БАКАЛАВРСЬКА РОБОТА

БР. ІП - 25.00.00.000 ПЗ

Група ІП-23-1К

Чугайда Павло

2025

Івано-Франківський національний технічний університет нафти і газу

Факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Чугайда Павло Степанович

(прізвище, ім'я, по батькові)

УДК 004

(індекс)

БАКАЛАВРСЬКА РОБОТА

Застосування методів децентралізованих обчислень в соціальному нетворкінгу

(назва роботи)

Інженерія програмного забезпечення

(назва освітньої програми)

121 - Інженерія програмного забезпечення

(шифр і назва спеціальності)

Робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

Здобувач освітнього рівня Чугайда П.С.

(підпис, ініціали та прізвище здобувача)

Науковий керівник Вовк Роман Богданович, к.т.н., доцент

(підпис, прізвище, ім'я, по батькові, науковий ступінь, вчене звання керівника)

Допущено до захисту
Завідувач кафедри

доц. Бандура В.В.

(посада) (підпис) (дата) (ініціали та прізвище)

Івано-Франківськ – 2025

Івано-Франківський національний технічний університет нафти і газу

Інститут, факультет інформаційних технологій

Кафедра інженерії програмного забезпечення

Ступінь вищої освіти бакалавр

Спеціальність 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ:
Зав. кафедрою **ІПЗ**
доц. В.В. Бандура
“ ____ ” _____2025 р.

ЗАВДАННЯ

НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТОВІ

Чугайді Павлу Степановичу

(прізвище, ім’я, по-батькові)

1. Тема проекту (роботи) “ Застосування методів децентралізованих обчислень в соціальному нетворкінгу”

керівник проекту (роботи) Вовк Р.Б., доцент

затверджені наказом закладу вищої освіти від “ 28 ” квітня 2025 р. № 264/7

2. Строк подання студентом проекту (роботи) 10 червня 2025 р.

3. Вихідні дані до проекту (роботи) Результати і матеріали отримані під час проходження переддипломної практики

4. Зміст розрахунково - пояснювальної записки (перелік питань, які потрібно розробити)

1. Аналіз застосування концепцій децентралізованих обчислень в соціальному нетворкінгу
2. Застосування Solid та IPFS у децентралізованому соціальному нетворкінгу
3. Архітектурні підходи від централізації до децентралізації в соціальному нетворкінгу
4. Програмна імплементація методів децентралізованих обчислень в соціальному нетворкінгу
5. Функціональність реалізованого додатку соціального нетворкінгу

5. Перелік графічного матеріалу (з точним зазначенням обов’язкових креслень)

1. Розподілена соціальна мережа Mastodon (рис. 1.1)
2. Архітектура Solid (рис. 1.2)
3. Архітектура Поду (рис. 1.3)
4. Обмін даними у IPFS власником (рис. 1.4)
5. Запит даних отримувачем (рис. 1.5)

6. Консультанти розділів проекту (роботи)

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв

7. Дата видачі завдання **28 квітня 2025 р.**

Керівник _____
(підпис)

Завдання прийняв до виконання _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів проекту	Примітка
1	Аналіз застосування концепцій децентралізованих обчислень в соціальному нетворкінгу	10.05.2025	виконано
2	Застосування Solid та IPFS у децентралізованому соціальному нетворкінгу	19.05.2025	виконано
3	Архітектурні підходи від централізації до децентралізації в соціальному нетворкінгу	23.05.2025	виконано
4	Програмна імплементація методів децентралізованих обчислень в соціальному нетворкінгу	04.06.2025	виконано
5	Функціональність реалізованого додатку соціального нетворкінгу	17.06.2025	виконано
6	Оформлення пояснювальної записки дипломної роботи завідувачем кафедри	10.06.2025	виконано

Студент – дипломник _____
(підпис)

Керівник роботи _____
(підпис)

АНОТАЦІЯ

Бакалаврська робота містить 77 сторінок, 29 рисунків, список використаних джерел із 39 найменуваннями.

Метою даної роботи є дослідження архітектурних підходів, технічних рішень та практичних аспектів імплементації децентралізованих обчислень у соціальному нетворкінгу.

Об'єкт дослідження - системи соціального нетворкінгу та механізми організації обміну даними між користувачами.

Предмет дослідження - технології децентралізованих обчислень, зокрема Solid та IPFS, та їхнє застосування у побудові соціальних мереж.

В першому розділі визначено основні проблеми централізованих соціальних мереж і охарактеризовано переваги Solid та IPFS як ключових технологій децентралізації

В другому розділі розглянуто еволюцію архітектур соціальних мереж, визначено переваги IPFS та блокчейну, а також проаналізовано потенціал Solid як децентралізованої парадигми.

В третьому розділі запропоновано реалізацію інтегрованого рішення на основі Solid та IPFS, проаналізовано його продуктивність, функціональність і обмеження в умовах реального використання.

Висновок: проаналізовано проблематику централізованих соціальних мереж, розглянуто архітектурні особливості децентралізованих рішень, а також описано практичну імплементацію програмного додатку соціального нетворкінгу з використанням Solid та IPFS

КЛЮЧОВІ СЛОВА: ДЕЦЕНТРАЛІЗОВАНІ ОБЧИСЛЕННЯ, СОЦІАЛЬНИЙ НЕТВОРКІНГ, SOLID, IPFS, КОНФІДЕНЦІЙНІСТЬ, РОЗПОДІЛЕНЕ ЗБЕРІГАННЯ, WEB 3.0

ANNOTATION

The bachelor's thesis contains 77 pages, 29 figures, a list of used sources with 39 names.

The purpose of this work is to study architectural approaches, technical solutions and practical aspects of implementing decentralized computing in social networking.

The object of research is social networking systems and mechanisms for organizing data exchange between users.

The subject of research is decentralized computing technologies, in particular Solid and IPFS, and their application in building social networks.

The first section identifies the main problems of centralized social networks and characterizes the advantages of Solid and IPFS as key decentralization technologies.

The second section considers the evolution of social network architectures, identifies the advantages of IPFS and blockchain, and analyzes the potential of Solid as a decentralized paradigm.

