токенів.

Перевагами є висока децентралізація і надійність механізму, що робить його одним із найстабільніших способів верифікації. Наприклад, у випадку з Bitcoin щедра винагорода стимулює активну участь у мережі. Основні недоліки стосуються низької швидкості обробки транзакцій, високих витрат на енергію, комісій та обладнання. Процес також є неефективним у плані екології, оскільки середній час створення блоку для Bitcoin становить 10 хвилин, а енергоспоживання досягає значних обсягів.

Приклади використання: Bitcoin, Dogecoin, Litecoin.

Практика друга. Доказ частки володіння (Proof of Stake). У даній моделі користувачі блокують певну кількість токенів у процесі, що називається стейкінгом, для отримання прав валідатора. Заблоковані токени генерують пасивний дохід і підтримують мережу, доки користувач не розморозить їх для інших цілей, наприклад, обміну. Можливість стати валідатором розподіляється випадковим чином, але шанси зростають пропорційно до кількості «застейканих» токенів. Крім додавання блоків, валідатори зберігають дані та підтримують мережу, а у разі порушення правил консенсусу їхній стейк конфіскується.

Головними перевагами є висока енергоефективність і низькі операційні витрати. Proof of Stake вважається найкращим вибором для масштабування у Web3, оскільки він не потребує великих витрат на обладнання та проведення транзакцій. Однак модель має недоліки: вона менш децентралізована та безпечна порівняно з Proof of Work, а також передбачає залежність впливу користувача від розміру його гаманця, що створює певну нерівність.

Приклади використання: Ethereum, Cardano, Tezos, Algorand.

Практика третя. Делегований доказ частки володіння (Delegated Proof of Stake). Це модифікована версія Proof of Stake, де учасники мережі голосують за

делегатів, використовуючи стейкінг-пули. Делегати, що отримали найбільшу підтримку, отримують права валідаторів, проте можуть бути замінені іншими кандидатами за потреби. Цей механізм зосереджується на репутації делегатів та їхній здатності підтримувати мережу.

До переваг цієї моделі можна віднести її ефективність і демократичність. Цей механізм є вдосконаленою версією моделі Proof of Stake, оскільки забезпечує більшу фінансову доступність для користувачів та створює стимули для валідаторів залишатися відповідальними за стабільність і безперервність роботи мережі. Однак система менш децентралізована порівняно з іншими, а також вимагає активної участі користувачів. Крім того, передача контролю над мережею обмеженій кількості делегатів підвищує ризик атак, зокрема атаки 51%.

Приклади використання: EOS, Lisk, Ark, Tron, BitShares, Steem.

Практика четверта. Доказ авторитету (Proof of Authority). Цей механізм використовується переважно у приватних блокчейнах. Валідатори обираються на основі репутації через процес перевірки, що може включати аналіз минулої діяльності.

Основними перевагами є висока масштабованість і мінімальні витрати на обчислювальні ресурси. Недоліками є зниження рівня децентралізації через концентрацію влади, а також необхідність публічності валідаторів, що зменшує їхню анонімність.

Приклади використання: Xodex, JP Morgan (JPMCoin), VeChain (VET), Ethereum Testnet Kovan.

Практика п’ята. Доказ історії (Proof of History). Даний підхід додає елемент часу до протоколу блокчейна. У процесі верифікації кожен блок отримує часовий штамп, що забезпечує хронологічний запис транзакцій. Proof of History зазвичай працює в поєднанні з іншими механізмами, такими як Proof of Work або Proof of Stake.

Переваги цього механізму включають високу швидкість і безпеку без шкоди для децентралізації, а також низькі витрати на транзакції. Наприклад, Solana, що

використовує цей підхід, вважається найшвидшим блокчейном, зі швидкістю створення блоку в 400 мілісекунд. Недоліками є значне накопичення даних через високу швидкість транзакцій, а також високі вимоги до обладнання, які виключають можливість участі звичайних користувачів у ролі валідаторів.

Приклад використання: Solana [5].

Загалом виділяють декілька типів блокчейну [6]:

– публічний блокчейн – блокчейн, до якого будь-хто в світі може отримати доступ, надіслати транзакцію та очікувати, що вона буде включена до блокчейну, якщо є дійсною. Це означає, що кожен може приєднатися до мережі та взяти участь у процесі досягнення консенсусу без потреби в отриманні дозволів. У такій мережі неможливо цензурувати транзакції або змінювати їх ретроспективно. Однак публічні блокчейни є не дуже ефективними, оскільки для забезпечення довіри потрібно більше обчислювальних потужностей;

– консорціумний блокчейн – блокчейн, де консенсусний процес контролюється попередньо вибраною групою вузлів;

– приватний блокчейн – блокчейн, де доступ до прав на зміну чи навіть читання стану блокчейну обмежений лише кількома користувачами, а участь у мережі дозволена лише відомим вузлам. Як правило, такий тип використовується як внутрішня мережа для організацій. Права на запис є централізованими та контролюються однією організацією. Приватний блокчейн знижує ризик контрагента, дозволяючи обмін даними без посередництва третіх сторін;

– блокчейн з дозволами – блокчейн, де можна дозволяти виконання конкретних дій тільки певними адресами. Учасники мережі можуть обмежити, хто може брати участь у механізмі консенсусу, хто може створювати смарт-контракти, а також надавати повноваження деяким учасникам для валідації блоків транзакцій. Використовується контрольований доступ до вузлів. Блокчейн з дозволами може створювати відчуття більшої безпеки для його власників, проводячи суворі заходи безпеки та конфіденційності для бази даних. Однак це може порушити ідею

блокчейну, оскільки лише деякі учасники мають більше контролю. Це дає їм змогу вносити зміни, навіть якщо інші учасники мережі з цим не погоджуються.

Блокчейн (від англ. block chain – «низка блоків») – це вид розподіленого реєстру, в якому для досягнення консенсусу між вузлами мережі використовується послідовність блоків [3]. Блоки організовані в хронологічному порядку, зв'язані між собою та захищені криптографічними методами, як показано на рисунку 1.2.

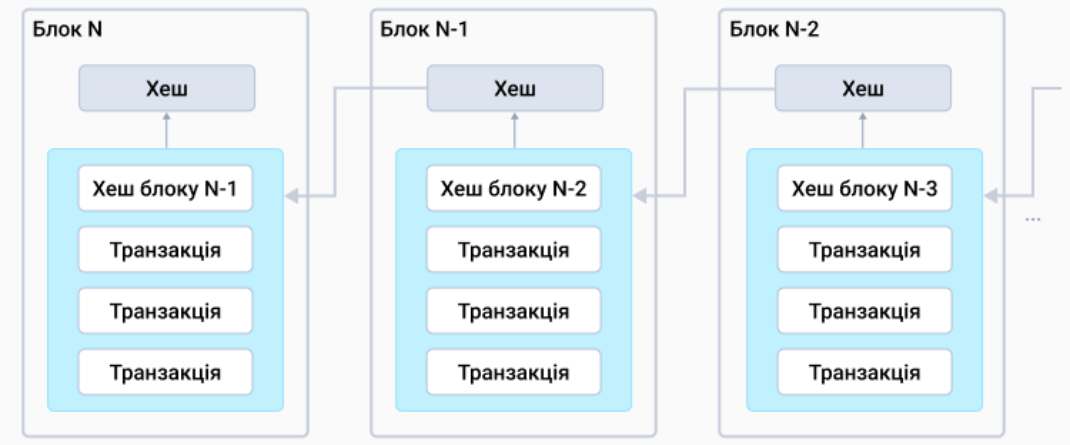

Рисунок 1.2 – Приклад загальної структури та організації блоків [3]

Ці блоки, зв'язані між собою, утворюють ланцюг. Після створення ланцюг не можна змінити. Хеш кожного нового блоку залежить від хешу попереднього, утворюючи таким чином послідовність даних, яку неможливо підробити. Кожен блок ланцюга пов'язаний із сусідніми блоками, хоча вони можуть містити різну інформацію. Якщо хтось спробує змінити дані в одному з блоків, хеш зміниться, і ланцюг буде порушено, що робить підробку практично неможливою [1].

Кожен блок містить корисне навантаження (payload). Це навантаження може включати інформацію про транзакції, операції, контракти, дані, внесені до реєстру щодо фізичних осіб, юридичних осіб, власності тощо. Інакше кажучи, практично будь-яка інформація може бути використана в якості корисного навантаження. Блокчейн являє собою реєстр записів, що постійно оновлюється, в який можна лише додавати дані, а зміна чи видалення інформації в попередніх блоках є неможливою [3, 6].

Отже, можна стверджувати, що технологія блокчейн ґрунтується на кількох фундаментальних принципах, які складають основні переваги блокчейну [3]:

– інфраструктура відкритих ключів (PKI). Блокчейн використовує шифрування з відкритим/приватним ключем та хешування даних для безпечного зберігання та обміну даними;

– блокчейн є розподіленим реєстром і функціонуватиме до останнього активного вузла мережі;

– криптографія. Блокчейн використовує різноманітні криптографічні методи, зокрема хеш-функції, дерева Меркла, а також публічні та приватні ключі;

– усі учасники мережі мають доступ до історії транзакцій блокчейну, і ніхто не має повного контролю над ним;

– у мережі блокчейн відсутня ієрархія, тобто серед вузлів немає головного;

– блокчейн поєднує унікальну комбінацію відкритості та захищеності даних користувачів. Високий рівень надійності забезпечується завдяки просунутим методам шифрування;

– дані в блокчейн-мережі неможливо видалити або замінити, оскільки вони підтверджені великою кількістю вузлів;

– технологія блокчейн забезпечує абсолютну прозорість, адже доступ до інформації про всі транзакції є відкритим, і кожен може перевірити її достовірність;

– мережа блокчейну є «довіреною» системою, оскільки транзакції здійснюються безпосередньо між учасниками, автоматично перевіряються та підтверджуються багатьма вузлами мережі, що усуває потребу в посередниках і повністю нівелює недовіру до однієї центральної організації. У результаті це веде до значного зниження вартості транзакцій завдяки зменшенню комісійних витрат, а також до підвищення швидкості транзакцій за рахунок скорочення часу.

Враховуючи природу та властивості технології блокчейн, можна стверджувати, що вона є механізмом, який забезпечує найвищий рівень зберігання, обліку, передачі та ідентифікації даних. Це робить блокчейн популярною та перспективною технологією практично в усіх сферах [3].

1.2 Смарт-контракти та їх роль у захищеній комунікації

Смарт-контракт – це самовиконуваний контракт, умови якого безпосередньо закодовані в програмному коді. Цей код розгортається в блокчейні, що є розподіленим реєстром, який надійно та послідовно фіксує всі транзакції. Після розгортання контракт автоматично виконує умови без необхідності залучення традиційних посередників, таких як банківські установи або юридичні органи для контролю чи забезпечення виконання угоди [7].

Смарт-контракти – це рішення, які працюють на основі встановленого набору правил, закодованих технологією блокчейн. Ці правила, які зазвичай називаються «логікою» контракту, визначають, як контракт поводитиметься за різних умов. Блокчейн здійснює моніторинг виконання певних умов, відомих як «тригери». Коли ці умови виконуються, контракт автоматично здійснює відповідну дію [7]. Це можна уявити у вигляді умовних операторів: при виконанні певної умови – виконуються відповідні інструкції. Такий підхід забезпечує високий рівень автоматизації, що робить смарт-контракти надзвичайно корисними для процесів, які залучають кілька сторін [8].

Після підписання смарт-контракту всіма залученими сторонами, він зберігається в цифровому форматі у блокчейні і стає чинним, коли виконуються всі передбачені умови [8].

Смарт-контракти дозволяють створювати протоколи обміну даними, які не вимагають попередньої довіри між сторонами. Учасники можуть бути впевнені, що контракт буде виконано лише за умови дотримання всіх передбачених умов. До того ж, використання смарт-контрактів усуває потребу в посередниках, що суттєво знижує витрати на транзакції [9].

Смарт-контракти можуть бути особливо корисними в ситуаціях, коли традиційні правові механізми є надто дорогими або непридатними для застосування [7].

Розглянемо, як класифікуються смарт-контракти за декількома критеріями [10]:

а) середовище виконання:

1) централізовані смарт-контракти (платформа функціонує на централізованій архітектурі з одним валідатором, де реєстр зберігається на єдиному сервері, що знижує рівень децентралізації та стійкості до зовнішніх атак). Прикладом можуть бути контракти мобільних операторів або банків;

2) децентралізовані (створюються у блокчейні);

б) процес постановки завдань і виконання умов смарт-контракту:

1) довільно програмовані (тюрінг-повні);

2) обмежені (тюрінг-неповні);

3) встановлені (суворо типізовані);

в) рівень приватності (стосовно умов смарт-контракту):

1) повністю відкриті;

2) частково відкриті;

3) повністю конфіденційні.

Для того щоб смарт-контракти на базі блокчейну стали широко застосовуваними в реальному житті, потрібно забезпечити відповідні умови для їх впровадження [10]:

– потрібне саме середовище – блокчейн, тобто розподілений реєстр, на основі якого буде створено смарт-контракт, а також інша необхідна інфраструктура;

– важливим є поширення криптовалют як засобу оплати та підвищення рівня довіри до них. Смарт-контракти використовують криптовалюту як «паливо» та одиницю розрахунку;

– потрібно розширити коло осіб, які мають доступ до смарт-контрактів і здатні їх використовувати;

– необхідне правове регулювання використання: як і криптовалюти, смарт-контракти підлягають правовому регулюванню, що тільки починає формуватися.

Смарт-контракт і традиційний контракт є угодами, які зобов'язують сторони виконувати певні обов’язки. Проте, між ними існують суттєві відмінності, що наведені у таблиці 1.1 [8].

Таблиця 1.1 – Порівняння традиційного контракту та смарт-контракту

	Смарт-контракт	Звичайний контракт
Носій	Алгоритм або програма	Паперовий документ
Мова виконання	Мови програмування: Solidity, Javascript, RIDE та інші	Юридична
Підстави для виконання	Умови, прописані в смарт-контракті	Закони, нормативи, договори, бажання учасників
Зміни	Після ініціалізації контракту його неможливо змінити	Можна вносити в будь-який момент
Виконання	Прописані в смарт-контракті умови виконуються автоматично всіма учасниками	Кожен учасник сам вирішує, виконувати контракт чи ні, як і за яких умов виконувати
Наслідки невиконання	Штрафи та санкції прописані в смарт-контракті, тому покарання відбувається автоматично	Щоб домогтися виконання або покарати порушника, потрібно звернутися до суду
Посередники	Ті, які прописані в законі	Нотаріуси, юристи, банки, продавці, довірені особи, поручителі та інші
Створення	Потрібна допомога програміста	Потрібна допомога юриста, нотаріуса, тощо
Носії цінності	Цифрові гроші та сертифікати	Документи, готівка та цифрові гроші

З роками смарт-контракти продемонстрували свої позитивні сторони та можливості, які потребують подальшого розвитку.

До переваг можна віднести [7]:

– автоматизація. При використанні смарт-контрактів умови угоди виконуються автоматично при досягненні визначених умов, що знижує потребу в ручному втручанні та мінімізує ризик людських помилок;

– безпека. Для забезпечення безпеки та незмінності транзакцій використовуються методи криптографії. Після розгортання контракту його неможливо змінити, що забезпечує високий рівень довіри та безпеки;

– прозорість. Код і виконання смарт-контрактів є доступними у блокчейні, що дозволяє всім сторонам перевірити умови і результати. Така прозорість знижує ймовірність спорів та підвищує рівень довіри;

– ефективність. Завдяки відсутності посередників, смарт-контракти зменшують час і вартість транзакцій, що робить процеси більш ефективними. Це є дуже важливим для галузей, де швидкість і економічність є критично важливими.

Недоліками є [7]:

– складність. Розроблення смарт-контрактів вимагає спеціальних знань технології блокчейну та мов програмування (наприклад Solidity). Це може бути суттєвою перепоною для тих, хто не володіє технічними навичками;

– необоротність. Смарт-контракт неможливо змінити чи скасувати після його розгортання у блокчейні. Це створює ризик, оскільки будь-які помилки або вразливості, виявлені після розгортання, не можуть бути легко виправлені;

– масштабованість. Із зростанням обсягу транзакцій в блокчейні можуть виникати проблеми з продуктивністю, що впливає на швидкість їх обробки;

– правова невизначеність. Правовий статус і застосовність смарт-контрактів різняться залежно від юрисдикції. У деяких регіонах смарт-контракти можуть не мати юридичної сили, що обмежує їх використання в окремих сферах.

Переваги та потенціал смарт-контрактів дозволяють застосовувати їх у сферах, де необхідно чітко, прозоро та оперативно виконувати рутинні завдання, які не потребують дорогого юридичного супроводу [10].

В той же час, вони можуть бути використані для управління різноманітними операціями – від грошових переказів до складних бізнес-процесів, як-от нарахування заробітної плати чи управління запасами. Завдяки широкому спектру можливостей, технологія смарт-контрактів може використовуватись у найрізноманітніших галузях, таких як [8]:

– оподаткування. Автоматичний розрахунок і збір податків;

– мистецтво та медіа. Відстеження права власності на медіаконтент та твори мистецтва;

– вибори. Автоматизація виборчого процесу та точний підрахунок голосів;

– постачання. Відстеження руху товарів від місця їх вироблення до пункту призначення, забезпечуючи своєчасну та повну оплату товарів і послуг;

– страхування. Управління заявками за страховими полісами. Смарт-контракти допомагають усунути шахрайство та зменшити витрати шляхом автоматизації процесів, таких як перевірка заявок і розподіл виплат;

– інтернет речей (IoT). Автоматизація управління IoT-пристроями, зокрема їх розташуванням, функціоналом та станом;

– охорона здоров’я. Обмін та управління даними пацієнтів, включаючи медичну історію, інформацію про страхування та інші персональні дані.

На сьогодні використання смарт-контрактів у повсякденному житті обмежене технічними, правовими та соціальними факторами. Втім, уже очевидно, що ця технологія має значний потенціал [10].

1.3 Застосування блокчейну для захисту даних

Завдяки структурі блокчейну досягається високий рівень захищеності інформації, оскільки будь-яка спроба змінити вміст блоку вимагає обчислювально

нереалістичних змін у всіх наступних блоках ланцюга. У результаті дані в блокчейні є надзвичайно захищеними, що ускладнює доступ для хакерів та гарантує цілісність інформації.

Однією з основних переваг блокчейну є прозорість. Дані про транзакції, записані у блокчейні, доступні всім учасникам мережі, що забезпечує високий рівень довіри. Завдяки цій відкритості кожен учасник може перевірити точність інформації, що підвищує надійність системи. Крім того, децентралізована структура блокчейну практично унеможливлює фальсифікацію даних, адже для цього потрібні одночасні зміни на всіх вузлах мережі.

Головною особливістю блокчейну є його незмінність. Записи, внесені до блокчейну, не можуть бути змінені або видалені, що має критичне значення у таких сферах, як фінанси, логістика та право, де точність і достовірність даних є надважливими [1, 11].

Висновки до розділу 1

У першому розділі розглянуті основні принципи роботи технології блокчейн, її структура та типи. Розглянуті механізми валідації транзакцій в мережі, які називаються механізмами консенсусу, описані найпоширеніші з них, визначені їх основні переваги та недоліки та наведені приклади використання відповідних механізмів у реальних мережах.

Також, розглянуті смарт-контракти, їх класифікації, їх переваги і недоліки та їх роль у блокчейн-мережах для досягнення захищеної комунікації. Виконане порівняння з традиційними контрактами, що представлене у вигляді таблиці 1.1. Після цього розглянуті галузі, в яких можна використовувати смарт-контракти.

Наостанок, розглянуте використання технології блокчейн в контексті захисту даних.

2 ОГЛЯД ІСНУЮЧИХ СИСТЕМ ЗАХИЩЕНОЇ КОМУНІКАЦІЇ

2.1 Традиційні підходи до захищеної комунікації

В загальному сенсі, безпечна комунікація – це коли два суб'єкти спілкуються і не хочуть, щоб їх підслуховувала третя сторона. Для цього вони повинні спілкуватися таким чином, щоб їх не можна було підслухати чи перехопити. Безпечна комунікація включає засоби, за допомогою яких люди можуть обмінюватися інформацією з різним ступенем впевненості, що треті сторони не зможуть перехопити сказане. За винятком усного спілкування віч-на-віч без можливості підслуховування, можна з упевненістю сказати, що жодна комунікація не є безпечною в цьому сенсі, хоча практичні перешкоди, такі як законодавство, ресурси, технічні проблеми (перехоплення і шифрування), а також сам обсяг комунікації, обмежують можливості стеження [12].

Захищена комунікація – це метод передачі даних та інформації між двома або більше суб'єктами, який запобігає несанкціонованому доступу, підслуховуванню або перехопленню. Така комунікація спирається на різні технології та практики, покликані забезпечити безпеку даних від початку до кінця.