The third section proposes the implementation of an integrated solution based on Solid and IPFS, analyzes its performance, functionality and limitations in real-world use.

Conclusion: the issues of centralized social networks are analyzed, the architectural features of decentralized solutions are considered, and the practical implementation of a social networking software application using Solid and IPFS is described.

KEYWORDS: DECENTRALIZED COMPUTING, SOCIAL NETWORKING, SOLID, IPFS, PRIVACY, DISTRIBUTED STORAGE, WEB 3.0

ЗМІСТ

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ACL - Access Control List (Список контролю доступу)

AS - Autonomous System (Автономна Система)

CDN - Content Delivery Network (Мережа доставки контенту)

CID - Content Identifier (Ідентифікатор контенту)

DHT - Distributed Hash Table (Розподілена хеш-таблиця)

IPFS - InterPlanetary File System

IPNS - InterPlanetary Naming System

LDP - Linked Data Platform (Платформа Пов'язаних Даних)

MCM - Мобільні соціальні мережі (Mobile Social Networks)

MSN - Mobile Social Networks (Мобільні соціальні мережі)

NAT - Network Address Translation (Перетворення мережевих адрес)

PeerID - Peer Identifier (Ідентифікатор вузла/піра)

P2P - Peer-to-Peer (Одноранговий)

Pod - Personal Online Data Store (Персональне Онлайн Сховище Даних)

PoS - Proof of Stake (Доказ частки володіння)

PoW - Proof of Work (Доказ виконаної роботи)

QUIC - Quick UDP Internet Connections (Протокол Quick UDP Internet Connections)

RDF - Resource Description Framework (Фреймворк опису ресурсів)

RDFa - Resource Description Framework in Attributes (Фреймворк опису ресурсів в атрибутах)

SHA1 - Secure Hash Algorithm 1 (Алгоритм безпечного хешування 1)

SHA256 - Secure Hash Algorithm 256 (Алгоритм безпечного хешування 256)

Solid - Social Linked Data (Соціальні пов'язані дані)

SPARQL - SPARQL Protocol and RDF Query Language (Протокол SPARQL та мова запитів RDF)

SSS - Shamir's Secret Sharing (Алгоритм Шаміра для розподілу секрету)

TCP - Transmission Control Protocol (Протокол управління передачею)

UDP - User Datagram Protocol (Протокол дейтаграм користувача)

W3C - World Wide Web Consortium (Консорціум Всесвітньої павутини)

WebID - Web Identifier (Веб-ідентифікатор)

ВСТУП

У сучасних умовах стрімкого зростання обсягів персональних даних, які циркулюють у цифровому просторі, питання захисту приватності та інформаційної безпеки користувачів набувають особливої ваги. Традиційні централізовані соціальні мережі, такі як Facebook, Instagram або Twitter, функціонують за принципом концентрації даних на окремих серверах, що створює ризики витоків, маніпуляцій та несанкціонованого використання персональної інформації. Ці проблеми актуалізують потребу в альтернативних моделях соціальної взаємодії, які базуються на принципах децентралізації, прозорості та користувацького контролю.

В останнє десятиліття значного розвитку набули технології децентралізованих обчислень, зокрема Solid та IPFS, які відкривають нові горизонти у створенні соціальних платформ нового покоління. Їх інтеграція дозволяє забезпечити автономне зберігання даних, розподілений доступ до інформації та мінімізацію залежності від централізованих сервісів. У цьому контексті дослідження можливостей застосування Solid та IPFS у сфері соціального нетворкінгу є своєчасним, науково обґрунтованим та важливим для подальшого розвитку безпечного цифрового середовища.

Сучасний етап розвитку інформаційних технологій супроводжується зростанням вимог до конфіденційності, безпеки та контролю над персональними даними. У той час як централізовані соціальні мережі залишаються домінантними у цифровій комунікації, все частіше з’являються критичні зауваження щодо їхньої непрозорості, централізованого контролю та використання особистої інформації в комерційних або політичних цілях. Такі обставини стимулюють інтерес до децентралізованих моделей соціальної взаємодії, в яких користувачі самостійно розпоряджаються своїми даними та мають повний контроль над процесами зберігання, доступу та обміну інформацією.

Актуальність роботи

Сучасні централізовані соціальні мережі, такі як Facebook або Twitter, часто стикаються з проблемами конфіденційності, цензури, зловживання владою над даними користувачів та уразливістю до зовнішніх атак. На цьому тлі децентралізовані підходи, зокрема на базі технологій Solid і IPFS, стають перспективною альтернативою, що дає змогу повернути контроль над даними користувачам, підвищити безпеку, забезпечити більшу прозорість у роботі сервісів і зменшити залежність від монопольних платформ. Дослідження децентралізованих соціальних мереж набуває особливого значення в контексті розвитку Web 3.0 та пошуку шляхів для забезпечення цифрової автономії особистості.

У цьому контексті децентралізовані технології, такі як Solid (Social Linked Data) та IPFS (InterPlanetary File System), формують основу нової парадигми цифрової взаємодії. Solid дозволяє користувачам зберігати персональні дані у власних сховищах (PODs), тоді як IPFS забезпечує ефективне зберігання та обмін контентом у розподіленій файловій мережі. Об’єднання цих технологій створює передумови для побудови соціальних мереж, де головними принципами виступають децентралізація, приватність, стійкість до цензури та відкритість.

Метою даної роботи є дослідження архітектурних підходів, технічних рішень та практичних аспектів імплементації децентралізованих обчислень у соціальному нетворкінгу.

У рамках дослідження здійснено аналіз технологій Solid та IPFS, розглянуто особливості їх інтеграції, реалізовано програмну модель соціального додатку з оцінкою його продуктивності, а також окреслено перспективи розвитку децентралізованого вебу.

Завдання дослідження:

1. Провести аналіз проблем централізованих соціальних мереж.
2. Дослідити принципи роботи технологій Solid та IPFS.

3. Обґрунтувати доцільність їх поєднання у сфері соціального нетворкінгу.

4. Розробити архітектурну модель децентралізованого соціального додатку.

5. Реалізувати програмний прототип на базі Solid та IPFS.

6. Оцінити функціональність, продуктивність і обмеження розробленої системи.

Об’єкт дослідження - системи соціального нетворкінгу та механізми організації обміну даними між користувачами.

Предмет дослідження - технології децентралізованих обчислень, зокрема Solid та IPFS, та їхнє застосування у побудові соціальних мереж.

Методи дослідження

- Аналіз і порівняння існуючих технічних рішень;
- Методологія проектування децентралізованих систем;
- Програмна імплементація;
- Емпіричне тестування та вимірювання продуктивності;
- Формалізація архітектурних моделей.

Наукова новизна

У роботі запропоновано комплексний підхід до інтеграції Solid та IPFS для реалізації соціального нетворкінгу, орієнтованого на конфіденційність і децентралізацію, з оцінкою функціональності й продуктивності в умовах реального використання.

Практичне застосування

Результати роботи можуть бути використані для створення нових типів децентралізованих соціальних сервісів, зокрема в галузях, де особливо важлива конфіденційність і самовизначення користувачів: у журналістиці, громадській діяльності, наукових спільнотах, освітніх ініціативах.

Бакалаврська робота містить 77 сторінок, 29 рисунків, 3 розділи список використаних джерел із 39 найменуваннями.

РОЗДІЛ 1. АНАЛІЗ ЗАСТОСУВАННЯ КОНЦЕПЦІЙ ДЕЦЕНТРАЛІЗОВАНИХ ОБЧИСЛЕНЬ В СОЦІАЛЬНОМУ НЕТВОРКІНГУ

1.1. Особливості побудови децентралізованої соціальної мережі

У контексті сучасної цифрової парадигми, платформи соціальних медіа еволюціонували до статусу критично важливих компонентів повсякденної діяльності, фундаментально трансформуючи процеси комунікації, дистрибуції інформації та соціальної інтеракції. Незважаючи на їх повсюдне розповсюдження та значний вплив, архітектура більшості традиційних соціальних медіа платформ є централізованою. Ця централізація породжує низку значних проблем, зокрема щодо приватності даних, потенційної цензури та концентрації контролю. Існують обґрунтовані занепокоєння стосовно методів обробки користувацьких даних корпоративними структурами та пропонуються різні стратегії їх захисту. Проте, значна частина запропонованих рішень спрямована на коригування існуючих централізованих систем, а не на принципове переосмислення парадигми веб-взаємодії.

1.1.1. Проблематика традиційних централізованих соціальних мереж

Централізована природа більшості соціальних медіа платформ зумовлює ряд системних недоліків. Збір та агрегація величезних обсягів користувацьких даних на централізованих серверах створює значні ризики порушення приватності та компрометації даних. Відсутність повного контролю користувачів над власними даними та непрозорість алгоритмів їх обробки посилюють ці ризики. Крім того, централізований контроль над інфраструктурою та контентом надає власникам платформ необмежені повноваження щодо модерації, що може призводити до цензури контенту та

обмеження свободи вираження поглядів. Ця концентрація влади суперечить принципам відкритості та децентралізації, які були покладені в основу раннього розвитку мережі Інтернет.

У відповідь на ідентифіковані виклики, пов'язані з традиційними соціальними медіа платформами, спостерігається зростаючий науковий та інженерний інтерес до застосування методів децентралізованих обчислень. Ця парадигма пропонує перспективні рішення шляхом розподілу контролю, обробки та зберігання даних по мережі взаємопов'язаних вузлів, усуваючи необхідність покладання на єдиний центральний орган чи сервер. Такий архітектурний зсув імплементує стійкість до цензури, підвищує надійність системи та знижує ризик масових витоків даних.

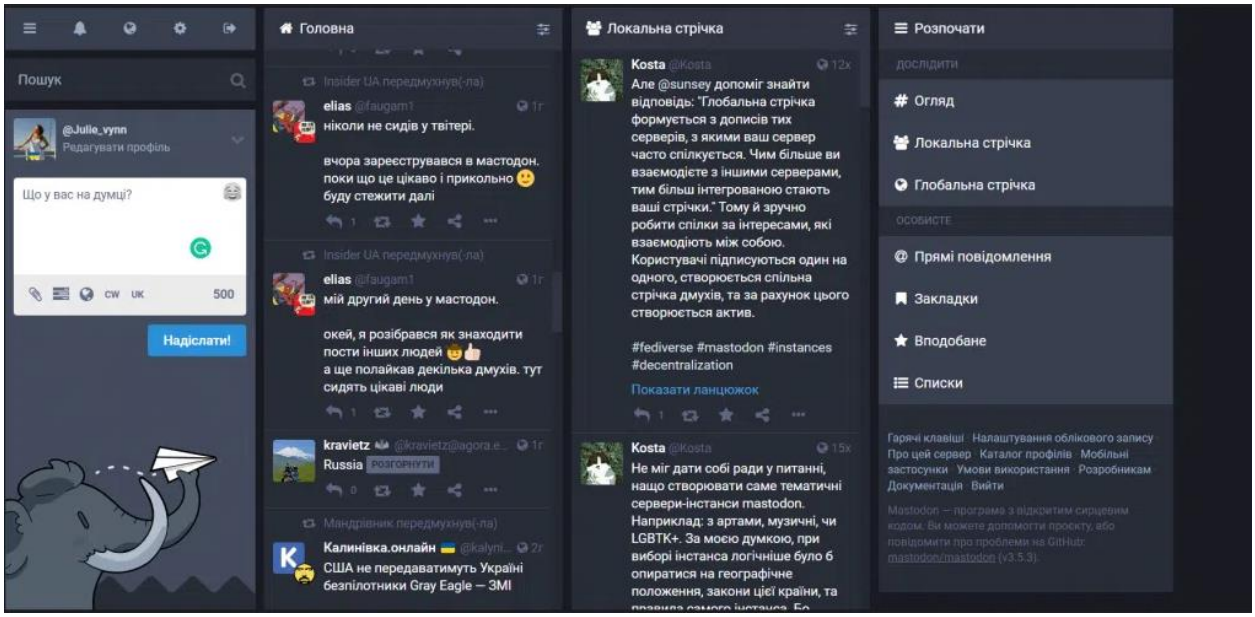

Рисунок 1.1 – Розподілена соціальна мережа Mastodon

На тлі цієї тенденції з'явилися різні децентралізовані соціальні медіа додатки, які прагнуть забезпечити підвищений рівень приватності користувачів та сприяти диверсифікації соціального ландшафту. Наприклад, платформи, побудовані за федеративним принципом, такі як Mastodon, дозволяють користувачам створювати та адмініструвати власні незалежні сервери (інстанси), які модеруються їхніми власниками та спільнотами.