Основними аспектами захищеної комунікації є конфіденційність, цілісність та автентифікація. Конфіденційність гарантує, що дані, які надсилаються, будуть доступні тільки для передбачуваного одержувача. Зазвичай це досягається за допомогою шифрування, коли інформація перетворюється на код, який може розшифрувати лише той, хто має правильний ключ.

Цілісність означає, що дані захищені від зміни або підробки під час передачі. Якщо повідомлення перехоплено і змінено, перевірка цілісності (наприклад, хешування) може виявити ці зміни, гарантуючи, що отримана інформація є саме тією, яку було надіслано.

Аутентифікація полягає у перевірці автентичності сторін, які беруть участь у комунікації. Це гарантує, що людина або система, з якою ви спілкуєтеся, дійсно є

тією, за кого себе видає. Аутентифікація може бути досягнута за допомогою паролів, біометричної верифікації або криптографічних ключів [13].

Безпечний зв'язок означає постійну доступність, цілісність і конфіденційність мережі. Найкращими стратегіями безпечного зв'язку, що використовуються організаціями, які хочуть захистити свої дані є [2]:

– фізична безпека. Хоча більшість комунікацій сьогодні відбувається через Інтернет, сервери є головними компонентами комунікаційної системи. В ідеалі, сервери повинні бути розташовані в закритому приміщенні з обмеженим доступом. Організації, що піклуються про безпеку зв'язку, часто обирають локальне розгортання будь-якого сервісу, щоб забезпечити максимальну безпеку;

– мережа та архітектура системи зв'язку. Надійність будь-якої комунікаційної мережі значною мірою залежить від безперервного та безпечного потоку. Щоб забезпечити це, мережа повинна складатися з автономних блоків, які можуть працювати незалежно для забезпечення безперебійного зв'язку;

– запобігання несанкціонованому доступу. Для забезпечення безпеки комунікації необхідно запровадити суворі заходи контролю доступу. Конфіденційна інформація, включаючи ім'я користувача та його особисті дані, не повинна бути доступною навіть для співробітників. Багатофакторна автентифікація – це один із способів забезпечити безпечне спілкування між людьми, щоб ніхто не міг підслуховувати, викрадати дані або поширювати дезінформацію;

– шифрування даних при передачі. Дані, що передаються через ненадійну мережу, таку як Інтернет, є найбільш вразливими під час транзиту. Тому дуже важливо запровадити захисний механізм, наприклад, наскрізне шифрування (E2E). Це дозволяє безпечно передавати дані між двома сторонами, запобігаючи будь-якому втручанню з боку несанкціонованих сторонніх користувачів. Криптографічний ключ розшифровує повідомлення, коли воно досягає одержувача. Для безпеки комунікації також важливо захистити управління цими криптографічними ключами;

– контроль адміністратором. Існує необхідність періодичної перевірки доступу співробітників і контролю допуску, щоб уникнути витоку даних або поширення дезінформації;

– регулярний аудит. Аутсорсинг аудиту безпеки надійною та відповідною вимогам третьою стороною може бути корисним для забезпечення комунікаційної безпеки;

– внутрішні тренінги. Протоколи безпеки можуть не спрацювати, якщо люди не дотримуються стандартних практик безпечної комунікації. Внутрішнє навчання може допомогти в перевірці інформації та уникненні кібератак;

– обережне використання сторонніх систем. Для належної роботи комунікаційних сервісів потрібні метадані для кожної комунікації. Детальна інформація про комунікацію, включно з тим, хто, коли, де і як її здійснював, може бути зібрана та збережена. Постачальник послуг повинен повідомляти про мету кожного зібраного фрагмента інформації.

2.2 Використання криптографії в сучасних комунікаційних системах

Криптографія – це мистецтво та наука захисту інформації шляхом перетворення її у формат, що не піддається прочитанню. Таке перетворення гарантує конфіденційність повідомлення між відправником і отримувачем, унеможливлюючи несанкціонований доступ. Історично криптографія відігравала важливу роль у забезпеченні безпеки комунікацій. Від стародавніх цивілізацій, таких як Єгипет, де використовувалися прості підстановочні шифри, до складних методів, розроблених під час Другої світової війни, ця галузь зазнала значної еволюції. Наприклад, машини «Енігма» були незамінними для шифрування повідомлень, підкреслюючи історичну важливість криптографії. До основних термінів криптографії належать: шифрування, дешифрування, відкритий текст, шифротекст, ключ та алгоритм. Шифрування – це процес перетворення відкритого тексту, тобто оригінального зрозумілого повідомлення, на шифротекст, закодовану

версію, яка приховує початковий зміст. Дешифрування, у свою чергу, – це процес відновлення відкритого тексту із шифротексту. Ці процеси гарантують, що навіть якщо неавторизована особа перехопить повідомлення, вона не зможе зрозуміти його без належного ключа. Ключ – це інформація, яка використовується під час шифрування та дешифрування, а алгоритм – це набір правил, що визначають, як саме виконуються ці процеси [14].

Криптографія вивчає математичні методи, які дозволяють досягати або надавати такі цілі чи послуги [15]:

– конфіденційність. Це послуга, що забезпечує доступ до інформації лише тим, хто має на це дозвіл. Вона включає захист усіх даних користувача, що передаються між двома точками протягом певного часу, а також захист трафіку від аналізу;

– цілісність. Це послуга, яка гарантує, що активи комп'ютерної системи та передана інформація можуть бути змінені лише авторизованими користувачами. Зміни можуть включати запис, редагування, зміну статусу, видалення, створення або затримку/повторну передачу повідомлень. Важливо зазначити, що цілісність стосується активних атак, тому вона більше пов'язана з виявленням, а не запобіганням. Крім того, цілісність може бути забезпечена з відновленням або без нього, при цьому перший варіант є більш пріоритетним;

– аутентифікація. Це послуга, яка забезпечує правильне ідентифікування джерела повідомлення. Тобто інформація, що передається через канал, повинна бути аутентифікована за походженням, датою відправлення, змістом, часом надсилання тощо. З цієї причини ця послуга поділяється на два основні типи: аутентифікація сутностей і аутентифікація джерела даних. Варто звернути увагу, що другий тип аутентифікації автоматично забезпечує цілісність даних;

– невідмовність. Це послуга, яка запобігає тому, щоб відправник або отримувач могли відмовитися від своїх попередніх зобов'язань чи дій.

Ці послуги безпеки надаються за допомогою криптографічних алгоритмів, які можуть бути як алгоритмами з симетричним ключем (Private-Key), так алгоритмами з відкритим ключем (Public-Key) [14].

Реалізація криптографічних систем висуває низку вимог. По-перше, критично важливою є продуктивність алгоритмів. Шифрувальні алгоритми мають працювати на швидкостях, відповідних пропускній здатності комунікаційних каналів. Повільна робота криптографічних алгоритмів призводить до незадоволення користувачів і незручностей. Водночас висока швидкість шифрування може супроводжуватися значними витратами, оскільки традиційно досягнення таких швидкостей забезпечувалося за рахунок використання спеціалізованих апаратних пристроїв [15].

Окрім вимог до продуктивності, забезпечення безпеки є ще складнішим завданням. Шифрувальний алгоритм, що працює на універсальному комп’ютері, має обмежену фізичну безпеку, оскільки більшість операційних систем не можуть ефективно забезпечити захищене зберігання ключів у пам’яті. З іншого боку, апаратні пристрої для шифрування можуть бути фізично захищені, що ускладнює втручання зловмисників у систему. Таким чином, спеціалізоване апаратне забезпечення стає основним вибором для розробників протоколів безпеки. Однак апаратні рішення мають відомі недоліки, такі як знижена гнучкість та потенційно високі витрати. Ці обмеження особливо помітні у випадках, коли системи безпеки розробляються на основі нових парадигм протоколів захисту [15].

Сучасні протоколи безпеки часто відокремлюють вибір криптографічного алгоритму від самого процесу проєктування протоколу. Вибір алгоритму для конкретної захищеної сесії зазвичай узгоджується безпосередньо між користувачами протоколу. Нові пристрої, що підтримують такі застосунки, мають бути не лише сумісними з окремими криптографічними алгоритмами та протоколами, але й здатними гнучко працювати з різними алгоритмами, обираючи найбільш підходящий для конкретної задачі. Наприклад, стандарт безпеки для Інтернету IPSec підтримує кілька алгоритмів шифрування, таких як DES, 3DES,

Blowfish, CAST, IDEA, RC4 і RC6. Завдяки цій різноманітності програмні системи виглядають більш доцільним вибором через їхню гнучкість [15].

2.3 Огляд аналогів систем захищеної комунікації

Будь-яке випадкове чи ненавмисне зловживання даними може завдати значної шкоди. Тому варто приділяти більше уваги безпеці комунікацій [2].

Більшість популярних додатків для обміну повідомленнями належать технологічним гігантам, бізнес-моделі яких значною мірою базуються на зборі даних користувачів. Таким чином, особиста комунікація може стати джерелом даних для рекламних алгоритмів або опинитися під загрозою витоку через недоліки безпеки чи несанкціонований доступ. Через це все більше користувачів обирають системи, орієнтовані на конфіденційність, щоб повернути контроль над своїми даними [16].

Розглянемо деякі з найбільш захищених комунікаційних платформ.

Signal – це додаток для обміну миттєвими повідомленнями та SMS із наскрізним шифруванням. Він дозволяє користувачам відправляти особисті або групові повідомлення, фотографії та голосові повідомлення на різних пристроях [16]. Приклади інтерфейсу для ПК та мобільних пристроїв наведені на рисунках 2.1 та 2.2 відповідно.

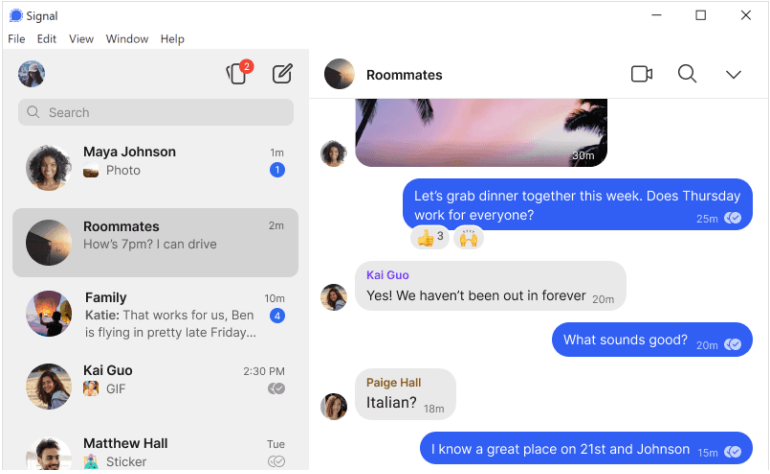

Рисунок 2.1 – Інтерфейс Signal для ПК [17]

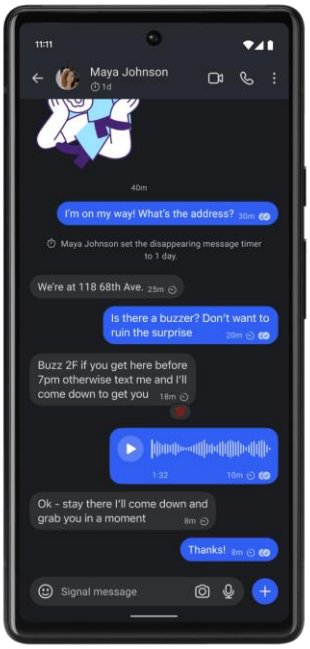

Рисунок 2.2 – Інтерфейс Signal для мобільних пристроїв [17]

Основною перевагою Signal перед аналогічними додатками є його орієнтація на безпеку та конфіденційність. Серед основних функцій безпеки можна виділити наступне [16]:

– коли двоє або більше користувачів Signal починають розмову, використовується наскрізне шифрування. Це означає, що жоден посередник, включаючи саму компанію Signal, не може прочитати повідомлення;

– вихідний код Signal є відкритим, що дозволяє незалежним аналітикам із усього світу перевіряти його роботу та оцінювати рівень безпеки. Вони також можуть повідомляти розробників Signal про знайдені помилки, сприяючи вдосконаленню додатку. Хоча теоретично існує ризик, що компанія може використовувати код, відмінний від опублікованого, відкритість коду створює високий рівень довіри до безпеки додатку;

– Signal є неприбутковою організацією, що мінімізує стимул збирати та монетизувати дані користувачів.

Ось кілька недоліків, які варто врахувати перед використанням [16]:

– Signal вимагає від користувачів введення номера телефону для пошуку контактів та забезпечення унікальності акаунтів. Деякі експерти висловлюють

занепокоєння щодо використання номерів телефону, оскільки це потенційно може бути способом ідентифікації користувачів;

– наскрізне шифрування є потужною функцією, однак, воно також може мати певні обмеження, до яких користувачам доведеться адаптуватися;

– наскрізне шифрування працює лише тоді, коли обидва учасники розмови використовують Signal. В додатку є можливість надсилати повідомлення будь-кому зі списку контактів, але для максимального рівня безпеки співрозмовник також має використовувати цей додаток;

– тільки співрозмовники мають доступ до наскрізно зашифрованої розмови. Якщо вони втратять доступ до чату, його відновлення стане неможливим навіть для команди Signal;

– через фокус на безпеці та конфіденційності Signal може не мати деяких розважальних функцій, які доступні в інших месенджерах. Такі функції, хоч і зручні, можуть створювати вразливості, які розробники Signal вирішили свідомо уникнути.

Wickr Me – це додаток для захищеної комунікації, створений для забезпечення приватності користувачів. Він забезпечує анонімність та повністю наскрізне шифрування. Приклади інтерфейсу Wickr Me наведені на рисунку 2.3.

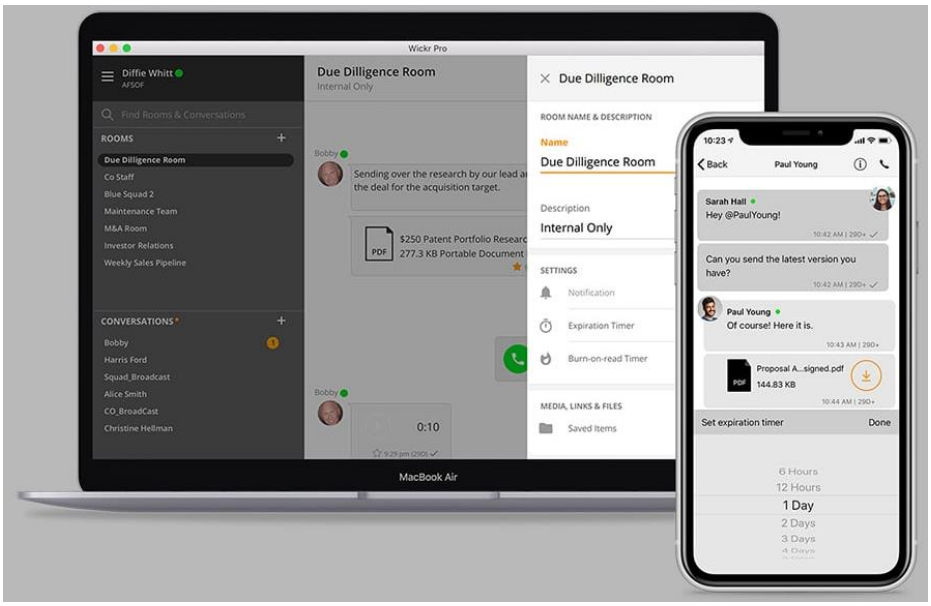

Рисунок 2.3 – Інтерфейс Wickr Me [18]

Його основними характеристиками та функціями є [19]:

– додаток використовує протоколи шифрування (включаючи наскрізне шифрування), що гарантує конфіденційність повідомлень та захищеність від несанкціонованого доступу;

– Wickr Me зосереджений на захисті приватності користувачів. Додаток не зберігає особисті дані, повідомлення або контакти на своїх серверах;

– користувачі можуть встановлювати таймер для повідомлень – від кількох секунд до кількох днів. Після завершення таймера повідомлення автоматично видаляються з пристроїв як відправника, так і отримувача;

– для використання Wickr Me не потрібно надавати особисту інформацію, що дозволяє спілкуватися без розкриття особистості;

– Wickr Me працює на різних пристроях, дозволяючи користувачам безперешкодно перемикатися між смартфонами, планшетами та комп’ютерами;

– користувачі можуть безпечно надсилати файли різних типів, включаючи зображення, документи та відео, безпосередньо через додаток;

– додаток підтримує групові розмови, дозволяючи кільком учасникам спілкуватися одночасно, зберігаючи конфіденційність;

– Wickr Me забезпечує функціонал, який унеможливлює створення знімків екрану повідомлень, підвищуючи рівень захищеності;

– користувачі можуть змінювати сповіщення, теми та параметри повідомлень відповідно до своїх уподобань.

Недоліками Wickr Me є:

– Wickr Me не підтримує хмарне збереження даних. У разі втрати доступу до пристрою користувач ризикує втратити всі повідомлення та інші дані;

– функція автоматичного видалення повідомлень підвищує рівень конфіденційності, але може спричинити випадкову втрату важливих даних, якщо користувачі не встигли зберегти потрібну інформацію;

– інтерфейс Wickr Me може здатися менш інтуїтивно зрозумілим порівняно з іншими популярними додатками, що створює певні труднощі для нових користувачів та потребує більше часу для адаптації;

– Wickr Me менш поширений порівняно з популярними месенджерами, такими як WhatsApp. Це може знижувати ефективність спілкування з людьми, які не користуються цим додатком.

Висновки до розділу 2

У другому розділі розглянуті традиційні підходи до захищеної комунікації, її аспекти та стратегії безпечного зв’язку. Також, розглянуто як, задяки криптографічним методам, досягається безпека та конфеденційность в сучасних комунікаційних системах.

Після цього, виконаний огляд аналогів систем захищеної комунікації, а саме додатків Signal та Wickr Me. Розглянуті їх головні функції і проаналізовані переваги та недоліки, з чого можна зробити висновок, що розглянуті системи забезпечують високий рівень захисту комунікації завдяки шифруванню, але базуються на централізованій інфраструктурі, що може створювати ризики через єдину точку відмови. Натомість система захищеної комунікації на основі блокчейну усуває ці ризики завдяки децентралізованій архітектурі, забезпечуючи додаткову прозорість та стійкість до зовнішніх атак.

3 ПРОЄКТУВАННЯ СИСТЕМИ ЗАХИЩЕНОЇ КОМУНІКАЦІЇ НА ОСНОВІ БЛОКЧЕЙНУ

3.1 Розроблення архітектури системи

У магістерській дисертації структурна схема системи захищеної комунікації на основі блокчейну представлена на кресленику у додатку Б.

В рамках даної роботи системою захищеної комунікації є так званий децентралізований додаток, побудований на інфраструктурі Web3.

Web3 – це наступне покоління веб-технологій, спрямоване на створення більш користувацько-орієнтованого, прозорого та децентралізованого інтернету. Цей термін асоціюється з технологією блокчейн та охоплює набір протоколів, стандартів і інструментів, які дозволяють користувачам взаємодіяти з децентралізованими додатками (dApps) і цифровими активами в мережі. Web3 надає можливість користувачам здійснювати з'єднання та транзакції безпосередньо між собою, з мінімальною залежністю від централізованих серверів чи посередників. [20].

Децентралізовані додатки (або dApps) – це програмні рішення, які функціонують на базі блокчейн-мережі або однорангової (P2P) мережі комп’ютерів. Замість підпорядкування одній централізованій структурі, dApps розподілені по мережі, де їх управління здійснюється спільно всіма користувачами. Їх особливістю є незалежність від контролю чи втручання з боку централізованих організацій.

Серед переваг таких додатків варто відзначити захист конфіденційності користувачів, відсутність цензури та високу гнучкість у розробці. Однак, існують і певні недоліки, зокрема обмежена масштабованість, складнощі зі створенням зручного інтерфейсу, а також труднощі з внесенням змін до коду [21].

Централізовані та децентралізовані додатки мають принципово різну архітектуру та спосіб управління. Централізований додаток має єдиного власника, який повністю контролює його функціонування. Прикладне програмне

забезпечення для такого додатка розташоване на одному або кількох серверах, які перебувають під управлінням цієї компанії чи організації. Користувачі взаємодіють із додатком, завантажуючи його копію на свій пристрій. Дані при цьому надсилаються та отримуються через центральний сервер, який виступає основним вузлом обробки інформації [21].

Децентралізований додаток, навпаки, функціонує на основі блокчейн-мережі, без необхідності централізованого сервера. Управління такими додатками розподілене серед усіх учасників мережі, що забезпечує їх незалежність від контролю однієї організації [21].

Варто зазначити, що децентралізовані додатки можуть взаємодіяти лише з тими блокчейн мережами, які підтримують смарт-контракти. Функціональність смарт-контрактів залежить від архітектури та можливостей кокнретного блокчейну. Деякі, такі як Ethereum, Binance Smart Chain чи Solana, створені з урахуванням підтримки смарт-контрактів. Натомість інші орієнтовані виключно на збереження транзакцій у реєстрі (наприклад, Bitcoin у своїй початковій концепції).