Mastodon — це вільна та відкрита (free and open-source) децентралізована соціальна мережа мікроблогінгу. На відміну від традиційних централізованих платформ, таких як Twitter або Facebook, Mastodon не є єдиним вебсайтом або сервісом, що контролюється однією компанією. Натомість, він складається з великої кількості незалежних серверів, які називаються інстансами (instances). Однак, навіть у таких моделях, існує постійний виклик, пов'язаний з адміністративним навантаженням на управління серверною інфраструктурою, що може призводити до проблем з доступністю контенту або навіть відключень окремих інстансів.

1.1.2. Ключові децентралізовані технології: Solid та IPFS

У контексті подальшої децентралізації веб-простору та соціальних мереж зокрема, особливої уваги заслуговують два помітні децентралізовані інструменти: Solid та IPFS (InterPlanetary File System). Кожен з них пропонує інноваційні підходи до вирішення проблем, властивих централізованим системам.

Solid ставить за мету повернення користувачам контролю над їхніми персональними даними. В основі Solid лежить концепція "Подів" (Pods – Personal Online Data Stores), які є стандартизованими сховищами даних, повністю контрольованими користувачем. У контексті соціальних мереж, Поди можуть зберігати профільну інформацію користувача, контакти (списки друзів/підписок), публікації, медіафайли та інші пов'язані дані. Додатки, сумісні з Solid, можуть отримувати доступ до цих даних (за згодою користувача), але самі дані залишаються власністю користувача та зберігаються в його Поді, а не на серверах додатка. Це кардинально змінює модель взаємодії, передаючи владу над даними від платформи до користувача.

IPFS (InterPlanetary File System) є peer-to-peer протоколом для зберігання та обміну гіпермедіа, який забезпечує децентралізовану файлову

систему. На відміну від традиційного вебу, де контент адресується за місцезнаходженням (URL – Uniform Resource Locator), IPFS адресує контент за його вмістом (CID – Content Identifier). Це означає, що для доступу до файлу не потрібно знати, де він фізично зберігається, а достатньо мати його унікальний ідентифікатор вмісту. Файл може зберігатися на багатьох вузлах мережі одночасно, що забезпечує високу доступність та стійкість до цензури. У застосуванні до соціальних мереж, IPFS може використовуватися для децентралізованого зберігання медіаконтенту (фото, відео), документів, або навіть статичних частин інтерфейсу додатків, забезпечуючи їх постійну доступність незалежно від стану окремих серверів.

1.1.3. Інтеграція solid та IPFS для децентралізованої соціальної мережі

Синергія Solid та IPFS надає потужну основу для розробки справжніх децентралізованих соціальних мереж. Інтегруючи ці технології, запропонований соціальний медіа додаток прагне надати безпечну, стійку до цензури та орієнтовану на користувача платформу. У такій архітектурі:

- Solid використовується для управління ідентичністю користувача, його профілем, списком контактів та метаданими публікацій, забезпечуючи повний контроль користувача над своїми соціальними зв'язками та персональною інформацією.

- IPFS застосовується для децентралізованого зберігання власне контенту публікацій – тексту, зображень, відео та інших файлів. CID вмісту зберігаються в Поді користувача (Solid), зв'язуючи метадані з фактичним контентом, який розподілений по мережі IPFS.

Така інтеграція дозволяє вирішити ключові обмеження традиційних платформ, надаючи користувачам можливість відновити контроль над своїми цифровими ідентичностями та взаємодіями, не покладаючись на централізованих посередників.

Представлена дипломна робота присвячена дослідженню, дизайну, реалізації та оцінці ефективності соціального медіа додатку, побудованого на базі інтеграції Solid та IPFS. Дослідження включатиме аналіз технічної архітектури системи, оцінку її функціональності, дослідження аспектів зручності використання (usability) та практичної реалізації.

Крім того, робота досліджуватиме ширші наслідки застосування цих децентралізованих технологій для еволюції соціальних медіа платформ. Будуть обговорені як потенційні можливості, які вони відкривають (підвищення приватності, стійкість до цензури, розширення прав користувачів), так і виклики, що стоять на шляху їх широкого впровадження (масштабованість, управління ідентичністю в децентралізованому середовищі, модерація контенту, користувацька адаптація).

Отже, сприяючи триваючому науковому та інженерному дискурсу щодо децентралізації та її потенціалу для трансформації сфери соціальних медіа, дане дослідження робить внесок у прокладання шляху до більш справедливого, прозорого та орієнтованого на користувача цифрового майбутнього. Розробка та аналіз децентралізованих рішень, таких як запропонований додаток на базі Solid та IPFS, є важливим кроком у напрямку створення соціальних мереж, які поважають приватність, надають користувачам контроль над їхніми даними та забезпечують стійкість до зовнішнього тиску.

1.2. Застосування Solid та IPFS у децентралізованому соціальному нетворкінгу

У сучасній цифровій екосистемі платформи соціальних медіа зайняли місце ключових інструментів соціальної комунікації та інформаційного обміну, суттєво модифікуючи патерни людської взаємодії. Проте, домінуюча архітектура цих платформ, що базується на централізованій моделі

зберігання та обробки даних, асоціюється з низкою системних ризиків та викликів. Серед найбільш значущих проблем виділяються питання приватності даних користувачів, потенційні механізми цензури та концентрація контролю в руках обмеженого числа корпоративних суб'єктів. Існують широкі наукові та суспільні дискусії щодо методів управління користувацькими даними комерційними організаціями та розробляються різноманітні стратегії їх захисту. Однак, значна частина запропонованих технічних та регуляторних рішень є паліативними заходами, спрямованими на усунення наслідків проблем в рамках існуючої централізованої парадигми, а не на фундаментальне переосмислення архітектури Вебу.