В даній роботі в якості блокчейн-мережі було обрано Ethereum, яка є публічною. Ethereum найбільш відома серед інвесторів завдяки своїй нативній криптовалюті – ether (ETH), а серед розробників – завдяки широким можливостям використання у блокчейн-розробках. Вона дозволяє відкрито створювати та підтримувати безпечні цифрові реєстри [22].

Для взаємодію додатку з блокчейном використовується 1 смарт-контракт, який включає основні функції системи захищеної комунікації, такі як створення акаунту, додавання контактів, відправка та прочитання повідомлення, отримання списку користувачів системи, та інші. Також, смарт-контракт безпечно управляє ключами для шифрування та дешифрування. Кожна взаємодія з смарт-контрактом, що тягне за собою зміну даних в системі, фіксується у вигляді транзакції. Ці транзакції групуються в дерево Меркла і зберігаються у вигляді блоків у блокчейні.

Для інтеграції смарт контракту виконані етапи, що включають:

– підключення до блокчейн-мережі Ethereum;

– розроблення клієнтської частини програми (Frontend) та API;

– інтеграція з гаманцем (який одночасно виступає і механізмом автентифікації в системі) для можливості здійснення транзакцій та передачі даних;

– написання та ініціалізація смарт-контракту у блокчейні;

– розгортання смарт-контракту.

Беручи до уваги особливості децентралізованих додатків, в системі наявна тільки 1 роль – користувач, який може бути як зареєстрованим, так і незареєстрованим. Взаємодія з системою відбувається через Веб-інтерфейс, доступ до якого здійснюється через веб-браузер.

У магістерській дисертації діаграма прецедентів системи захищеної комунікації на основі технології блокчейн представлена на кресленику у додатку Е.

3.2 Механізми автентифікації та передачі даних в системі

В якості гаманця використовується MetaMask, який одночасно є і механізмом автентифікації в системі. Взаємодія з MetaMask відбувається через його розширення в браузері.

MetaMask – це програмний гаманець, призначений для взаємодії з блокчейном Ethereum і є важливим інструментом для роботи з децентралізованим додатком. Він забезпечує управління ключами облікових записів, відправлення та отримання транзакцій, роботу з токенами, що базуються на Ethereum, а також безпечне підключення до децентралізованих додатків [23].

Кожен гаманець має унікальну адресу, на яку користувачі відправляють невелику кількість ether (токени блокчейн-мережі Ethereum), передаючи таким чином інформацію (повідомлення) іншим користувачам системи. Гаманець містить приватні ключі, які використовуються як паролі для ініціювання транзакцій. [22].

В мережі є два типи облікових записів: облікові записи користувачів системи захищеної комунікації та облікові записи смарт-контрактів. Враховуючи, що в

системі використовується лише 1 смарт-контракт, то присутній лише 1 обліковий запис другого типу. Обидва типи мають баланс токенів, можуть здійснювати транзакції на інші облікові записи та виконувати код іншого смарт-контракту. Ідентифікація облікових записів здійснюється за їхньою адресою [24].

У магістерській дисертації діаграма послідовності передачі даних в системі на прикладі відправки повідомлення представлена на кресленику у додатку В.

Передача даних в системі на прикладі відправки повідомлення від одного користувача іншому виглядає наступним чином:

а) авторизований користувач вводить повідомлення та натискає кнопку для його відправки;

б) MetaMask ініціює транзакцію, відправником якої є поточний користувач системи, а отримувачем є смарт-контракт. В транзакції передається вся необхідна інформація для виконання відповідної функції смарт-контракту;

в) користувач підтверджує транзакцію. MetaMask накладає криптографічний підпис за допомогою приватного ключа користувача і відправляє транзакцію;

г) смарт-контракт отримує транзакцію та виконує код відповідної функції, надсилаючи отримане повідомлення адресату;

д) результат виконання транзакції повертається поточному користувачеві.

Варто зазначити, що передача даних в системі при інших діях користувача, таких як створення акаунту чи додавання користувача до списку контактів, є ідентичною.

У магістерській дисертації схема процесу шифрування та дешифрування повідомлень в системі представлена на кресленику у додатку И.

Код контракту виконується під час надсилання до нього транзакції. При цьому контракт зчитує дані, вказані користувачем у транзакції, і повертає результат виконання. За виконання транзакцій відповідає Ethereum Virtual Machine, середовище виконання транзакцій у мережі Ethereum [25].

3.3 Механізми валідації транзакцій в системі

Транзакції є основою роботи блокчейн-мережі Ethereum. Ethereum є своєрідним середовищем, в якому працює децентралізований додаток, а транзакції дозволяють користувачам взаємодіяти з цим додатком.

В системі, транзакції підписуються криптографічним підписом і представляють дію, ініційовану користувачем захищеної системи для комунікації, наприклад, відправку повідомлення. Одночасно, користувач системи є учасником блокчейн мережі.

Транзакції в мережі поділяються на 2 типи: прості перекази та транзакції смарт контрактів. В системі використовується транзакції виключно другого типу. Вони складаються з таких компонентів, як:

– адреса відправника (адреса поточного користувача системи захищеної комунікації);

– адреса отримувача (адреса смарт-контракту);

– дані, що передаються (параметри функцій для взаємодії зі смарт-контрактом, наприклад, назва функції для відправки повідомлення, адреса отримувача повідомлення та зміст повідомлення);

– цифровий підпис (криптографічний підпис, який створюється за допомогою приватного ключа);

– кількість токенів та ліміт обчислювальних зусиль для виконання транзакції.

У магістерській дисертації життєвий цикл транзакції в системі представлений на кресленику у додатку Г.

Після створення транзакції вона надсилається в мережу Ethereum і проходить перший етап валідації, яку виконує локальний вузол відправника. Дана валідація включає наступні етапи:

– перевірка цифрового підпису;

– перевірка балансу відправника;

– перевірка даних транзакції.

Якщо цифровий підпис виявляється невалідним, у відправника недостатньо токенів мережі для відправки повідомлення або в транзакції не вистачає одного з її основних компонентів, транзакція може бути відмінена на кожному з наведених етапів. Якщо транзакція проходить вищенаведені перевірки, вона транслюється на інші вузли в мережі, які, в свою чергу, також перевіряють її дійсність.

В блокчейн-мережі Ethereum, що використовується, для включення транзакції у новий блок використовується механізм валідації транзакцій (механізм консенсусу) Proof of Stake. Варто зазначити, що включення в блок, його перевірка та підтвердження відбуваються поза межами розробленої системи.

Висновки до розділу 3

У третьому розділі розроблена архітектура системи захищеної комунікації, що ґрунтується на використанні децентралізованого додатку. Децентралізований додаток побудований на інфраструктурі Web3. Завдяки цьому забезпечується контроль над тим, як використовується особиста інформація користувачів, і хто має до неї доступ.

Децентралізований додаток, в свою чергу, функціонує на основі блокчейн-мережі Ethereum. Для взаємодії з Ethereum використовується розроблений смарт-контракт, який включає основні функції системи.

Визначена та описана робота MetaMask, який виступає інтегрованим механізмом автентифікації, управління ключами користувачів, відправлення і отримання транзакцій та безпечного підключення до децентралізованого додатку.

Розроблений процес передачі даних в системі. Визначена та описана роль і робота транзакцій в системі. Обрані та описані механізми валідації транзакцій як в рамках розробленої системи, так і поза нею.

Розроблена архітектура відповідає вимогам безпеки, функціональності та сумісності з сучасними Web3-технологіями, що створюює базу для подальшого вдосконалення системи.

4 РЕАЛІЗАЦІЯ ТА РОЗГОРТАННЯ СИСТЕМИ

4.1 Вибір платформи та інструментів для розроблення

Інфраструктура Web3, на які побудований децентралізований застосунок, складається 4-х компонентів, перші три з яких є основою спроєктованої та реалізованої системи:

– технологія блокчейн;

– смарт-контракти;

– децентралізовані додатки (dApps);

– децентралізовані автономні організації (DAOs).

Реалізована система є веб-застосунком на основі React. Для розроблення інтерфейсу використаний Next.js, що доповнює React, додаючи такі функції, як серверний рендеринг і маршрутизація, що розширило можливості системи.

Смарт-контракт написаний мовою Solidity. Solidity – це об'єктно-орієнтована мова високого рівня для створення смарт-контрактів, яка розроблена спеціально для роботи з віртуальною машиною Ethereum (EVM) [26].

Для розгортання локального блокчейну використаний Hardhat. Це свого роду середовище розробки для Ethereum. Hardhat постачається з вбудованою локальною мережею Hardhat Network – це вузол Ethereum, спеціально розроблений для потреб розробки. Він дозволяє розгортати смарт-контракти, виконувати тести та відлагоджувати код безпосередньо на локальній машині. Цей компонент може працювати як внутрішній процес. Альтернативно, Hardhat Network може працювати і як автономний сервер, до якого можуть підключатися зовнішні клієнти [26]. В системі, Hardhat Network використаний як окремий демон, який обробляє запити JSON-RPC та WebSocket. Зовнішніми клієнтами є фронтенд децентралізованого додатку та цифровий гаманець. Для забезпечення взаємодії між смарт-контрактом і фронтендом використані бібліотеки ethers та web3modal.

Hardhat обраний через те, що він має багато інструментів для підтримки Solidity. Він завжди знає, які смарт-контракти виконуються, що вони роблять і чому

не вдається виконати ту чи іншу дію, надаючи трасування стека Solidity. Приклад виняткової ситуації Hardhat Network, в результаті якої транзакцію було скасовано, наведений на рисунку 4.1.

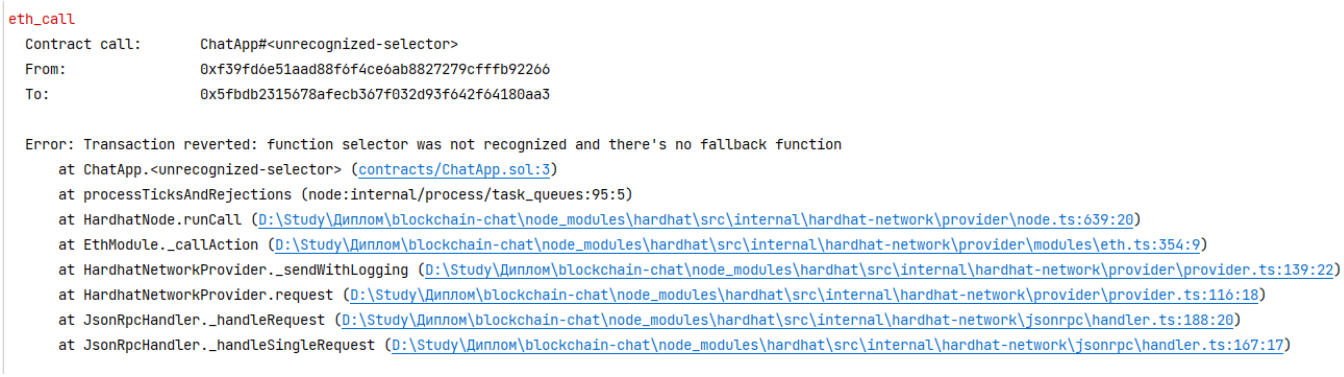

Рисунок 4.1 – Приклад виняткової ситуації Hardhat Network з відміною транзакції

Також, Hardhat Network використовує власну трасувальну інфраструктуру для надання детального логування, що спрощує процес розробки та відлагодження смарт-контрактів. Логи успішного та неуспішного виклику наведені на рисунках 4.2 та 4.3 відповідно.

Рисунок 4.2 – Лог Hardhat Network при успішному виклику функції

Рисунок 4.3 – Лог Hardhat Network при виникненні виняткової ситуації віртуальної машини

Цифровим гаманцем обраний Metamask, оскільки він призначений для взаємодії саме з блокчейном Ethereum. Робота Metamask описана у підрозділі 3.2 даної роботи.

У магістерській дисертації схема мережевої архітектури системи захищеної комунікації на основі технології блокчейн представлена на кресленику у додатку К.

Дана схема ілюструє взаємодію між основними компонентами системи, зокрема клієнтськими пристроями, децентралізованим додатком, MetaMask, локальною Ethereum мережею на базі Hardhat і смарт-контрактом, який відповідає за основні функції системи. Вона демонструє, як компоненти взаємодіють між собою для забезпечення захищеної комунікації.

4.2 Опис коду та технічних рішень

Структура проєкту системи захищеної комунікації на основі технології блокчейн наведена на рисунку 4.4. Варто звернути увагу на такі директорії, як: assets, Components, Context, contracts, pages, scripts, styles та Utils.

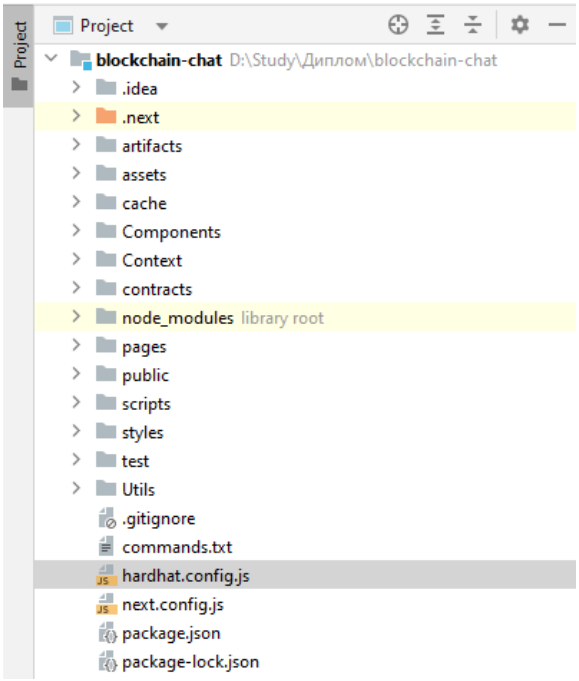

Рисунок 4.4 – Структура проєкту системи захищеної комунікації

Директорія assets містить графічні матеріали, що використовуються у системі. Структура директорії assets наведена на рисунку 4.5.

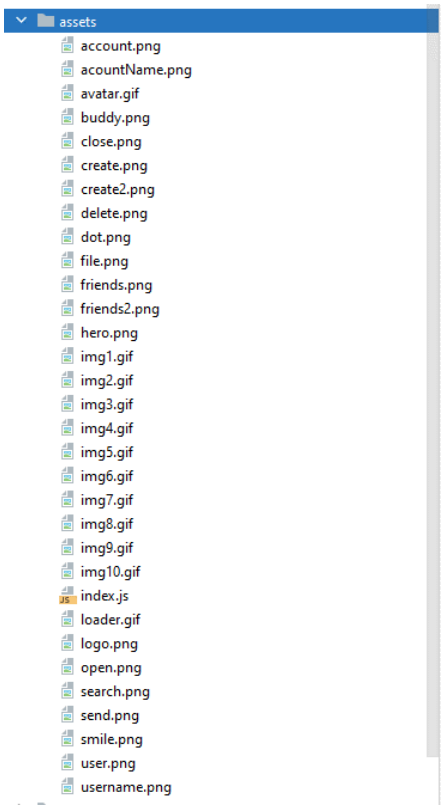

Рисунок 4.5 – Структура директорії assets

Файл index.js використовується для експорту всіх графічних матеріалів для можливості подальшого їх використання всередині компоненів, які зберігаються в директорії Components.

Директорія Components містить компоненти, які використовуються на сторінках. Кожен компонент має відповідні стилі для цього компонента. Для логічного розмежування компонентів використані внутрішні директорії, структура який відображена на рисунку 4.6.

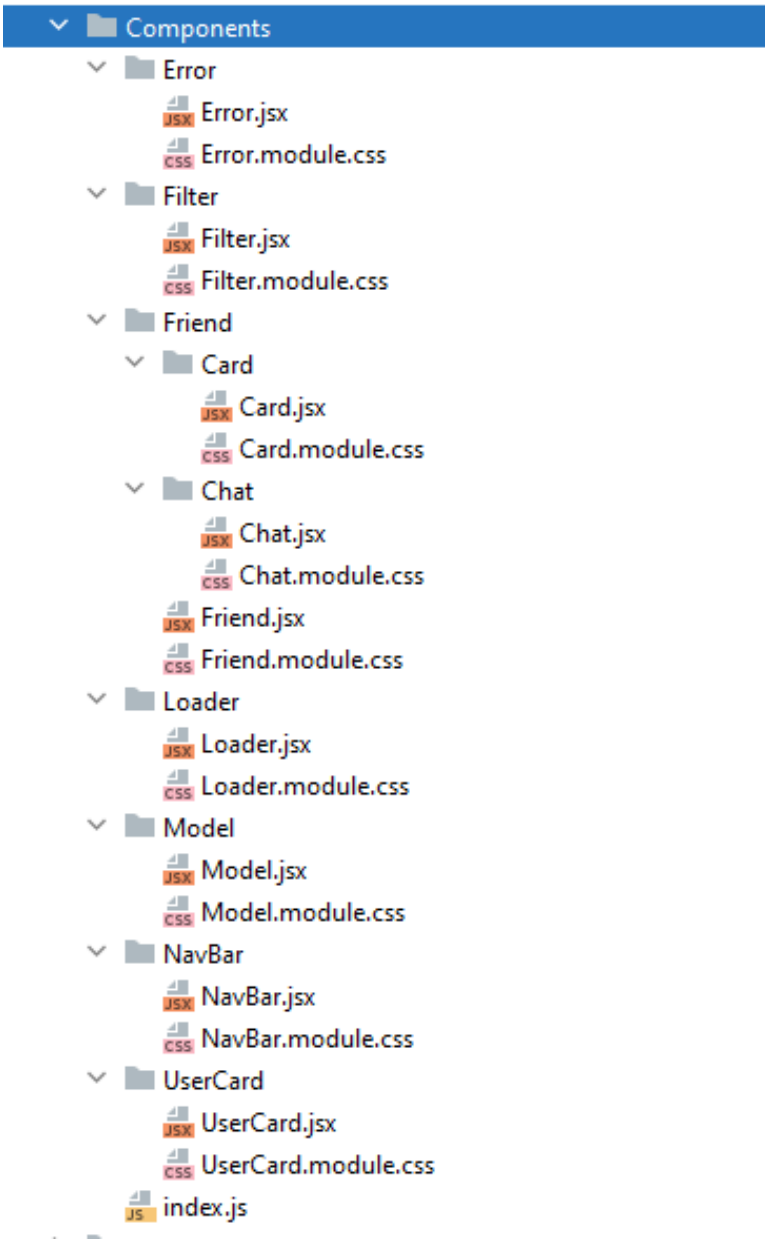

Рисунок 4.6 – Структура директорії Components

Варто зазначити, що компоненти мають залежності. Наприклад, компонент Chat містить в собі компонент Loader для відображення іконки завантаження сторінки замість пустого екрану під час відкриття чатів.

Всередині компонентів викликаються функції, які оголошені у файлі Context/ChatAppContext.js, що дозволяє взаємодіяти зі смарт-контрактом, наприклад, створювати облікові записи, отримувати інформацію про облікові записи, надсилати повідомлення тощо. Також в компоненті Chat використовується допоміжна функція для конвертації шістнадцяткової позначки часу на звичайну дату для відображення точних дати та часу повідомлень.

Директорія Context відповідає за контекст та можливості системи. Структура директорі Context наведена на рисунку 4.7.

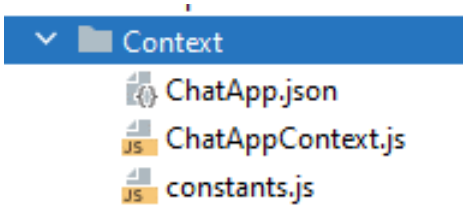

Рисунок 4.7 – Структура директорі Context

ChatApp.json містить ABI смарт-контракту, що є важливою складовою системи. ABI (Application Binary Interface) смарт-контракту – це стандартний інтерфейс для взаємодії з контрактами в екосистемі Ethereum. Він визначає спосіб взаємодії з контрактами як ззовні (людьми), так і між самими контрактами. Після компіляції і розгортання у блокчейні смарт-контракт має вигляд довгого набору цифр та букв, для генерації якого використовується шістнадцяткова та/або двійкова система. ABI допомагає людям перетворювати зрозумілий людині код (в нашому випадку код написаний мовою Solidity) у комп'ютерний код і назад [27].

У магістерській дисертації схема трансформації коду Solidity в машинний код і назад представлена на кресленику у додатку Ж. Дана схема представляє описану вище взаємодію в графічному вигляді.

Якби ABI не було, потрібно було б писати весь код, використовуючи необроблений низькорівневий шістнадцятковий або байт-код.