Фундаментальні недоліки централізованих соціальних мереж полягають у їхній монолітній структурі, де користувацькі дані та контроль над функціонуванням платформи зосереджені в єдиній точці. Це створює сприятливі умови для масового збору та потенційного несанкціонованого використання персональної інформації, підвищує вразливість до витоків даних та кібератак на центральні сховища. Крім того, централізований контроль надає власникам платформ дискреційні повноваження щодо модерації контенту, що може призводити до суб'єктивної цензури, обмеження свободи вираження поглядів та формування "інформаційних бульбашок". По суті, користувачі стають не власниками, а продуктом платформи, втрачаючи автономію над своїми цифровими ідентичностями та соціальними графами.

В якості реакції на вищезазначені проблеми централізації, активно розвивається напрямок децентралізованих обчислень та їх застосування у сфері соціального нетворкінгу. Принципи децентралізації передбачають розподіл обчислювальних ресурсів, зберігання даних та контролю між множиною незалежних вузлів мережі, усуваючи залежність від єдиного центрального сервера чи адміністратора. Цей архітектурний зсув імплементує підвищену стійкість до цензури, оскільки відсутній єдиний

пункт відмови, який можна було б легко заблокувати або контролювати. Також знижується ризик масштабних витоків даних, оскільки дані зберігаються розподілено, а не консолідовано в одному сховищі.

Подальший розвиток децентралізованого соціального нетворкінгу тісно пов'язаний з впровадженням фундаментальних децентралізованих протоколів та систем. У цьому контексті особливий інтерес становлять проєкти Solid та IPFS, які пропонують інноваційні підходи до реалізації децентралізованої вебової інфраструктури.

1.2.1. Опис проєкту Social Linked Data

Solid (Social Linked Data) являє собою проєкт, спрямований на повне повернення контролю над даними користувачам. Ключовою концепцією Solid є персональне онлайн сховище даних (Personal Online Data Store - Pod). Под — це стандартизоване веб-сховище, де користувач може зберігати всі свої дані: профільну інформацію, контакти, фотографії, календарні події, фінансові транзакції, а в контексті соціальних мереж — також метадані публікацій (автор, час, зв'язки з іншими публікаціями), список підписок/підписників тощо. Користувач є єдиним власником та контролером свого Поду, розміщуючи його на обраному ним Solid-сумісному сервері (який може бути власним домашнім сервером або послугою від стороннього постачальника). Додатки (наприклад, соціальні мережі, додатки для управління календарем, фінансами тощо) не зберігають дані користувача на своїх серверах. Натомість, вони отримують доступ до даних з Поду користувача (з використанням стандартизованих протоколів авторизації, як-от OAuth) лише з дозволу користувача. Це дозволяє користувачеві використовувати різні додатки, які працюють з одними й тими ж даними у його Поді, забезпечуючи інтероперабельність та усуваючи прив'язку даних до конкретної платформи. У соціальній мережі на базі Solid, ваша цифрова ідентичність, список друзів та історія взаємодій залишаються під вашим

контролем, незалежно від того, який клієнтський додаток ви використовуєте для доступу до мережі.

Платформа Solid максимально спирається на існуючі стандарти W3C для реалізації архітектури, представленої на рисунку 1.2. Платформа специфікує всі протоколи, необхідні для схеми, зокрема протоколи аутентифікації, комунікації "додаток-Под" та комунікації "Под-Под".

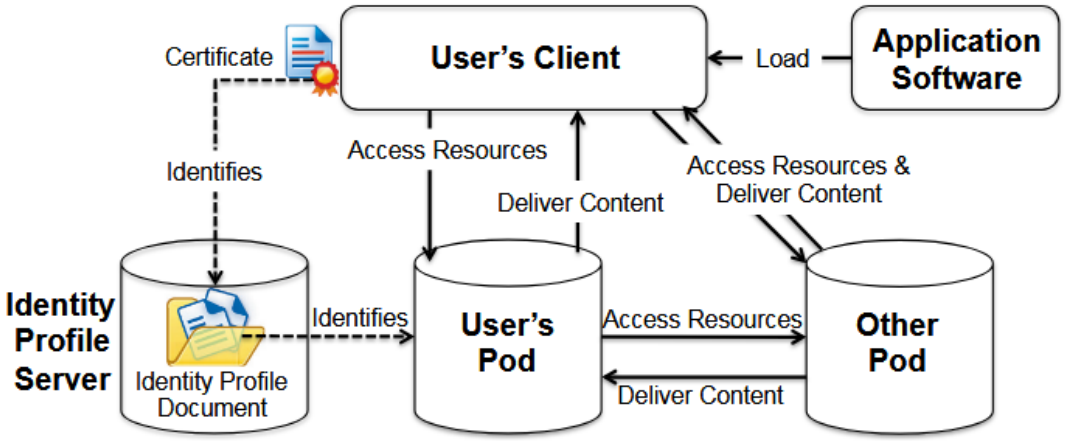

Рисунок 1.2 - Архітектура Solid

Користувач контролює свою ідентичність за допомогою профільного документа RDF, який часто зберігається на сервері Подів. Користувач завантажує Solid-додаток від провайдера додатків. Додаток отримує інформацію про Под користувача з профілю ідентичності. Потім він слідує за посиланнями з профілю, щоб виявити дані у Поді користувача, а також у інших Подах, виконуючи аутентифікацію за необхідності.

У Solid додатки використовують протоколи аутентифікації як засіб для виявлення ідентичності користувача, даних його профілю, а також відповідних посилань, що вказують на Под користувача та дані додатків. Децентралізована аутентифікація, глобальний простір ідентифікаторів та глобальний єдиний вхід (single sign-on) є критично важливими компонентами екосистеми Solid. Solid використовує WebID для забезпечення цих функцій, хоча існують інші рішення, які потенційно можуть взаємодіяти з Solid. У

Solid користувач повинен зареєструватися у провайдера ідентичності, яким, найімовірніше, буде його провайдер Поду. Провайдер ідентичності зберігає профільний документ WebID користувача, пов'язаний із криптографічним ключем.