ABI створюється автоматично під час компіляції смарт-контракту і містить інформацію для кожної функції в межах смарт-контракту, що включає [27]:

– назву функції (для ідентифікації);

– аргументи функції (включає тип, порядок і структуру даних);

– тип повернення (вказує тип даних, який повертає виклик функції);

– події (якщо контракт містить події, ABI також описує їх разом з параметрами, які вони повертають).

Частина ABI, що відповідає функції смарт-контракту для додавання користувача в список контактів представлена на рисунку 4.9. Для порівняння, на рисунку 4.10 представлена ця сама функція в смарт-контракті.

ChatApp.sol | ChatApp.json

```
{
  "_format": "hh-sol-artifact-1",
  "contractName": "ChatApp",
  "sourceName": "contracts/ChatApp.sol",
  "abi": [
    {
      "inputs": [
        {
          "internalType": "address",
          "name": "friendKey",
          "type": "address"
        },
        {
          "internalType": "string",
          "name": "name",
          "type": "string"
        }
      ],
      "name": "addFriend",
      "outputs": [],
      "stateMutability": "nonpayable",
      "type": "function"
    },
```

Рисунок 4.9 – Функція «addFriend» в ABI смарт-контракту

```
function addFriend(address friendKey, string calldata name) external {
    require(userExists(msg.sender), "Create an account first");
    require(userExists(friendKey), "User is not registered!");
    require(msg.sender != friendKey, "Users cannot add yourself as a friend");
    require(areFriends(msg.sender, friendKey) == false, "These users are already friends");

    _addFriend(msg.sender, friendKey, name);
    _addFriend(friendKey, msg.sender, userList[msg.sender].name);
}
```

Рисунок 4.10 – Функція смарт-контракту «addFriend»

Також, у директорії Context розташований згаданий раніше ChatAppContext.js. Це своєрідний API контексту, що містить функціонал для взаємодії зі смарт-контрактом. Наявні такі функції, як отримання інформації про користувача, відправка та прочитання повідомлення, додавання користувача до списку контактів, створення акаунту та фільтрація списку користувачів системи, які зареєстровані в системі, але ще не додані до списку контактів. Кожна з цих функцій, крім виконання допоміжної логіки, в свою чергу викликає відповідну функцію смарт контракту.

ChatAppContext.js використовує деякі допоміжні функції з Utils/apiFeature.js, зокрема:

– функція під’єднання до гаманця Metamask;

– функція перевірки, чи гаманець під’єднаний (щоб у разі негативного результату відобразити користувачу відповідне інформаційне повідомлення, враховуючи, що взаємодія з системою без гаманця не є можливою);

– функція під’єднання до смарт-контракту (дана функція викликається при кожній взаємодії зі смарт-контрактом на зміну даних, що тягне за собою створення транзакції у блокчейні).

У магістерській дисертації діаграма залежностей API контексту і компоненту Chat представлена на кресленику у додатку Д. Дана діаграма показує взаємодію API

контексту з компонентом Chat. Варто зазначити, що взаємодія API контексту з будь-яким іншим з компонентів є ідентичною.

Останнім файлом у директорії Context є constants.js. Він слугує для зберігання важливих констант, таких як, наприклад, Hardhat адреса застосунку у шістнадцятковому представленні. Також він зберігає налаштування мережі для їх імпорту у Metamask гаманець. Налаштування включають:

– ідентифікатор мережі, який Metamask застосовує для підпису транзакцій (chainId);

– ім'я, яке Metamask використовуватиме для ідентифікації мережі (chainName);

– налаштування нативної валюти мережі (nativeCurrecncy);

– адреса, через яку MetaMask підключатиметься до мережі блокчейн (rpcUrl);

Налаштування для локальної мережі представлені на рисунку 4.12.

```
localhost: {
  chainId: `0x${Number( value: 31337).toString( radix: 16)}`,
  chainName: "localhost",
  nativeCurrency: {
    name: "ETH_TST",
    symbol: "ETH_TST",
    decimals: 18,
  },
  rpcUrls: ["http://127.0.0.1:8545/"],
},
```

Рисунок 4.12 – Конфігурація для локальної мережі Metamask

Варто звернути увагу, що ідентифікатор мережі, який застосовується для підпису транзакцій також вказується в конфігураційному файлі Hardhat і він має співпадати з ідентифікатором, налаштованим у Metamask. Конфігураційний файл для Hardhat наведений на рисунку 4.13. В даній конфігурації також вказується точна версія мови Solidity, яка використовується для написання смарт-контракту.

Рисунок 4.13 – Конфігураційний файл Hardhat

Директорія contracts містить лише один файл – ChatApp.sol, що є смарт-контрактом, який написаний мовою Solidity. Всі наявні функції смарт-контракту використовуються API контекстом для взаємодії з Ethereum блокчейн-мережею. До функцій смарт-контракту належать:

– відправка повідомлення користувачу з відповідним публічним ключем;

– читання повідомлення від користувача з відповідним публічним ключем;

– перевірка існування користувача в мережі за його публічним ключем;

– отримання ідентифікатора користувача за його публічним ключем;

– створення нового акаунту з відповідним ідентифікатором та адресою;

– додавання користувача до списку контактів за його публічним ключем;

– перевірка чи є два акаунти друзями в системі за двома публічними ключами;

– отримання списку контактів для поточного користувача;

– отримання ідентифікатору чату за двома публічними ключами користувачів;

– отримання списку всіх користувачів мережі.

4.3 Налаштування та розгортання системи

До початку налаштування системи потрібно встановити NodeJs та менеджер пакетів для мови програмування JavaScript – NPM. Рекомендовані версії для встановлення NodeJs – v18.12.1, NPM – v8.19.2.

Далі, потрібно перейти в корінь проєкту і виконати команду «npm install». Дана команда завантажить та додасть всі необхідні залежності для короткної роботи додатку, включаючи ті, що вказані у файлі package.json, який знаходиться у корені проєкту. Даний файл представлений на рисунку 4.14.

```
{
  "name": "chatapp",
  "version": "0.1.0",
  "private": true,
  "scripts": {
    "dev": "next dev",
    "build": "next build",
    "start": "next start",
    "lint": "next lint"
  },
  "dependencies": {
    "ethers": "^5.7.2",
    "next": "12.3.1",
    "react": "18.2.0",
    "react-dom": "18.2.0",
    "web3modal": "^1.9.9"
  },
  "devDependencies": {
    "@nomicfoundation/hardhat-toolbox": "^2.0.0",
    "hardhat": "^2.12.0"
  }
}
```

Рисунок 4.14 – Файл проєкту package.json

Далі потрібно використати Hardhat для ініціалізації локальної блокчейн-мережі Ethereum. Для того, щоб гаманець Metamask і децентралізований застосунок

могли під’єднатися до мережі, потрібно запустити Hardhat Network в автономному режимі. Для цього виконується команда «npx hardhat node». Це відкриває JSON-RPC інтерфейс для Hardhat Network і дозволяє підключитися до мережі блокчейн за адресою «http://127.0.0.1:8545». Додатково, після виконання вищезгаданої команди виділяється 20 тестових адрес разом з приватними ключами, що зображено на рисунку 4.15. Згодом їх можна буде додати в гаманець Metamask і використовувати для тестування системи.

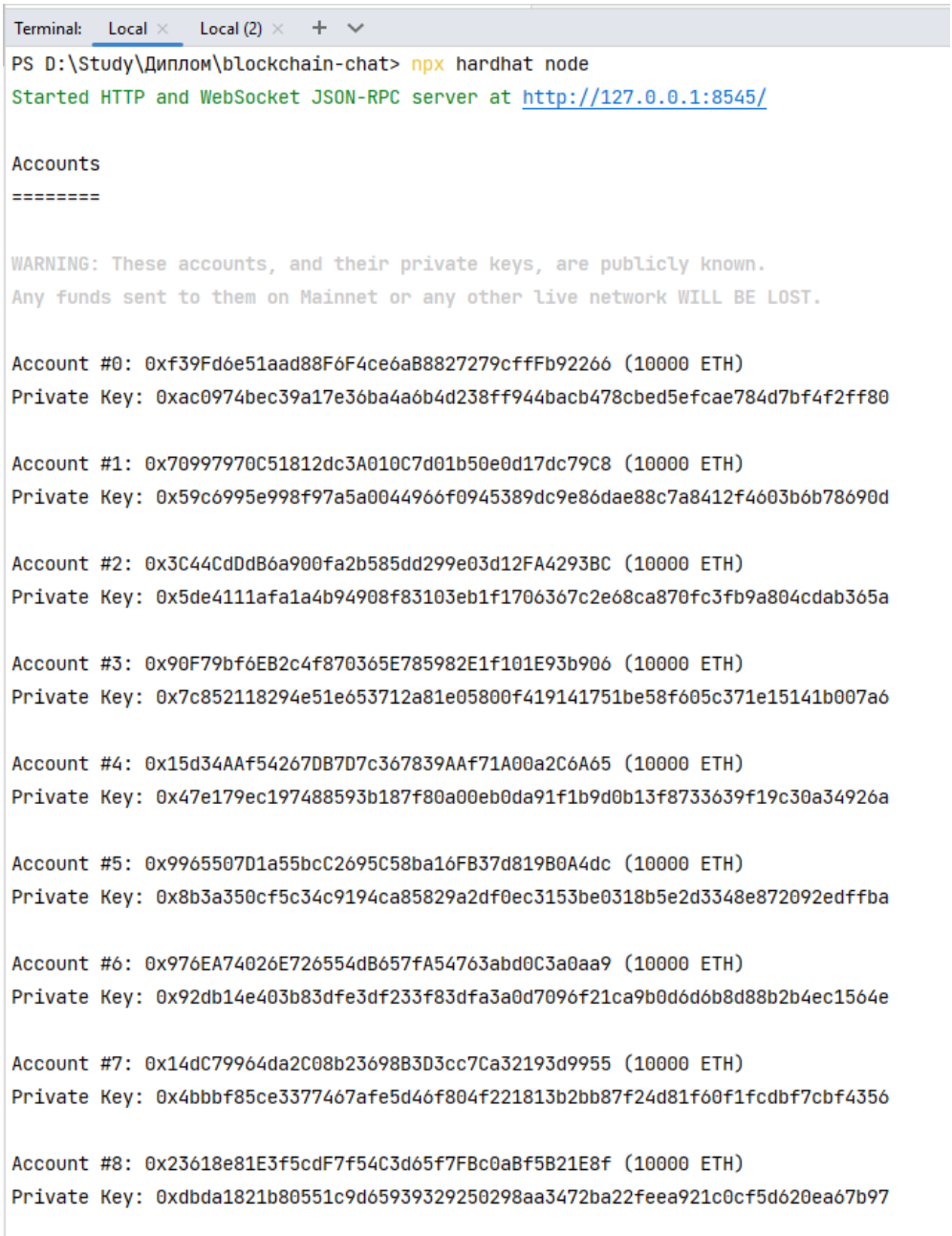

Рисунок 4.15 – Запуск локальної блокчейн мережі за допомогою Hardhat

Після цього в окремому терміналі виконується команда «npx hardhat run --network localhost scripts/deploy.js». Дана команда виконує компіляцію проєкту, після чого використовує скрипт scripts/deploy.js для розгортання смарт-контракту в мережі.

Наступним етапом є запуск сервера Next.js. Для цього використовується команда «npm run dev».

Після цього потрібно перейти в браузер, встановити розширення Metamask, яке доступне для всіх операційних систем та пройти процедуру реєстрації.

Далі потрібно відкрити адресу http://localhost:3000 в браузері. Початковий екран застосунку має вигляд, представлений на рисунку 4.16.

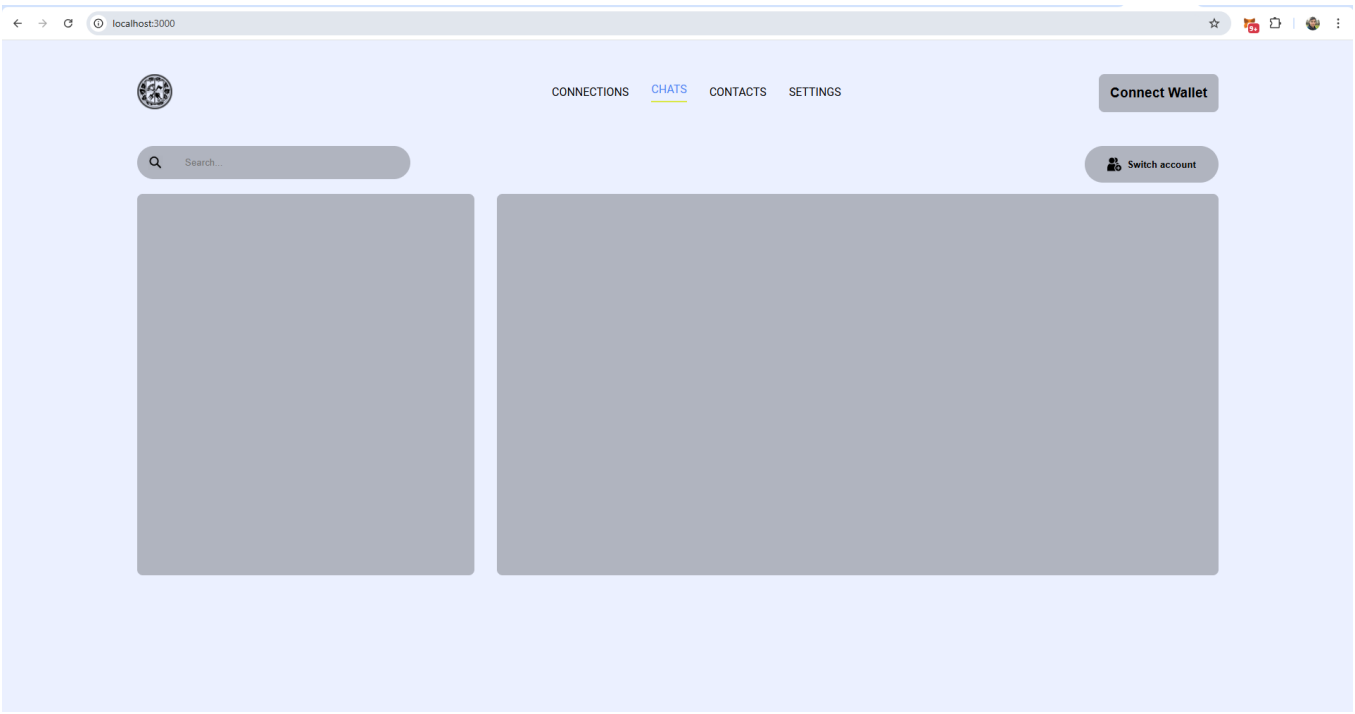

Рисунок 4.16 – Веб-інтерфейс при першому відкритті застосунку

Далі потрібно натиснути кнопку «Connect Wallet» в правому верхньому кутку, після чого дати дозвіл додатку підключатись до Metamask у вспливаючому вікні, як показано на рисунку 4.17.

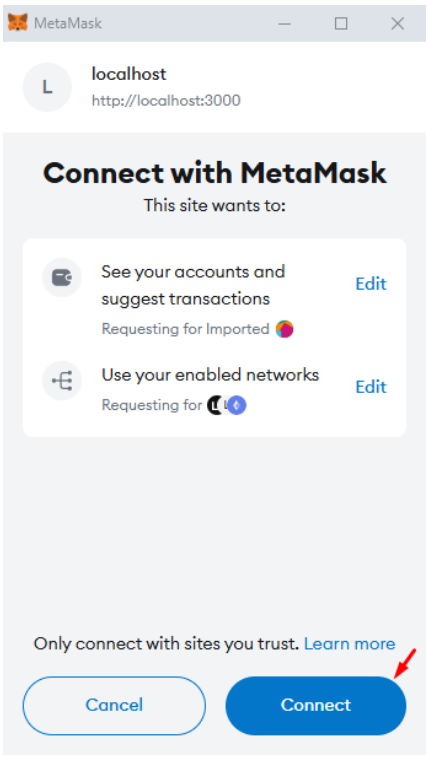

Рисунок 4.17 – Запит на дозвіл додатку під’єднуватись до Metamask

Для завершення процедури налаштування потрібно впевнитись, що Metamask налаштований. Налаштування мережі імпортувались успішно, що показано на рисунках 4.18 та 4.19.

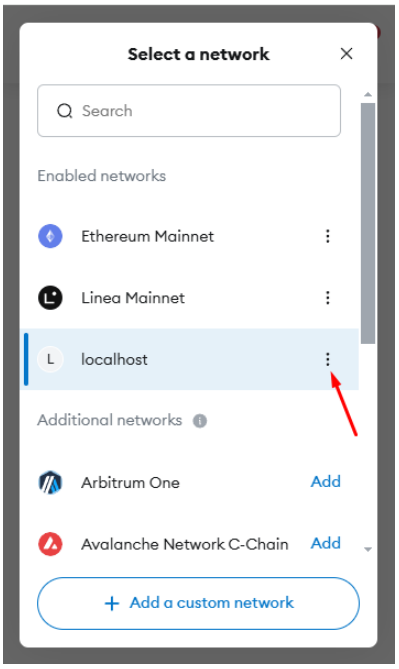

Рисунок 4.18 – Вибір мережі в налаштуваннях Metamask

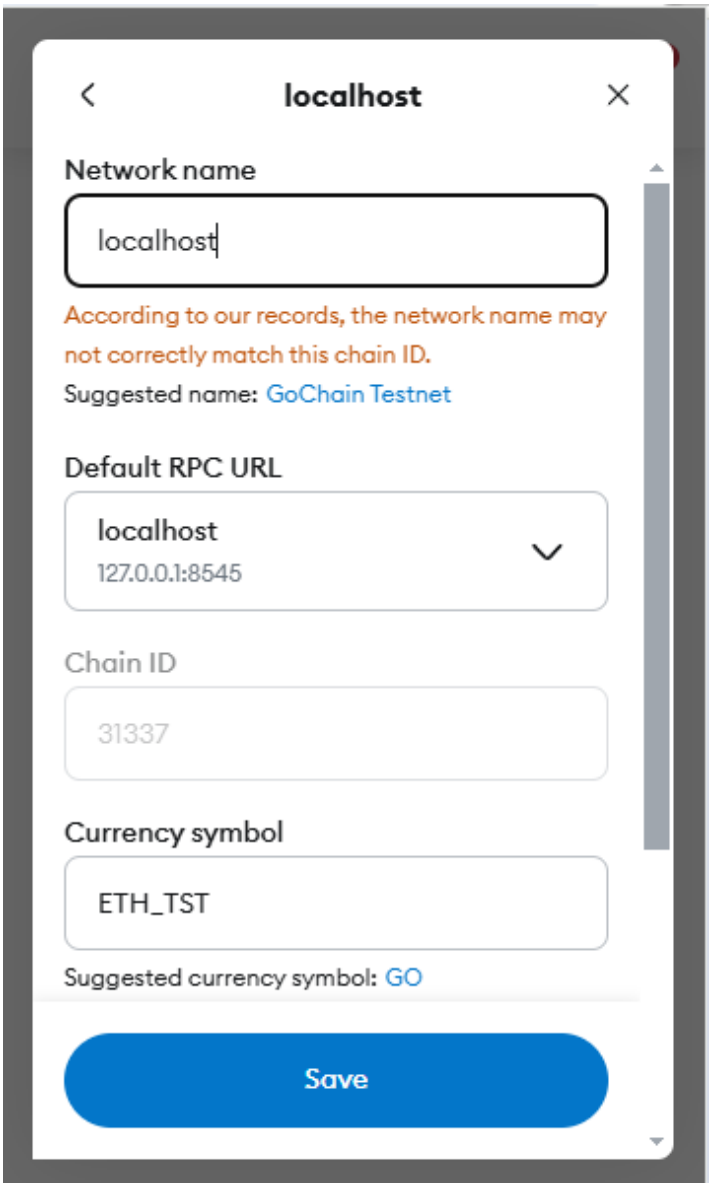

Рисунок 4.19 – Налаштування локальної мережі в Metamask

Якщо налаштування мережі виконані, а повідомлення про помилки відсутні, це означає, що налаштування і розгортання системи завершено.

Висновки до розділу 4

В рамках четвертого розділу реалізована та розгорнута система, спроєктована в третьому розділі.

Розроблений децентралізований додаток побудований на інфраструктурі Web3 на основі React з доповненням Next.js, що розширяє можливості системи. Для написання смарт-контрактів використана мова програмування Solidity. Для розгортання локального блокчейну використаний Hardhat, який має підтримку Solidity.

Також, в рамках даного розділу описаний код та технічні рішення, що використані при розробленні системи. Описана структура, функції системи, конфігурації, робота кожного з компонентів системи та їх взаємодія між собою.

Після цього описаний процес налаштування та розгортання розробленої системи, починаючи з інсталяції необхідних пакетів і завершуючи отриманням остаточно працюючої системи.

5 ТЕСТУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

5.1 Проведення тестування функціоналу системи

За замовчуванням, мережа Hardhat ініціалізується з новим, «чистим» блокчейном, в якому міститься лише так званий блок генезису (genesis block).

Генезисний блок, також відомий як «Блок 0» або «Блок 1», відіграє важливу роль у блокчейні, слугуючи основою для побудови всіх наступних блоків. Це єдиний блок у блокчейні, який не містить посилання на попередній блок. Він інтегрований у програмний код блокчейну як жорстко закодована структура. Враховуючи той факт, що кожен блок у блокчейні містить хеш попереднього блоку, генезисний блок відіграє основну роль у забезпеченні цілісності блокчейну. Цей криптографічний механізм гарантує, що будь-яке втручання або зміна даних призводить до порушення цілісності всієї системи блокчейну [28].

Для початку тестування системи потрібно мати як мінімум 2 акаунти Metamask з певною кількістю токенів мережі для перевірки створення транзакцій. Використовуються акаунти, які генерує Hardhat після ініціалізації, надаючи їх публічні та приватні ключі. Для того, щоб додати акаунт до Metamask, потрібно зайти в меню вибору акаунту та натиснути на відповідну кнопку для додавання нового акаунту, як представлено на рисунку 5.1.

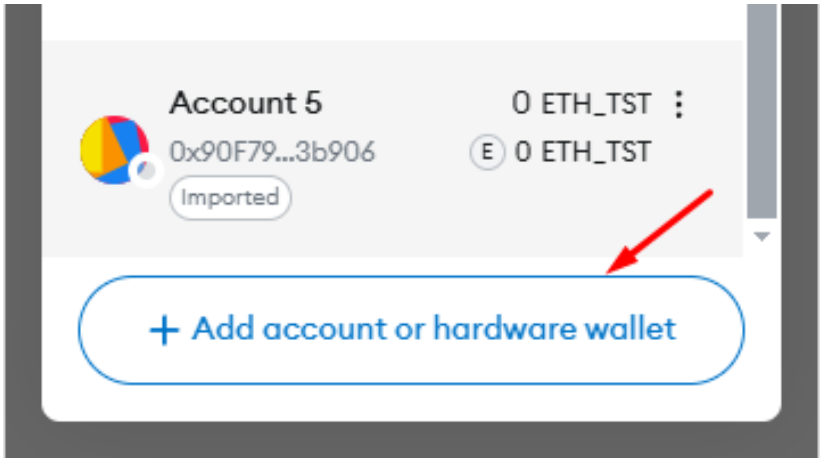

Рисунок 5.1 – Додавання нового акаунту в Metamask

Далі потрібно обрати пункт «Імпорт акаунту», використовуючи в якості типу приватний ключ, як показано на рисунках 5.2 та 5.3.

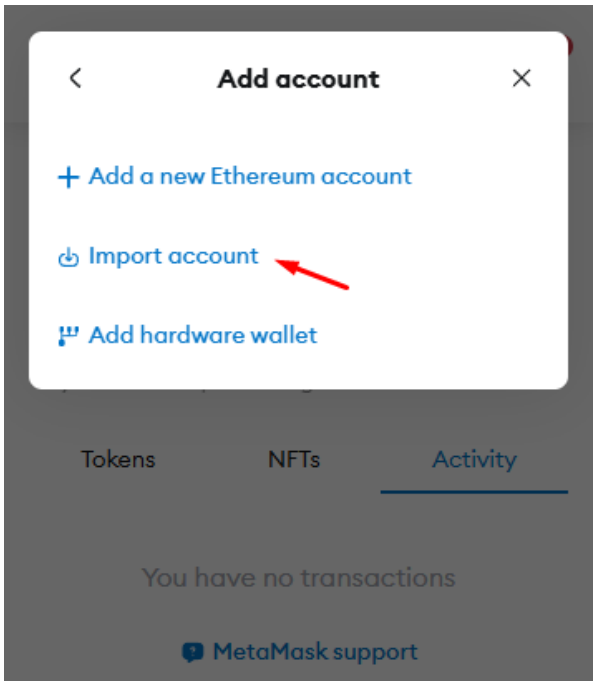

Рисунок 5.2 – Меню додавання нового акаунту в Metamask

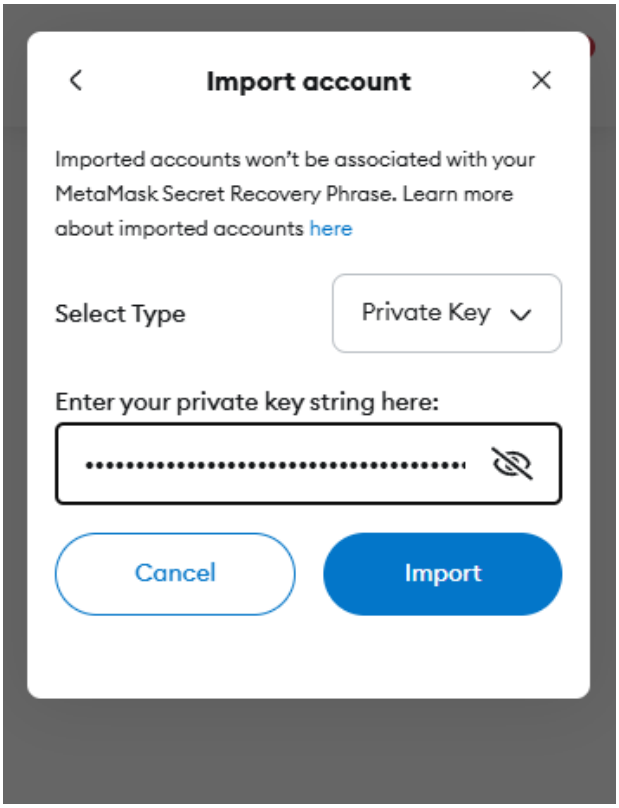

Рисунок 5.3 – Імпорт акаунту в Metamask за допомогою приватного ключа

Таким чином потрібно додати декілька акаунтів, щоб використовувати їх для реєстрації в децентралізованому додатку.

Для реєстрації потрібно натиснути на кнопку «Create Account» в верхньому правому кутку на головній сторінці додатку, після чого відбувається перенаправлення на сторінку реєстрації. Для створення акаунту в системі потрібно вввести ім’я користувача. Необхідність вводити адресу користувача відсутня, адже вона підтягується автоматично з розширення Metamask, що представлено на рисунку 5.4.

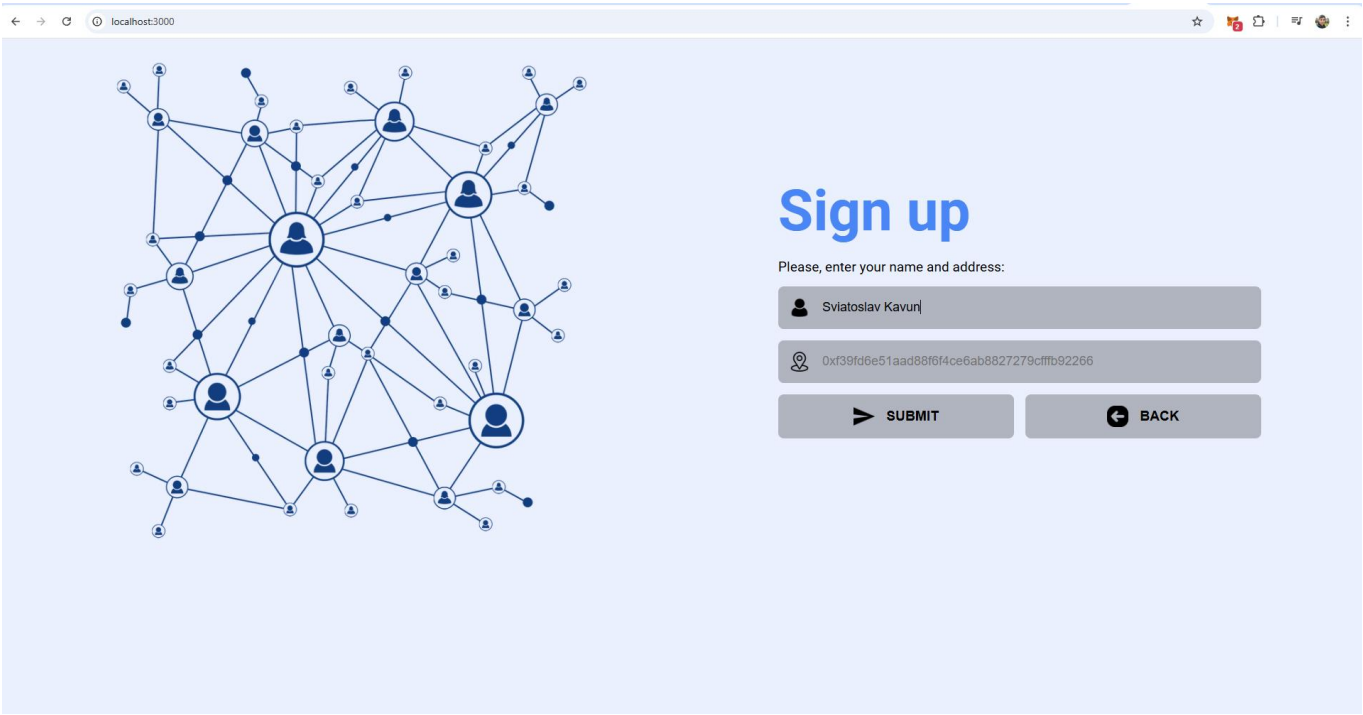

Рисунок 5.4 – Сторінка застосунку для створення акаунту в системі

Після натискання кнопки «Submit», починається взаємодія зі смарт-контрактом.

Застосунок ініціює транзакцію, відправником якої є поточний користувач, а адресою отримувача є адреса смарт контракту. Запит на підтвердження даної транзакції представлений на рисунку 5.5.

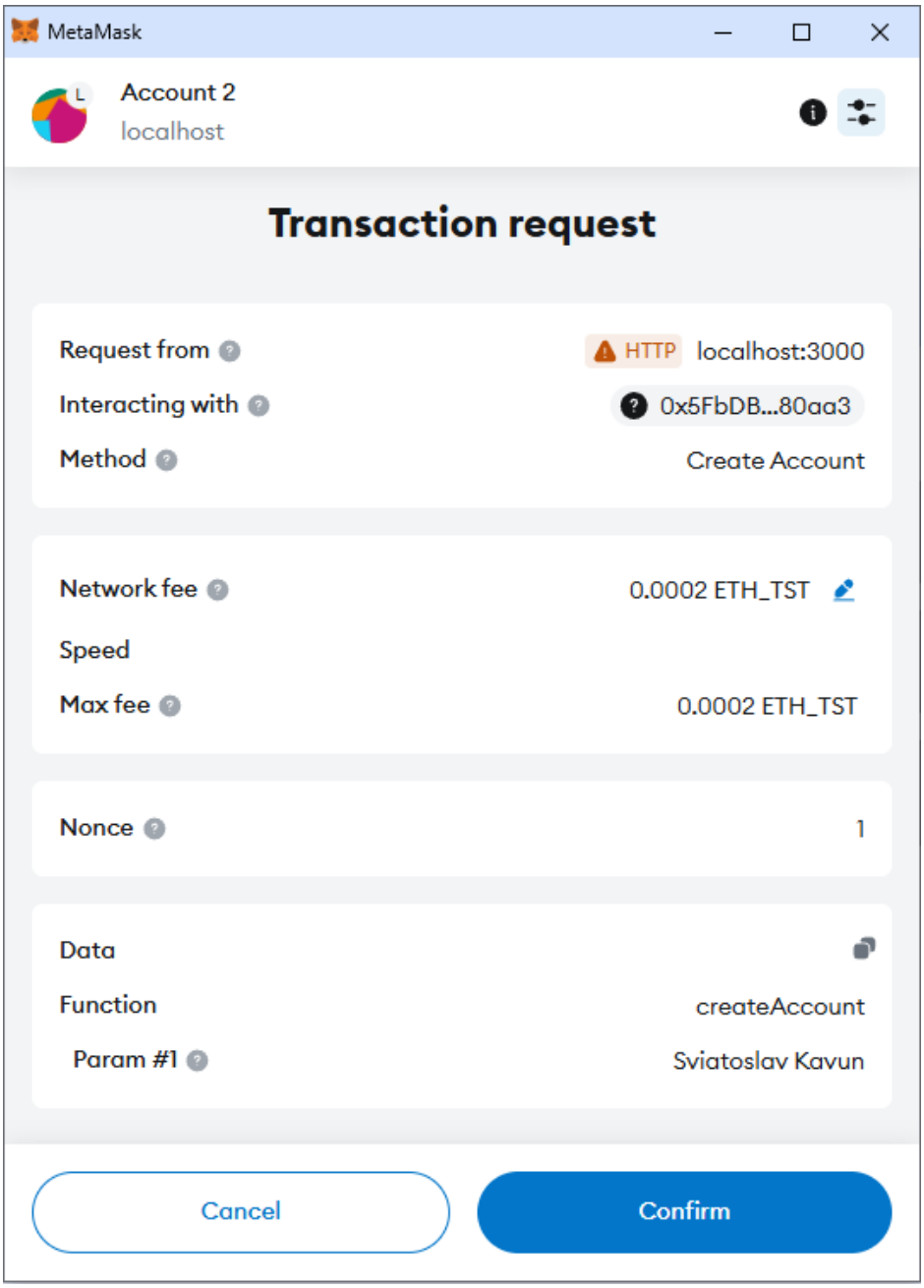

Рисунок 5.5 – Запит на підтвердження транзакції для створення користувача в системі

В даній транзакції варто звернути увагу на останнє поле «Data». Ці дані, що в зашифрованому вигляді передаються смарт контракту, є своєрідною інструкцією для виконання відповідної дії з блокчейном. Дія, що має бути виконана смарт-контрактом, передається в полі Function. В даному випадку викликається функція створення акаунту (createAccount), з передачею імені користувача в якості

параметру. Адреса ж користувача, для ідентифікації, отримується з самої транзакції, адже він є її відправником.

Для реєстрації ще одного користувача в системі, потрібно обрати інший Metamask акаунт, який ще не зареєстрований в системі і провести аналогічну процедуру створення акаунту. Таким чином, можна створити необхідну кількість користувачів для тестування системи. При зміні Metamask акаунту з використанням розширення Metamask відбувається авторизація в додаток під відповідним користувачем з відповідною адресою, що представлено на рисунку 5.6.

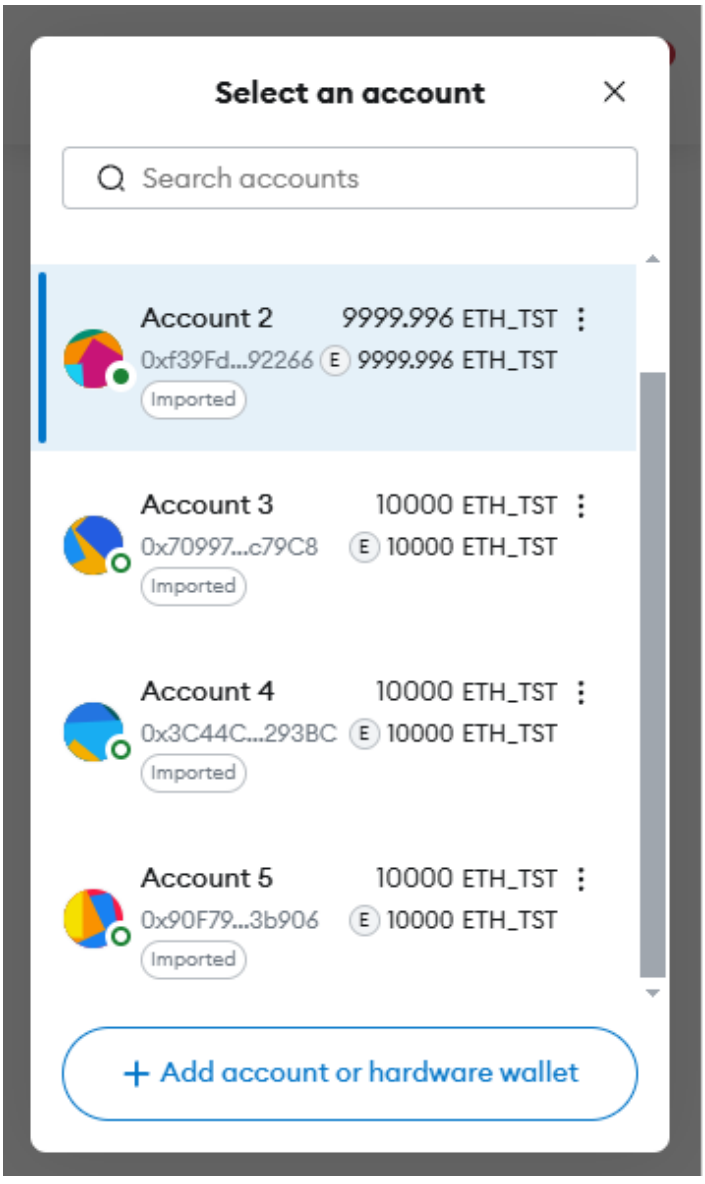

Рисунок 5.6 – Меню вибору акаунту в розширенні Metamask

У випадку, коли даний користувач відсутній в системі, йому буде запропонована процедура створення акаунту в системі.

Для перегляду всіх користувачів системи, доступних для додавання до списку контактів, потрібно обрати пункт меню «Connections», як показано на рисунку 5.7.

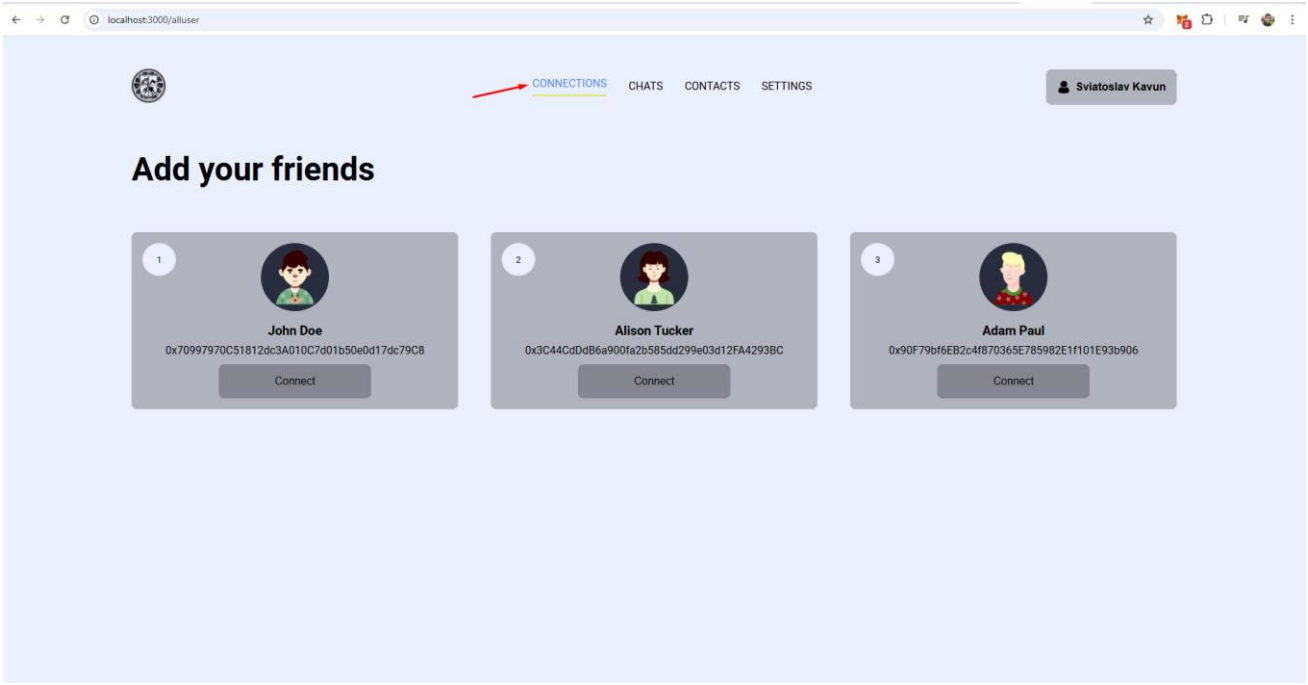

Рисунок 5.7 – Сторінка додавання нових користувачів

Для кожного користувача відображається його ім'я, а також його публічна адреса. Для додавання користувача до контактів потрібно натиснути кнопку «Connect», що ініціює нову транзакцію для поточного користувача (рисунок 5.8). При цьому, викликається відповідна функція смарт-контракту для додавання нового користувача до списку контактів (addFriend). В якості параметрів передається адреса користувача, який додається, та його ім’я.

Варто зазначити, що адреса користувача є його унікальним ідентифікатором в системі. Тобто, людина не може видати себе за когось іншого, адже існування двох користувачів з однаковими адресами є неможливим.