У Solid дані управляються у RESTful-спосіб, як визначено рекомендацією платформи пов'язаних даних (Linked Data Platform - LDP) [8]. Нові елементи даних створюються у контейнері (який можна назвати колекцією або каталогом) шляхом надсилання їх на URL контейнера за допомогою HTTP методу POST або виконання HTTP методу PUT в межах його URL-простору. Елементи оновлюються за допомогою HTTP методів PUT або PATCH. Елементи видаляються за допомогою HTTP методу DELETE. Елементи знаходяться за допомогою HTTP методу GET та слідування за посиланнями. GET-запит на контейнер повертає перелік елементів, що знаходяться у цьому контейнері.

Дані додатків у Solid зберігаються в документах, які ідентифікуються за допомогою Уніфікованого ідентифікатора ресурсу (URI). Solid розрізняє структуровані дані, представлені за допомогою Фреймворку опису ресурсів (Resource Description Framework - RDF), та неструктуровані дані, які можуть бути будь-якого типу (наприклад, відео, зображення, веб-сторінки тощо). Це дозволяє структурованим даним бути розібраними (парсинг) та серіалізованими у різних синтаксисах, таких як Turtle, JSON-LD (JSON з "контекстом") або RDFa (атрибути HTML). RDF слідує принципам REST, де ресурси мають індивідуальні URI.

Solid-додатки читають та записують дані, що зберігаються у Подах користувачів, за допомогою RESTful HTTP операцій. Окрім підтримки LDP, сервери Solid можуть пропонувати опціональну підтримку SPARQL. Сервери, що підтримують SPARQL, дозволяють додаткам виражати складні операції вибірки даних, включаючи операції, що потребують міжсерверної комунікації за допомогою SPARQL із слідуванням за посиланнями Це

спрощує розробку Solid-додатків, оскільки дозволяє розробнику делегувати складні операції вибірки даних з кількох Подів на сервер.

У Solid сервери Подів є агностичними до додатків, що означає, що нові додатки можуть бути розроблені без необхідності модифікації серверів. Наприклад, навіть незважаючи на те, що LDP 1.0 не містить нічого специфічного для "соціального", багато з Користувацьких сценаріїв Робочої групи Соціального Вебу[6] можуть бути реалізовані в Solid, використовуючи лише LDP та логіку додатків, без необхідності змінювати код на сервері.

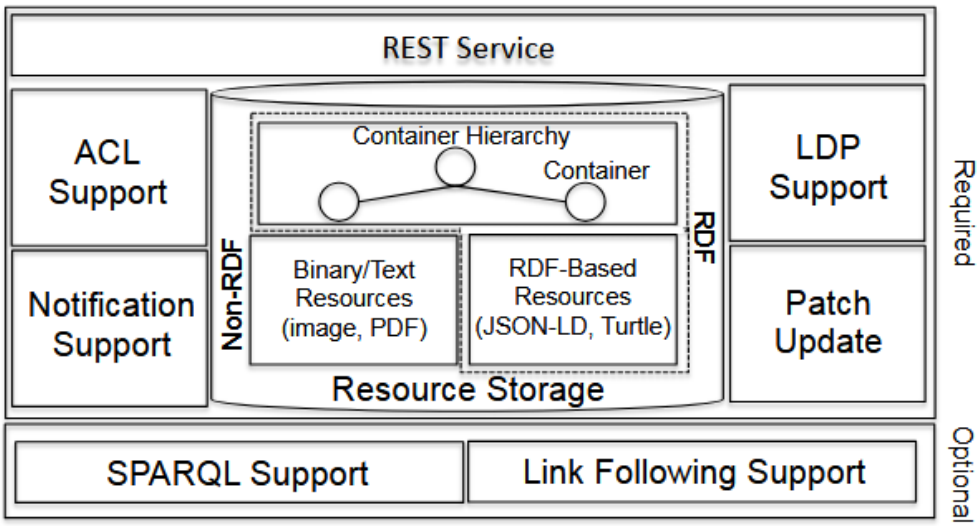

Рисунок 1.3 - Архітектура Поду

Под зберігає RDF-ресурси та не-RDF ресурси. Сервери Подів підтримують LDP, патчинг ресурсів, контроль доступу, оновлення в реальному часі та опціонально SPARQL. Поди використовують LDP для організації даних у контейнери, які групують ресурси, надаючи кожному контейнеру та ресурсу власний URI. Сервер Поду підтримує патчинг ресурсів, списки контролю доступу (ACL), оновлення в реальному часі та опціонально SPARQL. На даний час існує кілька прототипів Solid-серверів, зокрема gold, ldnode, ldphp та meccano, які наразі доступні як публічні сервіси у Вебі.

1.2.2. Розподілена система InterPlanetary File System

IPFS (InterPlanetary File System) є розподіленою peer-to-peer системою для зберігання та обміну файлами. На відміну від традиційної клієнт-серверної моделі HTTP, яка адресує файли за їхнім місцезнаходженням (URL сервера), IPFS адресує файли за їхнім вмістом.

Після завантаження файлу в IPFS, IPFS генерує хеші цих даних та повертає їх власнику. У класичному сценарії генеруються робочі вузли (worker nodes), і їхні пари публічного та приватного ключів зберігаються у смарт-контрактах. Коли власник отримує хеш від IPFS, він здійснює пошук у смарт-контракті верифікованих робочих вузлів, які відповідальні за надання послуг дешифрування клієнтам.

Послуги може надавати лише той робочий вузол, який був обраний власником. Після отримання хешу власник використовує алгоритм Шаміра для розподілу секрету (SSS) для розділення хешу IPFS файлу на k частин (шарів).