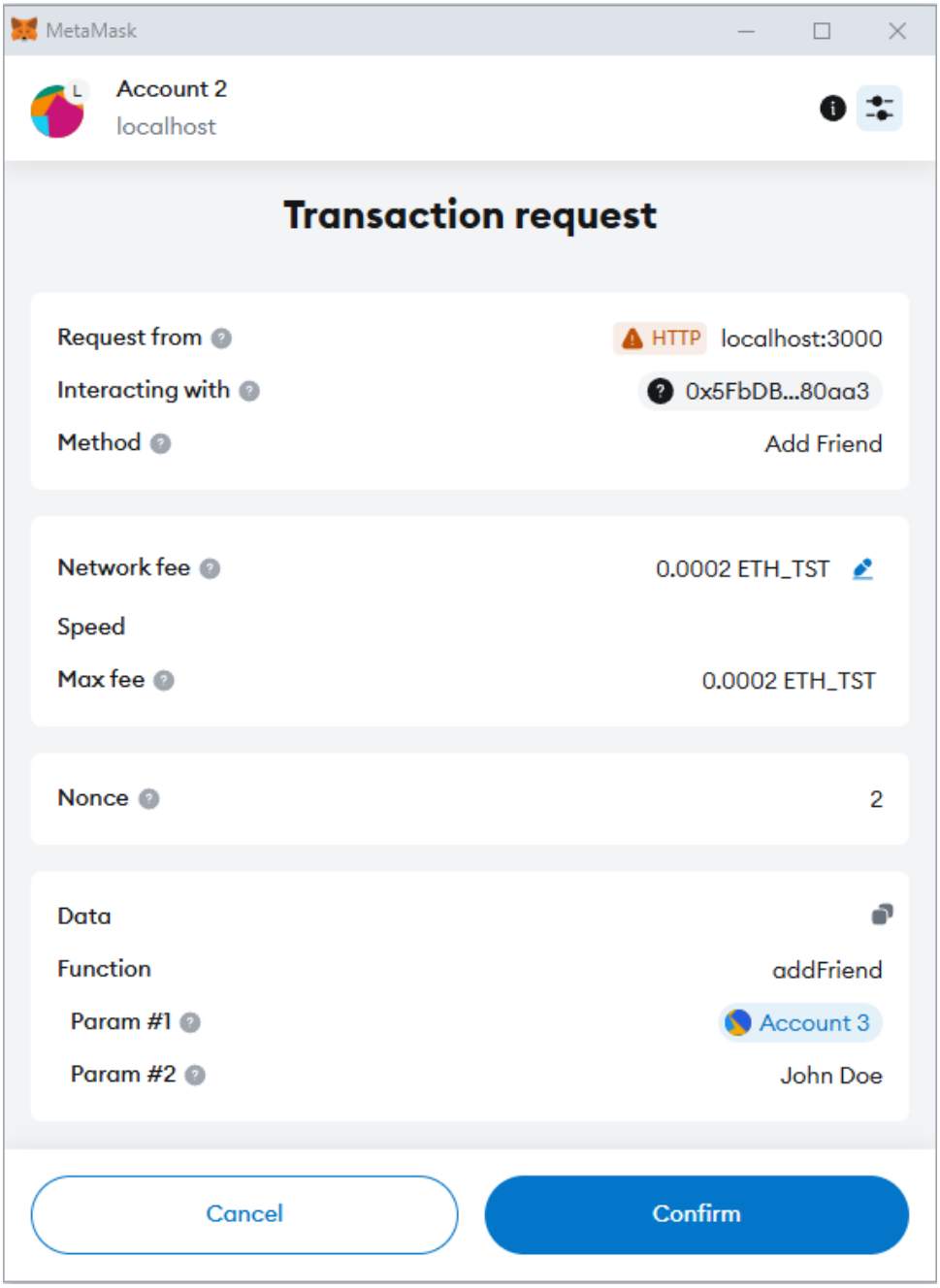

Рисунок 5.8 – Запит на підтвердження транзакції для додавання нового користувача до списку контактів

Після підтвердження транзакції контакт з’являється в чатах поточного користувача. Для перевірки коректної роботи функції потрібно додати всіх наявних користувачів до контактів та перевірити сторінку чатів поточного користувача, яка представлена на рисунку 5.9.

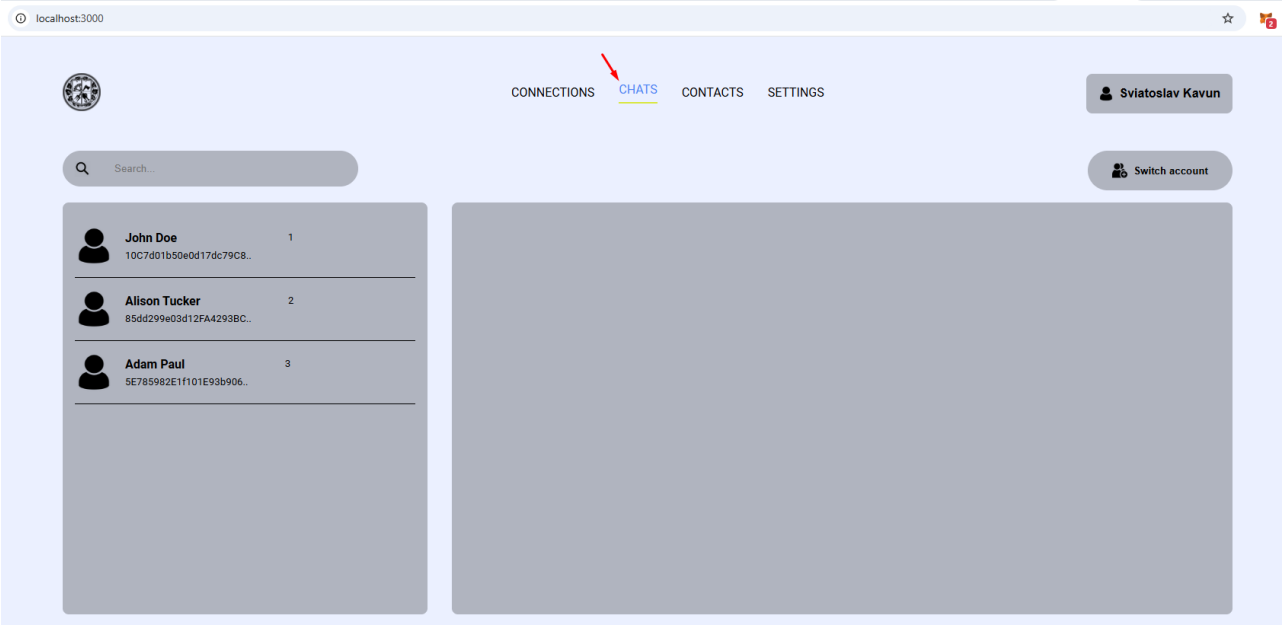

Рисунок 5.9 – Сторінка чатів поточного користувача

Для відправки повідомлення користувачу Adam, потрібно натискнути на чат з його іменем та написати повідомлення у полі для введення. Після натискання кнопки для відправлення повідомлення, система ініціює відповідну транзакцію, що представлено на рисунку 5.10.

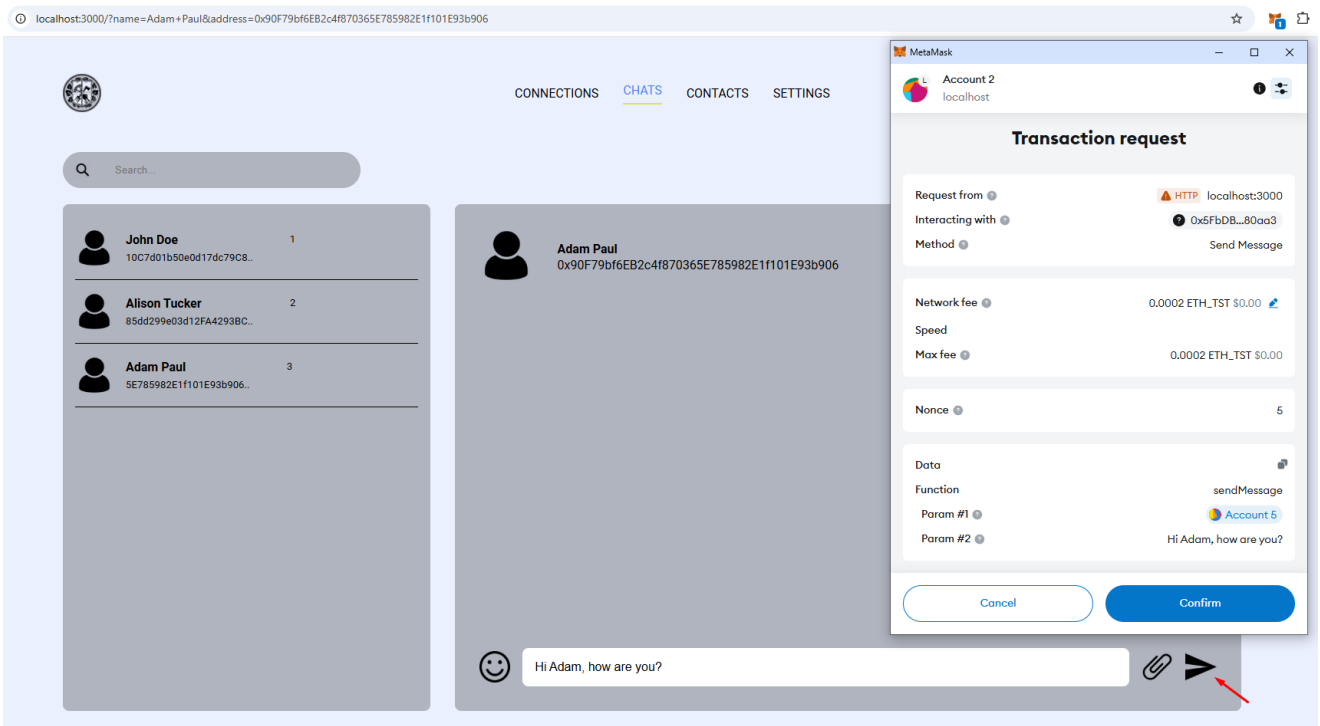

Рисунок 5. 10 – Процес відправки повідомлення в системі

За допомогою даної транзакції викликається функція смарт-контракту для відправки повідомлення користувачеві (sendMessage).

Параметрами, що передаються в рамках транзакції, є унікальна адреса отримувача та саме повідомлення. Таким чином, перехопити або прочитати повідомлення сторонній особі неможливо – його зможе прочитати тільки користувач з вказаною адресою.

Для того, щоб прочитати відправлене повідомлення, потрібно авторизуватись в акаунт користувача з іменем Adam і перейти в чати, що представлено на рисунку 5.11.

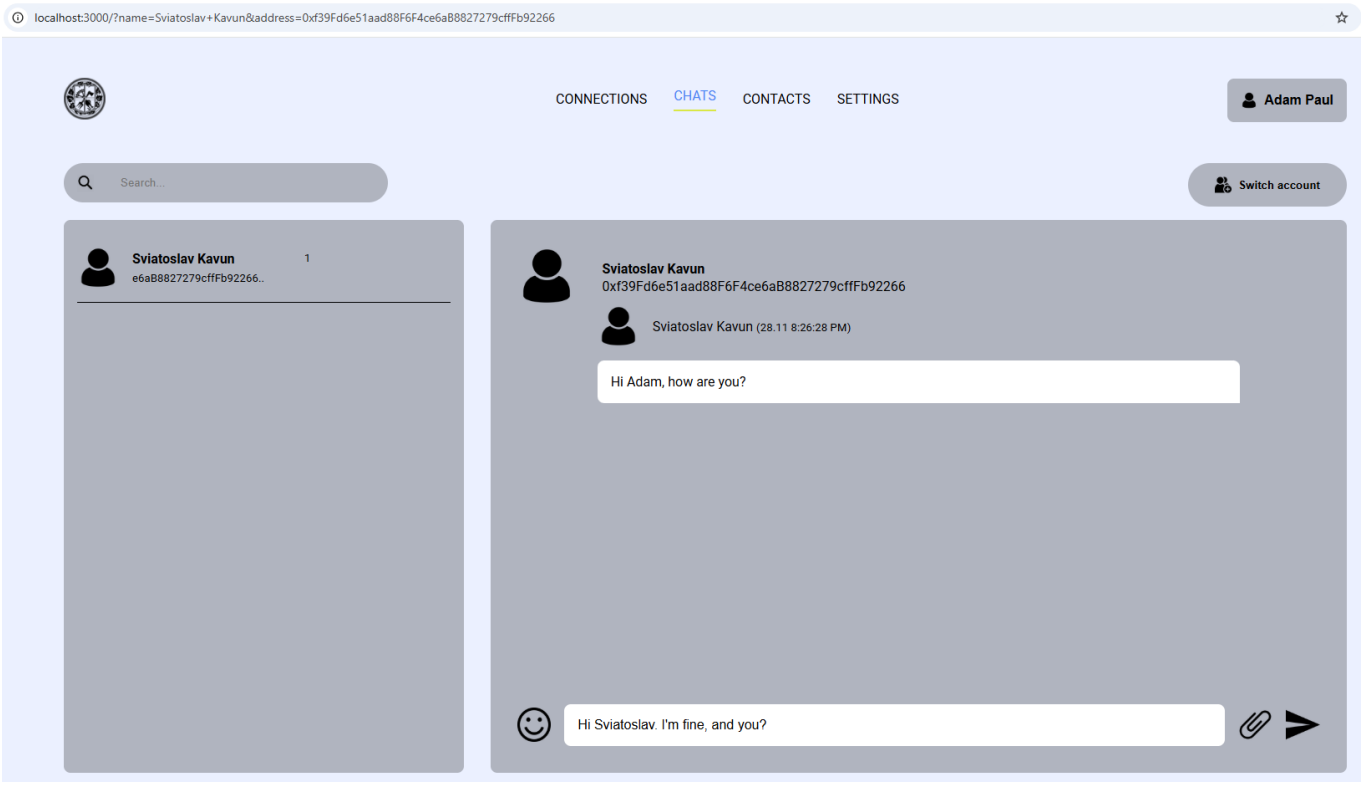

Рисунок 5.11 – Чат між поточним користувачем Adam та користувачем Sviatoslav

Після цього, потрібно знову авторизуватись в акаунт користувача під іменем Sviatoslav і перевірити повідомлення, надіслене користувачем Adam у відповідь (рисунок 5.12).

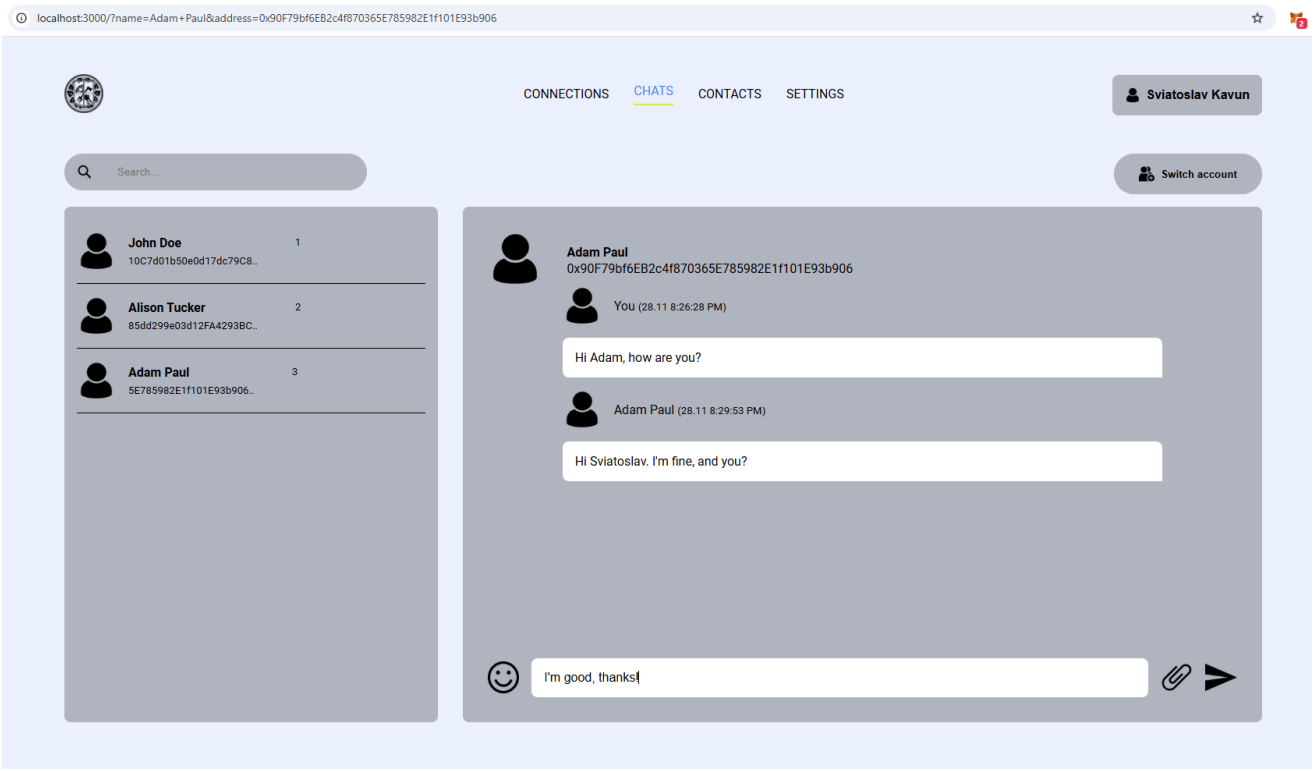

Рисунок 5.12 – Чат між поточним користувачем Sviatoslav та користувачем Adam

5.2 Аналіз можливих атак та їхнього впливу

Розглянемо найнебезпечніші атаки на розроблену систему захищеного зв'язку на основі технології блокчейн.

Одним з небезпечних типів атак є атака не на систему, а на клієнтів, адже жодні заходи для підсилення безпеки системи, шифрування і аудит коду від цього не захищають. Найважливішим завданням є безпечне управління приватними ключами, адже вони слугують основою для аутентифікації та контролю доступу в даній системі.

Вразливості у зберіганні та управлінні приватними ключами на стороні клієнта можуть призвести до серйозних загроз безпеці. Якщо приватні ключі зберігаються у незахищених середовищах, таких як локальне сховище, незашифровані файли чи недостатньо захищені розширення браузера, зловмисники можуть скористатися цими слабкими місцями для їх викрадення. Також, хакери можуть використовувати шкідливе програмне забезпечення, кейлогери чи

експлойти для доступу до пристрою користувача. Наслідками такого типу атак є несанкціоновані транзакції, що в свою чергу веде до компрометації ідентичності користувача в системі та підриву довіри до екосистеми децентралізованого застосунку для захищеної комунікації [29].

Крім того, зловмисники можуть зосереджувати свої зусилля на фішингових атаках, націлених на серверний гаманець Metamask, або на маніпуляції користувачами з метою отримання їхніх приватних ключів чи фраз відновлення. Це, в кінцевому рахунку, призводить до компрометації акаунтів та гаманців користувачів системи.

Інтерфейсні ризики становлять ще одну суттєву загрозу для децентралізованих додатків. Серед поширених атак можна виділити міжсайтовий скриптинг (XSS) і клікджекінг, які створюють суттєві загрози для користувачів. XSS-атаки можуть призводити до викрадення конфіденційної інформації користувачів, включно з даними гаманця, історією транзакцій та особистими даними, які згодом можуть бути використані для подальших атак або крадіжки особистості. Також, зловмисники можуть використовувати XSS для зміни поведінки або інтерфейсу системи, вводячи користувачів в оману щодо виконання небажаних дій, таких як підписання шкідливих смарт-контрактів. Атаки клікджекінгу можуть змушувати користувачів несвідомо авторизувати транзакції, що призводить до несанкціонованих дій, таких як відправка небажаних повідомлень [30].

Найсерйознішим же типом атаки може бути атака на смарт-контракт. Подібно до будь-якого коду, смарт-контракт може мати недоліки безпеки, які можуть поставити під загрозу дані користувача. Зловмисники можуть використовувати вразливості в смарт-контракті , щоб отримати контроль над його виконанням, видалити дані або змінити запрограмовані умови та правила його роботи [31].

5.3 Тестування та аналіз ефективності захисту даних

В розробленій системі для захищеної комунікації шифрування запитів і відповідей API є основою для забезпечення конфіденційності повідомлень і даних користувачів. Оскільки, додаток орієнтований на захист особистої інформації та передачу чутливих даних, шифрування між користувачами та сервером дозволяє гарантувати, що ніхто, крім учасників комунікації, не зможе отримати доступ до вмісту повідомлень.

Використані механізми шифрування на рівні блокчейну, зокрема застосування криптографічних підписів для верифікації запитів і підтвердження автентичності кожної комунікації. Кожен запит та відповідь зашифрована за допомогою асиметричної криптографії, де публічний ключ використовується для шифрування повідомлень, а приватний – для їх дешифрування. Це дозволяє досягти високого рівня безпеки в передачі даних і запобігає їхньому перехопленню або маніпулюванню під час комунікації.

Для захисту системи від вразливостей на клієнтській стороні, дотриманий принцип «не довіряти інформації з боку клієнта». Він полягає в тому, щоб припускати, що будь-які дані, отримані від клієнта, можуть бути скомпрометовані або змінені зловмисниками. Це гарантує, що цілісність і безпека роботи системи не залежать від того, наскільки надійними є дані, введені користувачем. Це захищає не тільки від таких технік, як міжсайтовий скриптинг (XSS) та ін'єкції JavaScript, але й від маніпулювання транзакціями та смарт-контрактами.

Враховуючи, що однією з найбільш серйозних проблем безпеки для розробленої системи є вразливість смарт-контракту, його створенню була приділена особлива увага. Для пом’якшення вразливостей смарт-контракту були дотримані методи безпечного програмування, такі як перевірка введених даних і належна обробка помилок.

Після створення смарт-контракту наступним кроком було написання автоматичних тестів для аналізу ефективності захисту даних і тестування

граничних випадків. В контексті захисту даних, тестування має на меті виявлення вразливостей, таких як неправильне управління доступом, витоки даних або небезпечні операції з даними.

Для тестування контракту використана вже згадана Hardhat Network – локальна мережа Ethereum. Вона вбудована в Hardhat і використовується як мережа за замовчуванням.

В тестах використовується бібліотека ethers.js для взаємодії з створеним контрактом Ethereum, а також фреймворк для тестування Mocha. Ці тести призначені для перевірки основної функціональності розробленого контракту, гарантуючи правильну роботу такий функцій як створення та управління акаунтами, управління контактами, надсилання та отримання повідомлень, а також коректну обробку помилок і крайніх випадків. Нижче наведені короткі пояснення до кожного з тестів:

– Should not allow creating an account if the user already exists. Цей тест підтверджує, що користувач не може створити новий акаунт, якщо він уже зареєстрований. Тест перевіряє, чи правильно контракт скасовує транзакцію з помилкою "User already exists", коли користувач намагається створити акаунт з уже зареєстрованою адресою.

– Should return the correct friends list. Цей тест перевіряє, чи функція для отримання списку контактів (getMyFriendsList) правильно повертає список контактів поточного користувача. Спочатку створюються кілька користувачів, деякі з них додаються в друзі для першого створеного користувача. Після цього перевіряється, чи список містить правильних друзів з очікуваними іменами.

– Should return an empty list if the user has no friends. Цей тест перевіряє, що якщо користувач не має друзів, функція для отримання списку контактів (getMyFriendsList) повертає порожній список. Це підтверджує, що користувачі без друзів обробляються коректно в контракті.

– Should allow users to send messages to friends. Цей тест перевіряє, що користувачі можуть надсилати повідомлення своїм друзям. Після створення

акаунтів і додавання друзів, перший користувач надсилає повідомлення дургому. Тест перевіряє, чи було надіслано повідомлення та чи коректно воно отримане другим користувачем.

– Should not allow sending messages to users who are not friends. Цей тест перевіряє, що користувачі не можуть надсилати повідомлення тим, хто не знаходиться в списку їх контактів. Якщо перший юзер намагається надіслати повідомлення другому, не будучи його другом, транзакція скасовується з повідомленням "You are not friends with the given user".

– Should return an empty array if there are no messages between users. Цей тест перевіряє, що якщо між двома користувачами немає жодного повідомлення, функція для прочитання повідомлення (readMessage) повертає порожній масив. Це підтверджує, що користувачі без історії повідомлень обробляються коректно.

– Should allow a user to create an account and retrieve their username. Цей тест перевіряє, що користувач може створити акаунт, і що функція для отримання імені користувача за його адресою (getUsername) працює коректно. Тест підтверджує, що ім'я користувача збігається з тим, що було призначено при створенні акаунту.

– Should revert if trying to get the username of a non-registered user. Цей тест перевіряє, що функція для отримання імені користувача за його адресою (getUsername) скасовує транзакцію з повідомленням "User is not registered", якщо спробувати отримати ім'я користувача, який не зареєстрований.

– Should return the correct username for multiple users. Цей тест перевіряє, що функція для отримання імені користувача за його адресою (getUsername) коректно працює для кількох користувачів. Він перевіряє, що після створення акаунту кожному користувачу правильно повертається його ім'я, що підтверджує коректну обробку функцією кількох користувачів одночасно.

Для запуску тестів необхідно виконати команду «npx hardhat test» в корені проєкту. Результат роботи наведеної команди представлений на рисунку 5.12.

```
Terminal:  Local  Local (2)  Local (3)
PS D:\Study\Диплом\blockchain-chat> npx hardhat test

  ChatApp Contract
    √ Should not allow creating an account if the user already exists (62ms)
    √ Should return the correct friends list (124ms)
    √ Should return an empty list if the user has no friends
    √ Should allow users to send messages to friends (66ms)
    √ Should not allow sending messages to users who are not friends
    √ Should return an empty array if there are no messages between users
    √ Should allow a user to create an account and retrieve their username
    √ Should revert if trying to get the username of a non-registered user
    √ Should return the correct username for multiple users


  9 passing (2s)
```

Рисунок 5.12 – Результат прогону автоматизованих тестів для розробленого смарт-контракту

Важливою функцією є «beforeEach» (рисунок 5.13), що слугує для підготовки тестових акаунтів і розгортання смарт-контракту перед кожним з тестів. Це спеціальна функція (хук) в бібліотеці для тестування Mocha, яку можна використовувати для встановлення попередніх умов для кожного тесту [32].

```
beforeEach( fn: async function () {
  // Get signers (accounts for testing)
  [owner, firstUser, secondUser, thirdUser] = await ethers.getSigners();

  // Deploy the contract
  const ChatApp = await ethers.getContractFactory("ChatApp");
  chatApp = await ChatApp.deploy();
  await chatApp.deployed();
});
```

Рисунок 5.13 – Функція для підготовки тестових акаунтів і розгортання смарт-контракту перед кожним з тестів

Функція на прикладі тесту для перевірки відправлення та читання повідомлення представлена на рисунку 5.14.

```
it( title: "Should allow users to send messages to friends", fn: async function () {
  // Create accounts and add friends
  await chatApp.connect(firstUser).createAccount( {name}: "User1");
  await chatApp.connect(secondUser).createAccount( {name}: "User2");
  await chatApp.connect(firstUser).addFriend(secondUser.address, "User2");

  // Send a message from firstUser to secondUser
  await chatApp.connect(firstUser).sendMessage(secondUser.address, "Hello, User2!");

  // Read messages from firstUser to secondUser
  const messages = await chatApp.connect(secondUser).readMessage(firstUser.address);

  // Assert that the message was sent
  expect(messages.length).to.equal( value: 1);
  expect(messages[0].msg).to.equal( value: "Hello, User2!");
});
```

Рисунок 5.14 – Тест для перевірки коректності роботи функцій для відправки та прочитання повідомлень

Висновки до розділу 5

В п’ятому розділі продемонстрований та протестований функціонал розробленої системи. Описаний процес підтвердження транзакцій, які є основою для передачі даних в системі.

Також, проаналізовані можливі атаки та їх вплив, після чого описані засоби та інструменти забезпечення захисту від наведених атак в розробленій системі.

Враховуючи те, що найсерйознішою вразливістю системи є вразливість смарт-контракту, розроблений смарт-контракт протестований за допомогою автоматизованих тестів. Ці тести створені в рамках даної роботи спеціально для смарт-контракту, який використовується в системі. В результаті тестування, такі вразливості, як неправильне управління доступом, витоки даних або небезпечні операції з даними виявлені не були, а в граничних випадках смарт-контракт поводиться коректно.

6 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

6.1 Опис ідеї стартап-проєкту

Ідея стартап-проєкту полягає у створенні децентралізованої системи захищеного зв’язку, яка забезпечує конфіденційність, прозорість і автентичність даних за допомогою блокчейн-технології та смарт-контрактів.

Система дозволить користувачам обмінюватися інформацією без участі посередників, гарантуючи захист від втручання чи несанкціонованого доступу.

Зміст ідеї, що пропонується, напрямки застосування проєкту та переваги для користувачів наведені у таблиці 6.1.

Таблиця 6.1 – Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Створення децентралізованої системи захищеного зв’язку, яка забезпечує конфіденційність, прозорість і автентичність даних за допомогою блокчейн-технології та смарт-контрактів.	Корпоративне спілкування, обмін конфіденційними повідомленнями між співробітниками та партнерами.	Гарантована безпека корпоративної інформації без ризику витоку через злам серверів чи внутрішню недбалість.
	Передача медичних даних між лікарями, пацієнтами та страховими компаніями.	Захищена передача особистої медичної інформації, яка відповідає вимогам конфіденційності (наприклад, GDPR – General Data Protection Regulation)

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	Захищений обмін юридично важливими документами та інформацією між адвокатами, клієнтами й судовими установами.	Високий рівень довіри до офіційної комунікації завдяки прозорості й автентифікації в блокчейні.
	Безпечний обмін фінансовими даними між банками, їх клієнтами та фінансовими організаціями.	Мінімізація ризиків шахрайства, пов'язаного з передачею даних, завдяки високому рівню шифрування.
	Платформа для приватного спілкування між користувачами.	Захист персональних даних і свобода від цензури або втручання з боку сторонніх організацій.
	Захищена комунікація між урядовцями, дипломатами та міжнародними організаціями.	Підвищення довіри до обміну інформацією та забезпечення її конфіденційності на державному рівні.
	Координація між об'єктами енергетики, транспорту чи охорони здоров'я.	Захист даних від кібератак та збереження стабільності роботи систем.

Перелік техніко-економічних властивостей та характеристик ідеї, проєкти-конкуренти та порівняльний аналіз показників представлений у таблиці 6.2. Для порівняльного аналізу показників використовувалась шкала «W, N, S», де W (слабкі) – гірші значення, N (нейтральні) – аналогічні значення, а S (сильні) – кращі значення.

Таблиця 6.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проєкту

№	Техніко-економічні характеристики ідеї	(потенційні) товари/концепції конкурентів				Оцінка (W, N, S)
		Моя система	Signal	Wickr Me	WhatsApp	
1.	Децентралізація	Так	Ні	Ні	Ні	S
2.	Можливість використання смарт-контрактів	Так	Ні	Ні	Ні	S
3.	Прозорість обміну даними	Так	Частково	Ні	Ні	S
4.	Шифрування повідомлень	Так	Так	Так	Так	N
5.	Відсутність центрального сервера	Так	Ні	Ні	Ні	S
6.	Можливість аудиту історії повідомлень	Так	Ні	Ні	Ні	S
7.	Зручність користування	Середня	Середня	Середня	Висока	W

8.	Вартість використання	Низька	Безкоштов-но	Платно для бізнесу	Безкоштов-но	N
9.	Стійкість до кібератак	Висока	Висока	Висока	Середня	S

Wickr Me, як і Signal, пропонує високий рівень шифрування та стійкість до атак, але також залежить від центральних серверів, що програє децентралізованій системі.

Для бізнесу Wickr Me може бути платним, що є слабкою стороною порівняно з нашим рішенням із низькою вартістю використання.

У порівнянні з WhatsApp, Wickr Me має подібні технічні характеристики, але менше підходить для масового використання через складність і вузький функціонал.

Таблиця 6.2 дозволяє чітко зрозуміти, де система має конкурентні переваги, а де є точки для покращення. Основні переваги проєкту – децентралізація, прозорість, використання смарт-контрактів та висока стійкість до атак. Шифрування повідомлень та вартість використання відповідають стандартам ринку, а зручність використання поки програє основним конкурентам через складність нової технології для користувачів.

6.2 Технологічний аудит ідеї стартап-проєкту

Для визначення технологічної здійсненності ідеї проєкту проаналізовані такі складові, як технології реалізації, наявність та доступність даних технологій. Результати даного аналізу представлені у таблиці 6.3.

Таблиця 6.3 – Технологічна здійсненність ідеї проєкту

№ п/п	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Користувацький інтерфейс додатку для захищеної комунікації	React або Vue.js	Наявні	Доступні
2.	Блокчейн-мережа для зберігання даних	Ethereum, Polygon	Наявні	Доступні
3.	Інтеграція з блокчейном	Web3.js або Web3.py	Наявні	Доступні
4.	Використання смарт-контрактів	Solidity для написання смарт-контрактів	Наявна	Доступна
5.	Шифрування повідомлень	AES-256, RSA, ECC	Наявні	Доступні
6.	Масштабування транзакцій	Optimism, Arbitrum	Наявні	Доступні
7.	Аудит смарт-контрактів	OpenZeppelin, MythX	Наявні	Доступні

Для реалізації проєкту обрані наявні технології, такі як Ethereum для блокчейн-мережі, Solidity для смарт-контрактів та React для фронтенду. Наведені технології є доступними авторам проєкту, що забезпечує можливість технологічної реалізації. Аналіз показує, що проєкт технологічно здійсненний без необхідності розробки нових інструментів.

6.3 Аналіз ринкових можливостей запуску стартап-проєкту

Для початку аналізу ринкових можливостей проєкту, потрібно провести аналіз попиту, визначивши наявність попиту, обсяг і динаміку розвитку ринку. Результати даного аналізу наведені у таблиці 6.4.

Таблиця 6.4 – Попередня характеристика потенційного ринку стартап-проєкту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	10–15 основних гравців (WhatsApp, Telegram, Signal, Wickr Me, Slack тощо)
2.	Загальний обсяг продаж, грн/ум.од	Перевищує 50 млрд грн (оцінка доходів основних додатків)
3.	Динаміка ринку (якісна оцінка)	Зростає через підвищений попит на захищені комунікації та приватність
4.	Наявність обмежень для входу	Високий рівень конкуренції, необхідність довіри користувачів, масштабування
5.	Специфічні вимоги до стандартизації та сертифікації	Відповідність GDPR, вимоги до шифрування (AES-256, RSA тощо)
6.	Середня норма рентабельності в галузі (або по ринку), %	25–40% (для SaaS-моделей)

Ринок захищених комунікацій є привабливим для входження за попереднім оцінюванням, зважаючи на наступні фактори.

Зростаюча динаміка ринку. Попит на приватність і безпечний обмін даними продовжує зростати через численні кіберзагрози та посилення регуляцій щодо захисту персональних даних.

Обмеження для входу. Хоча конкуренція висока, впровадження інноваційного рішення на основі блокчейну з унікальними характеристиками (децентралізація, відсутність центрального контролю) може стати конкурентною перевагою та дозволити зайняти перспективну нішу.

Норма рентабельності. Високий потенціал прибутковості галузі, особливо при використанні моделі SaaS (передплата) або інтеграції преміум-функцій, робить ринок фінансово вигідним.

Вимоги до стандартизації. Існують чіткі регуляції (GDPR, CCPA), але вони не є недосяжними для технологій, які використовуються у даному проєкті.

Отже, ринок є перспективним за умови правильної стратегії позиціонування, орієнтації на потреби клієнтів і дотримання стандартів безпеки.

Характеристика потенційних груп клієнтів представлена в таблиці 6.5.

Таблиця 6.5 – Характеристика потенційних клієнтів стартап-проєкту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Безпечний обмін конфіденційними повідомленнями	Бізнес-клієнти (компанії, що працюють з конфіденційними даними)	Потребують високого рівня безпеки, відповідності стандартам (GDPR, ISO 27001),	- Надійність шифрування - Відповідність законодавству - Інтеграція з

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
			інтуїтивного інтерфейсу	іншими системами
2.	Приватність у спілкуванні для особистого використання	Індивідуальні користувачі	Цінують простоту використання, відсутність реклами, анонімність	- Зручність інтерфейсу - Високий рівень приватності - Відсутність збору даних
3.	Захищене спілкування у командній роботі	Стартапи, малі та середні бізнеси	Шукають доступні рішення для команди, важлива можливість масштабування, зручність управління доступом	- Доступна ціна - Інструменти адміністрування - Можливість масштабування
4.	Захист даних від кібератак	Журналісти, правозахисники, активісти	Необхідність анонімності, захисту джерел інформації, роботи в умовах блокувань	- Стійкість до атак - Підтримка безпечних транзакцій
5.	Контроль доступу до	Організації державного сектору	Необхідний високий рівень захищеності даних, відповідність	- Високий рівень шифрування - Логування дій

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
	конфіденцій-них документів		державним стандартам	користувачів - Інтеграція з наявними системами документообігу

Кожна група клієнтів має свої особливості поведінки та вимоги до продукту. Це дозволяє краще адаптувати функціонал системи та виділити особливі переваги для кожного сегмента.

Фактори, що перешкоджають ринковому впровадженню проєкту представлені в таблиці 6.6.

Таблиця 6.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Висока конкуренція	Наявність популярних аналогів, таких як Wickr Me, Signal, що вже мають довіру користувачів	Розробити додатковий функціонал, якого немає у конкурентів
2.	Низький рівень довіри до нових рішень	Користувачі можуть скептично ставитися до безпеки або анонімності нового продукту	Активне рекламування, демонстрація прозорості коду та незалежний аудит безпеки

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
3.	Високі витрати на технології	Реалізація блокчейн-рішень може вимагати значних фінансових ресурсів	Залучення інвесторів або партнерів для зниження початкових витрат
4.	Технологічні атаки	Потенційний ризик хакерських атак, спрямованих на вразливості системи	Проведення регулярних тестувань безпеки, впровадження багаторівневої архітектури захисту

Основними загрозами для впровадження проєкту є висока конкуренція, недовіра до нових рішень і технологічні ризики. Для їх подолання компанії слід зосередитися на перевагах продукту, демонстрації безпеки та оптимізації витрат.

Фактори, що сприяють ринковому впровадженню проєкту представлені в таблиці 6.7.

Таблиця 6.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Зростання попиту на безпекові рішення	Збільшення кількості кіберзагроз стимулює попит на захищені канали комунікації	Офіційна презентація переваг продукту, орієнтація на клієнтів з високим рівнем ризику
2.	Інноваційність технології	Використання блокчейну забезпечує унікальність і прозорість продукту	Просування блокчейн-технології як ключової переваги

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
3.	Відсутність повної децентралізації у конкурентів	Конкуренти не реалізують децентралізований підхід для збереження даних	Позиціонування як найкраще децентралізоване рішення
4.	Зростання ринку Web3	Популяризація децентралізованих рішень і додатків сприяє підвищенню інтересу до блокчейн-продуктів	Інтеграція продукту з іншими Web3-екосистемами
5.	Державні ініціативи	Підтримка кібербезпеки на державному рівні, програми фінансування	Участь у державних тендерах, партнерство з урядовими установами

Як видно з таблиці, ринок пропонує значні можливості, зокрема зростання попиту на безпеку, інноваційність технології, відсутність децентралізованих аналогів і підтримку державних ініціатив. Використання цих можливостей через правильне позиціонування продукту та партнерство може забезпечити успішний вихід на ринок.

Для аналізу пропозиції визначимо загальні риси конкуренції на ринку. Результат даного аналізу представлений у таблиці 6.8.

Таблиця 6.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
1. Тип конкуренції – олігополія	Ринок включає обмежену кількість	Впровадження інновацій (блокчейн, смарт-контракти),

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
	великих гравців, але є місце для інноваційних стартапів.	забезпечення унікальності продукту, налагодження маркетингу для захоплення ніші.
2. Рівень конкурентної боротьби – міжнародний	Ринок децентралізованих комунікацій є глобальним і вимагає відповідності міжнародним стандартам.	Вихід на глобальні платформи, локалізація продукту під різні країни, дотримання законодавства та стандартів безпеки в різних регіонах.
3. Галузева ознака – міжгалузева	Система може бути корисною для різних секторів: фінансового, державного управління, приватного бізнесу, медицини тощо.	Позиціонування як універсального рішення, створення додаткових модулів під потреби різних галузей.
4. Конкуренція за видами товарів – товарно-видова	Конкуренція з іншими платформами захищеного зв’язку.	Підкреслення переваг децентралізації, акцент на захисті даних і смарт-контрактах у маркетингових кампаніях.

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства
5. За характером конкурентних переваг - нецінова	Основні переваги – інноваційність, безпека, прозорість, незалежність від центрального сервера.	Інвестування в дослідження та розвиток, створення простого і зручного інтерфейсу, активна робота з клієнтами для роз'яснення переваг.
6. За інтенсивністю – марочна	Ринок швидко зростає, багато стартапів та технологічних гігантів беруть участь у створенні подібних продуктів.	Побудова партнерських відносин, фокус на розвиток екосистеми навколо продукту.

Цей аналіз допомагає краще зрозуміти конкурентне середовище для відповідного позиціонування продукт на ринку.

Для більш детального аналізу умов конкуренціх в галузі використаємо модель 5 сил М. Портера. Результати наведені у таблиці 6.9.