У відповідь на отримання цих частин власник визначає n випадкових ключів, які будуть використані для їх шифрування. Після того, як усі частини зашифровані, вони зберігаються у блокчейні разом з іншою важливою інформацією, такою як, наприклад, авторизований отримувач файлу. Безпека даних забезпечується шляхом шифрування саме хешів. Пояснення цьому полягає в тому, що сам по собі хеш не є безпечним. Він лише представляє унікальний цифровий відбиток для певних даних. Якщо будь-який неавторизований клієнт, який не вніс депозит за цифровий контент, отримує доступ до хешу, то повний файл даних з IPFS може бути легко вилучений. У такому випадку власник зазнав би повних бізнес-втрат.

У традиційному сценарії тільки ті клієнти здатні дешифрувати хеш, які внесли депозит і були авторизовані робочими вузлами. Загальний потік завантаження файлу в IPFS власником показано на рисунку 1.4.

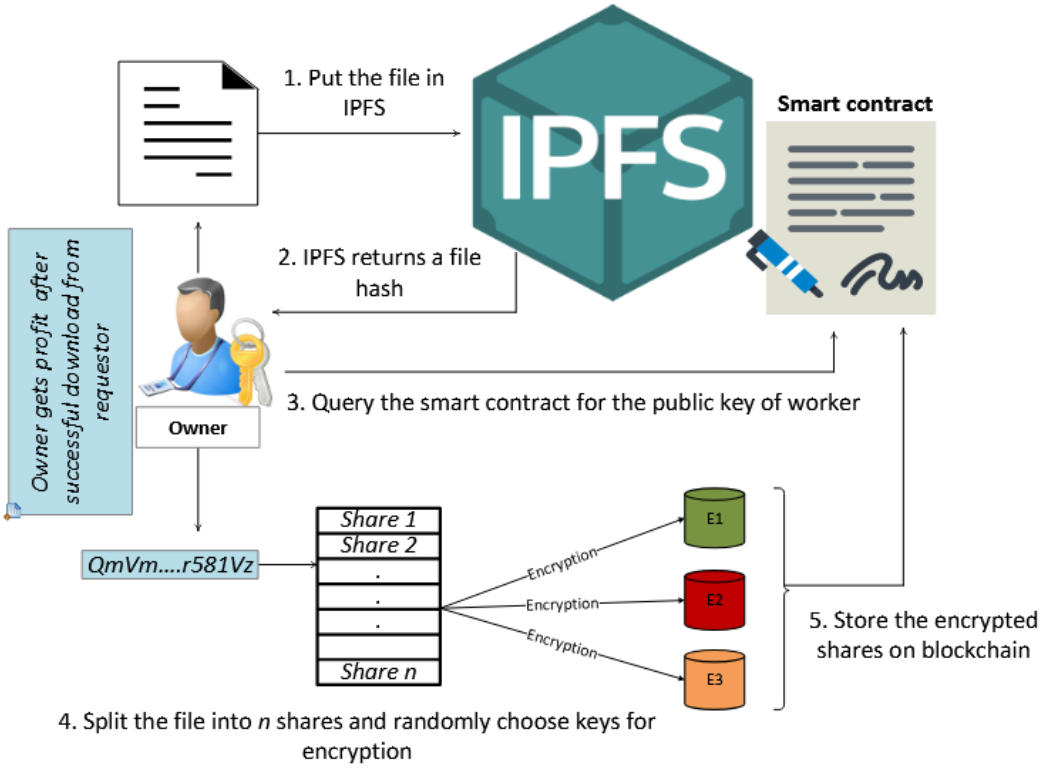

Рисунок 1.4 – Обмін даними у IPFS власником

Розглянемо секрет S, який має бути розділений на n частин таким чином, щоб кожна частина даних могла бути представлена як S1,…,Sn.

1. Комбінація k або більшої кількості Si дозволяє легко обчислити (відновити) оригінальний файл.

2. В іншому випадку, якщо є k−1 або менша кількість Si, тоді буде складно реконструювати фактичні дані, отже S стає невизначеним. Це відоме як (k,n) порогова схема, де k — мінімальна кількість частин для відновлення, а n — загальна кількість частин. Випадок, коли k=n, означає, що всі задіяні суб'єкти повинні брати участь для реконструкції фактичного секрету S. В нашому випадку розмір розділення (загальна кількість частин, n) та необхідний поріг (кількість частин для відновлення, k) взяті як 5 та 3 відповідно (тобто, використовується порогова схема (3, 5)). Коли частини згенеровані за допомогою алгоритму SSS, ці частини шифруються та зберігаються у смарт-контракті. Ці частини можуть бути дешифровані тільки

верифікованими робочими вузлами, чий публічний ключ шукається власником під час завантаження файлу в IPFS.

Детальна системна модель для клієнта, який запитує цифровий контент, показана на рисунку 1.5.