Таблиця 6.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Signal, WhatsApp, Wickr	Нові стартапи в галузі блокчейн-	Постачальники хмарних сервісів,	Компанії та організації, що потребують	Звичайні централізовані комуні-

		комунікацій; великі IT-компанії (Meta, Google тощо).	блокчейн-інфраструктури (Ethereum, Polygon).	високого рівня захисту даних (фінансові, державні, приватні).	каційні системи, використання VPN для захищеного зв'язку.
Висновки	Інтенсивність конкуренції висока через розвиток технологій.	Низький бар’єр входу через відкриті блокчейн-технології, але потрібні інвестиції у безпеку та маркетинг.	Постачальники диктують ціни на інфраструктуру, але є варіанти вибору блокчейн-платформ.	Клієнти диктують вимоги до безпеки, швидкості обміну даними, простоти використання.	Ринок зростає, але вони мають менший рівень безпеки порівнюючи з децентралізованими системами.

Ринок захищених комунікацій характеризується високою інтенсивністю конкуренції через наявність сильних гравців та низький бар’єр входу для нових стартапів. Основна загроза – швидкий вихід нових конкурентів та розвиток товарів-замінників. Постачальники блокчейн-інфраструктури частково диктують умови ринку, але вибір платформ дозволяє оптимізувати витрати. Клієнти вимагають високої безпеки, простоти використання та адаптації під їхні потреби. Для успіху

необхідно зосередитися на інноваціях, ефективному маркетингу та побудові екосистеми навколо продукту.

Результати визначення та обґрунтування переліку факторів конкурентоспроможності наведені у таблиці 6.10.

Таблиця 6.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
1.	Рівень захищеності даних	Сучасні користувачі та компанії вимагають максимального захисту даних через зростання кіберзагроз. Децентралізована структура та смарт-контракти забезпечують високий рівень безпеки.
2.	Простота використання	Більшість користувачів обирають зручні та інтуїтивно зрозумілі додатки. Простий інтерфейс сприяє залученню більшої кількості клієнтів і формуванню лояльності.
3.	Децентралізація	Відсутність центрального сервера підвищує надійність системи, знижує ризики блокування чи втрати даних, що є значущим для клієнтів, які цінують незалежність.
4.	Швидкість передачі даних	Користувачі очікують швидкого обміну повідомленнями без затримок. Технологічні оптимізації в блокчейн-мережі дозволяють забезпечити цей параметр.

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проєктів значущим)
5.	Гнучкість і масштабованість системи	Можливість адаптації під різні потреби клієнтів (наприклад, різні галузі чи географічні регіони) підвищує привабливість продукту на глобальному ринку.
6.	Інтеграція з іншими технологіями	Можливість підключення до інших систем (наприклад, CRM чи ERP) розширює функціонал і збільшує потенційну аудиторію користувачів.
7.	Вартість обслуговування	Конкурентна ціна на інфраструктуру та обслуговування (за рахунок оптимізації смарт-контрактів і вибору блокчейн-платформ) робить продукт доступним для ширшого кола клієнтів.

Основними факторами конкурентоспроможності стартапу є високий рівень захищеності даних, децентралізована архітектура, простота використання, швидкість передачі даних та гнучкість системи. Ці чинники забезпечують важливі переваги на ринку, сприяють залученню клієнтів із різних галузей та формуванню довіри до продукту.

Конкурентна вартість обслуговування та інтеграція з іншими технологіями додатково посилюють позиції стартапу.

За визначеними факторами конкурентоспроможності проведений аналіз сильних та слабких сторін стартап-проєкту, результати якого наведені у таблиці 6.11.

Таблиця 6.11 – Порівняльний аналіз сильних та слабких сторін проєкту

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні конкурентами						
			–3	–2	–1	0	+1	+2	+3
1.	Рівень захищеності даних	19						+	
2.	Простота використання	16			-				
3.	Децентралізація	20							+
4.	Швидкість передачі даних	17				0			
5.	Гнучкість і масштабованість системи	18					+		
6.	Інтеграція з іншими технологіями	17					+		
7.	Вартість обслуговування	18				0			

Заключним етапом ринкового аналізу можливостей впровадження проєкту є SWOT-аналіз, в якому розглядаються сильні та слабкі сторони, загрози та можливості стартапу.

Результати даного аналізу представлені у таблиці 6.12.

Таблиця 6.12 – SWOT- аналіз стартап-проєкту

Сильні сторони:	Слабкі сторони:
– децентралізація забезпечує високий рівень безпеки даних – унікальність продукту завдяки використанню технології блокчейн – гнучкість і масштабованість системи для різних галузей – висока адаптивність до змін ринкових потреб	– залежність від блокчейн-інфраструктури, яка може бути дорогою – обмежена кількість інтеграцій з іншими технологіями – відносно складний інтерфейс для масового споживача

Можливості: – ріст попиту на безпечні комунікаційні платформи через зростання кіберзагроз – вихід на глобальний ринок завдяки універсальності продукту – залучення бізнес-клієнтів і організацій, які цінують конфіденційність	Загрози: – висока інтенсивність конкуренції з боку великих гравців – залежність від змін у технологічних стандартах блокчейн-індустрії – ризик зниження попиту на децентралізовані рішення через складність їх використання

На основі SWOT-аналізу проаналізовані альтернативи ринкової поведінки, результат аналізу яких наведений у таблиці 6.13.

Таблиця 6.13 – Альтернативи ринкового впровадження стартап-проєкту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Запуск рекламної кампанії для підвищення впізнаваності бренду через соціальні мережі, технoблоги та конференції	Висока	3-6 місяців
2.	Оптимізація інтерфейсу користувача на основі зворотного зв’язку від тестувальників для полегшення використання	Висока	6-9 місяців
3.	Партнерство з іншими блокчейн-стартапами для розширення функціоналу та інтеграції з популярними платформами	Середня	9-12 місяців
4.	Створення безкоштовної базової версії додатку для залучення масового	Висока	6-12 місяців

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
	користувача та впровадження преміум-функцій для бізнес-клієнтів		
5.	Сертифікація продукту відповідно до міжнародних стандартів безпеки (наприклад, ISO/IEC 27001) для підвищення довіри клієнтів	Середня	12-18 місяців
6.	Вихід на глобальні ринки шляхом адаптації продукту до місцевих регуляторних вимог і запуску локалізованих версій додатку	Середня	18-24 місяці
7.	Розробка освітніх матеріалів для клієнтів (вебінари, інструкції) з метою демонстрації переваг децентралізації та блокчейн-рішень	висока	3-6 місяців
8.	Запуск пілотного проєкту з корпоративними клієнтами для демонстрації ефективності та отримання перших відгуків	висока	6-9 місяців

В якості основної альтернативи обраний запуск пілотного проєкту з корпоративними клієнтами для демонстрації ефективності та отримання перших відгуків. Запуск рекламної кампанії є найбільш ймовірною для отримання ресурсів та швидкою альтернативою, а розробка освітніх матеріалів також є швидким процесом і дозволяє оперативно залучати потенційних користувачів.

6.4 Розроблення ринкової стратегії проєкту

Для визначення стратегії охоплення ринку потрібно спершу проаналізувати потенційних споживачів продукту. Результати даного аналізу наведені у таблиці 6.14.

Таблиця 6.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1.	Блокчейн-ентузіасти, технічні спеціалісти	Висока	Середній	Висока	Висока складність
2.	Бізнес-клієнти, що потребують високого рівня безпеки та конфіденційності у комунікаціях	Висока	Високий	Середня	Середня складність
3.	Техноблоги, медіа, новітні технологічні стартапи	Середня	Низький	Середня	Низька складність

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
4.	Загальні користувачі, які цінують конфіденційність та інноваційні рішення	Середня	Середній	Низька	Низька складність
Обрані такі цільові групи, як: – Блокчейн-ентузіасти, технічні спеціалісти – Бізнес-клієнти, що потребують високого рівня безпеки та конфіденційності у комунікаціях					

Оскільки стартап має два основні сегменти клієнтів з різними потребами та рівнями готовності до використання технології блокчейн, найкраще підходить диференційований маркетинг. Це дозволяє розробити окремі програми ринкового впливу для кожного з сегментів.

Для блокчейн-ентузіастів і технічних спеціалістів – зосередитись на просуванні інноваційних можливостей продукту, доступності та інтеграції з існуючими блокчейн-мережами.

Для бізнес-клієнтів – акцент на безпеці, конфіденційності і відповідності міжнародним стандартам.

Результат формування базової стратегії розвитку представлений у таблиці 6.15.

Таблиця 6.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Запуск пілотного проєкту з корпоративними клієнтами для демонстрації ефективності стартап-проєкту та отримання перших відгуків	Диферен-ційований маркетинг	– Високий рівень безпеки та конфіденційності – Інноваційність технології блокчейн – Гнучкість та адаптація під потреби конкретних клієнтів	Стратегія диферен-ціації

Опис базової стратегії конкурентної поведінки наведений у таблиці 6.16.

Таблиця 6.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проєкт «першопрохід-цем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1.	Так, проєкт є першим в ряді	Компанія буде шукати нових	Компанія не буде копіювати конкретні	Стратегія лідера

№ п/п	Чи є проєкт «першопрохід-цем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
	рішень для захищеного зв'язку на основі блокчейн	споживачів, зокрема корпоративних клієнтів	характеристики продуктів конкурентів, натомість використовуватиме власні інноваційні технології для створення продукту	

Результат розроблення стратегії позиціонування наведені у таблиці 6.17.

Таблиця 6.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспромож-ні позиції власного стартап-проєкту	Асоціації, які мають сформувати комплексну позицію проєкту
1.	– високий рівень безпеки та конфіденцій-ності даних – інновацій ність технології	стратегія диферен-ціації	– використання блокчейн технологій для забезпечення захищеного зв’язку – інноваційний підхід у створенні продукту	– безпечний зв'язок, що гарантує конфіденцій-ність

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспромож-ні позиції власного стартап-проєкту	Асоціації, які мають сформувати комплексну позицію проєкту
	– простота використання для бізнесу		– гнучкість і адаптивність продукту	– інноваційність у підході до комунікацій – надійність для корпоративних клієнтів

6.5 Розроблення маркетингової програми стартап-проєкту

Для початку розроблення маркетингової програми потрібно сформувати маркетингову концепції товару, ключові переваги якої представлені у таблиці 6.18.

Таблиця 6.18 – Визначення ключових переваг концепції потенційного товару.

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
1.	Високий рівень безпеки та конфіденційності даних	Забезпечення захищеного зв'язку завдяки технології блокчейн, що унеможливлює перехоплення та несанкціонований доступ	Використання блокчейн-технології як основи для безпеки, чого немає в більшості конкурентів
2.	Інноваційність та відповідність	Унікальний підхід до комунікацій через децентралізовану систему з	Інноваційність продукту та переваги перед традиційними

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами
	сучасним трендам	підтримкою смарт-контрактів	централізованими системами
3.	Простота використання для корпоративних клієнтів	Інтуїтивний інтерфейс, налаштований під потреби бізнесу, та легка інтеграція у бізнес-процеси	Гнучкість і адаптивність системи для різних бізнес-сегментів
4.	Надійність у роботі	Постійна доступність системи, навіть у випадку відмови окремих вузлів блокчейн-мережі	Децентралізована структура забезпечує вищу стійкість до збоїв порівняно з централізованими рішеннями конкурентів

Три рівні моделі товару описані у вигляді таблиці 6.19.

Таблиця 6.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Система забезпечує захищену комунікацію на основі технології блокчейн. Основна функціональна вигода – конфіденційність, анонімність і прозорість обміну даними.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Децентралізоване зберігання даних. 2. Шифрування E2E (end-to-end).	Нм Нм	Тл Тх
	Якість: сертифіковане шифрування, регулярні аудити безпеки.		
	Пакування: інтуїтивно зрозумілий користувацький інтерфейс.		

	Марка: «Система захищеної комунікації на основі технології блокчейн»
ІІІ. Товар із підкріпленням	До продажу: безкоштовний пробний період.
	Після продажу: технічна підтримка, оновлення системи для усунення вразливостей.
За рахунок чого потенційний товар буде захищено від копіювання: патент на технологічне рішення	

Результати аналізу для визначення цінових меж наведені у таблиці 6.20.

Таблиця 6.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
	5–10 у.о./міс.	10–25 у.о./міс.	1000–3000 у.о./міс	Від 8 до 20 у.о./міс

Результати визначення оптимальної системи збуту представлені у таблиці 6.21.

Таблиця 6.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Цільова аудиторія віддає перевагу онлайн-	Технічна підтримка,	Прямий канал	Прямий збут через офіційний сайт та

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
	закупівлям, зручності доступу до продукту, швидкій інтеграції та можливості тестового періоду.	навчання клієнтів, регулярні оновлення		хмарні сервіси із можливістю пробного періоду

Результат розроблення концепції маркетингових комунікацій представлений у таблиці 6.22.

Таблиця 6.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комуніка-цій, якими корист-уються цільові клієнти	Основні позиції, обрані для позиціону-вання	Завдання рекламного повідомлення	Концеп-ція рекламно-го звернення
1.	Потреба у високому рівні безпеки, зручності використання та конфіденцій-	Соцмережі (LinkedIn, Twitter), професійні форуми, email-розсилки	Основними є надійність та простота інтеграції у бізнес-процеси.	Підкреслити переваги у захисті даних, адаптивності до потреб клієнта, зручності користування	Довіра до передо-вих техноло-гій безпеки комуніка-ції

№ п/п	Специфіка поведінки цільових клієнтів	Канали комуніка-цій, якими користу-ються цільові клієнти	Основні позиції, обрані для позиціону-вання	Завдання рекламного повідомлення	Концеп-ція рекламно-го звернення
	ності у спілкуванні				

Висновки до розділу 6

Проведений аналіз свідчить про наявність можливості ринкової комерціалізації проєкту.

На ринку існує попит на продукт, пов’язаний із забезпеченням безпечної та конфіденційної комунікації, динаміка ринку демонструє стабільне зростання, а потенційна рентабельність підтверджує економічну доцільність роботи у цьому сегменті.

Проєкт має перспективи впровадження завдяки чіткому визначенню цільових груп клієнтів, мінімізації впливу бар’єрів входження та конкурентних переваг у вигляді децентралізованого зберігання даних і E2E-шифрування.

Аналіз конкурентного середовища показав, що проєкт здатний витримувати конкуренцію завдяки використанню сучасних технологій та вдало вибудованій системі збуту.

Для ринкової реалізації проєкту найбільш доцільним є варіант використання багаторівневої збутової системи з акцентом на канали онлайн-комунікації.

Отже, подальша імплементація проєкту є доцільною, оскільки проєкт відповідає потребам ринку, має конкурентні переваги та перспективи успішної комерціалізації.

ВИСНОВКИ

Дане дослідження показало, що технологія блокчейн має великий потенціал для створення ефективної та безпечної системи для комунікації. Після вивчення фундаментальних принципів роботи блокчейну стало зрозуміло, що децентралізована структура та криптографічний захист забезпечують високу стійкість проти зовнішнього втручання та атак. Смарт-контракти, невід'ємна частина блокчейну, відіграють провідну роль у створенні автоматизованих, безпечних і прозорих процесів обміну даними.

Порівняння технології блокчейн із традиційними підходами до досягнення безпечних комунікацій виявляє явні переваги останнього з точки зору конфіденційності, незмінності та мінімізації ризику неавторизованого доступу. Огляд подібних систем підтвердив, що блокчейн пропонує нові можливості захисту даних, які виходять далеко за рамки традиційних методів. Тому, впровадження блокчейна в комунікаційні системи є важливим кроком для посилення безпеки інформаційних технологій.

В рамках даної роботи розроблена, реалізована та протестована система, що забезпечує високий рівень безпеки передачі даних завдяки децентралізованій архітектурі та сучасним методам криптографічного захисту. У ході дослідження досягнута поставлена ціль, а саме: спроєктована архітектура системи, обрані механізми валідації транзакцій, автентифікації та передачі даних, реалізовані технічні рішення з використанням обраних інструментів, а також проведене тестування функціоналу та аналіз можливих атак.

Результати роботи відповідають завданню на дипломне проєктування та висунутим вимогам. Система демонструє стійкість до загроз, характерних для децентралізованих рішень, а її ефективність підтверджена в ході тестування. Незважаючи на незначне збільшення затримок передачі даних через обчислювальну складність блокчейну, це компенсується підвищеним рівнем безпеки та прозорістю обробки транзакцій.

Практична значущість роботи полягає у створенні моделі, що може бути використана для подальшого розвитку безпечних комунікаційних систем у різних галузях, таких як фінанси, медицина чи державне управління. Результати дослідження мають практичну цінність та можуть бути впроваджені у реальних умовах, особливо в сферах, де критично важлива конфіденційність та цілісність даних.

Дослідження підтвердило перспективність застосування блокчейну для розв'язання задач захищеної комунікації, однак для досягнення ще вищого рівня ефективності потрібно продовжити дослідження в напрямах оптимізації швидкості обробки транзакцій та інтеграції з іншими технологіями.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Що таке блокчейн і як він працює простими словами. URL: https://112.ua/cto-takoe-blokcejn-i-kak-on-rabotaet-prostymi-slovami-41490

2. 8 strategies to ensure communication security. URL: https://www.rocket.chat/blog/communication-security

3. Основи та принципи технології блокчейн. URL: https://www.bitbon.space/ua/knowledge-base/distributed-ledger-technologies-blockchain/technological-aspects-of-blockchain/foundations-and-principles-of-the-blockchain-technology

4. Consensus Mechanisms In Blockchain: A Deep Dive Into The Different Types. URL: https://hacken.io/discover/consensus-mechanisms/

5. What Is a Consensus Mechanism. URL: https://builtin.com/blockchain/consensus-mechanism

6. Secure Peer-to-Peer communication based on Blockchain. URL: https://hal.science/hal-02180329/document

7. Що таке смарт-контракт у блокчейні та як він працює. URL: https://blog.whitebit.com/uk/about-smart-contracts/

8. Що таке смарт-контракти. URL: https://klona.ua/uk/blog/blockchain-smart-contract/shho-take-smart-kontrakty

9. Смарт-контракти і питання безпеки. URL: https://www.itsec.ru/articles/smart-kontrakty-i-voprosy-bezopasnosti

10. Що таке смарт-контракти: теорія зі схемами та прикладами. URL: https://trusteeglobal.com/academy/shho-take-smart-kontrakty-teoriya-zi-shemamy-ta-prykladamy/

11. The Role of Blockchain Technology in Personal Data Protection. URL: https://pdtn.org/blockchain-in-data-protection/

12. Secure communication. URL: https://www.slideshare.net/slideshow/secure-communication-74293583/74293583

13. Secure Communication Definition. URL: https://www.netmaker.io/glossary/secure-communication

14. Cryptography: The Mathematics Behind Secure Communication. URL: https://medium.com/@contact_18616/cryptography-the-mathematics-behind-secure-communication-d3ce62327864

15. Cryptography in Modern Communication Systems. URL: https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ca3000e6b9bea175cc29d737b8cf02d5ef9dae14

16. What is the Signal app and should you use it URL: https://nordvpn.com/blog/what-is-signal-app/

17. Signal. URL: https://signal.org/

18. Wickr Know How: AWS Wickr Overview. URL: https://wickr.com/wickr-know-how-wickr-pro-overview/

19. What are the benefits of using Wickr vs Elements URL: https://www.quora.com/What-are-the-benefits-of-using-Wickr-vs-Elements

20. What is Web3. URL: https://www.kraken.com/learn/what-is-web3

21. Decentralized Applications (dApps): Definition, Uses, Pros and Cons. URL: https://www.investopedia.com/terms/d/decentralized-applications-dapps.asp

22. What Is Ethereum and How Does It Work. URL: https://www.investopedia.com/terms/e/ethereum.asp

23. What Is MetaMask. URL: https://members.delphidigital.io/projects/metamask

24. Ethereum Upgrade Adds to Crypto Mania Sparked by Bitcoin’s Surge. URL: https://www.bloomberg.com/news/articles/2020-11-25/ethereum-upgrade-adds-to-crypto-mania-sparked-by-bitcoin-s-surge

25. Understanding Ethereum Transactions. URL: https://medium.com/coinmonks/understanding-ethereum-transactions-56b06be767e3

26. Hardhat Network. URL: https://hardhat.org/hardhat-network/docs/overview

27. What is a Smart Contract ABI and How to Get it. URL: https://www.cyfrin.io/blog/what-is-a-smart-contract-abi-and-how-to-get-it

28. What is a blockchain genesis block. URL: https://www.coinbase.com/learn/crypto-glossary/what-is-a-blockchain-genesis-block

29. Real-World Examples of dApps, Lessons Learned & Strategies for Protecting Every Layer. URL: https://hacken.io/discover/dapps-examples/

30. Best Practices for Building Secure Decentralized Apps. URL: https://wesoftyou.com/ai/best-practices-for-building-secure-decentralized-apps/

31. Web3 Security: Business Risks, Attack Types, and Best Practices for Protecting Web3. URL: https://evacodes.com/blog/web3-security

32. Mocha beforeEach Explained With a Side-by-Side Comparison. URL: https://www.testim.io/blog/mocha-beforeeach-explained/