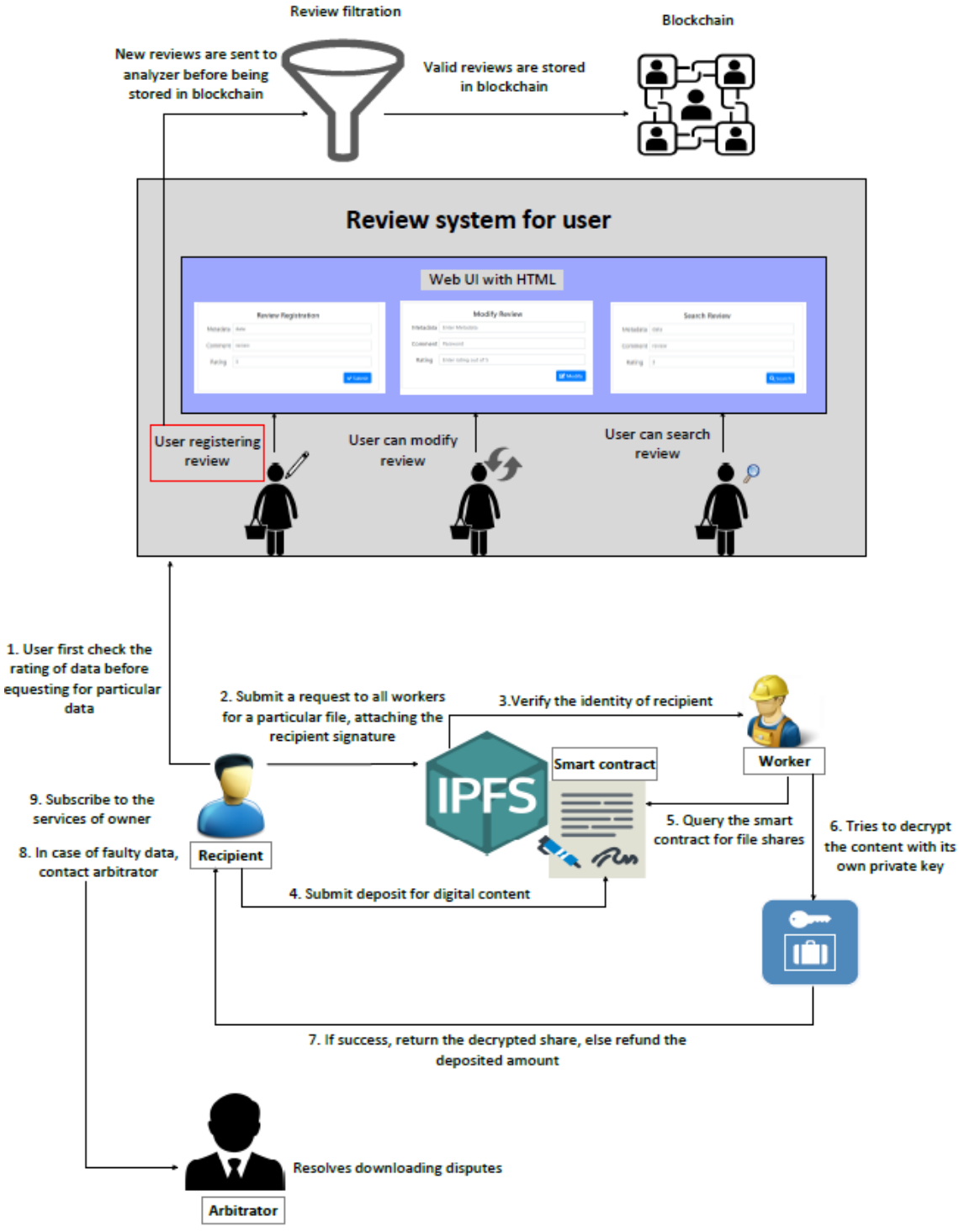

Рисунок 1.5 - Запит даних отримувачем

Клієнт спершу перевіряє існуючі відгуки від попередніх клієнтів, щоб якість даних могла бути належним чином перевірена перед внесенням ефірів.

Для обміну даними (торгівлі даними) між власником та клієнтом доступна система репутації, призначена для рецензування даних, що є об'єктом обміну. Зауважимо, що терміни "відгук" (review), "рейтинг" (rating) та "репутація" (reputation) використовуються як взаємозамінні. Перш за все, система відгуків надає рейтинги або оцінки, надані клієнтами, які вже використовували дані. Базуючись на цих відгуках, нові покупці можуть перевірити якість, надійність та цілісність даних. Незмінність відгуків забезпечується шляхом використання блокчейну, оскільки дані у блокчейні є стійкими до несанкціонованих змін. Після перевірки рейтингів, новий клієнт вирішує, чи варто купувати дані, чи ні. Після того, як клієнт задоволений відгуками, подається запит робочим вузлам для доступу до бажаного цифрового контенту. З цією метою розгортається смарт-контракт з функціями: set reviews, get reviews та get ratings. Коли клієнт отримує дані, він може подати власні відгуки, використовуючи функцію set reviews. Ці відгуки зберігаються у блокчейні як результат кожної транзакції, виконаної під час подання відгуку. Функція Isreviewexist дозволяє новим клієнтам перевірити, чи є якісь доступні відгуки для конкретних даних. Якщо ні, то цілісність та автентичність даних не можуть бути гарантовані з боку користувача.

Кожен файл у IPFS ідентифікується унікальним криптографічним хешем, який називається ідентифікатором вмісту (Content Identifier - CID). Коли користувач запитує файл за його CID, мережа IPFS шукає вузли, які мають копію цього файлу, і завантажує його з одного або кількох доступних вузлів. Це має значні переваги для соціальних мереж:

- Стійкість до цензури та доступність. Оскільки файл може зберігатися на багатьох вузлах одночасно, його важко заблокувати або видалити з мережі

повністю. Якщо один вузол стає недоступним, файл все ще можна отримати з інших вузлів.

- Ефективність. Якщо багато користувачів запитують один і той самий файл, вони можуть отримувати його з найближчих вузлів, що знижує навантаження на джерела та зменшує затримки.

- Перевірка цілісності. Оскільки CID є хешем вмісту, будь-яка зміна файлу призведе до зміни його CID. Це гарантує, що отриманий файл є саме тим, який був запитаний, і не був змінений під час передачі або зберігання. У соціальних мережах IPFS ідеально підходить для децентралізованого зберігання мультимедійного контенту (зображень, відео, аудіо), документів, або навіть статичних елементів інтерфейсу додатків, забезпечуючи їхню розподіленість, доступність та цілісність.

1.2.3. Інтеграція Solid та IPFS для децентралізованої соціальної мережі

Інтеграція Solid та IPFS пропонує архітектурну основу для створення децентралізованої соціальної мережі нового покоління. У такій моделі:

1. Solid використовується для управління ідентичністю користувача, його соціальним графом (хто на кого підписаний, хто є "другом"), профільною інформацією та метаданими публікацій (дата, час, автор, посилання на вміст). Усі ці дані зберігаються у Поді користувача, повністю під його контролем.

2. IPFS використовується для зберігання власне вмісту публікацій – тексту, зображень, відео, документів. Коли користувач створює публікацію з медіафайлами, ці файли завантажуються в мережу IPFS, отримують свій унікальний CID. Цей CID потім зберігається у Поді користувача в якості посилання з метаданими публікації.

Така синергія дозволяє створити соціальну платформу, де користувач володіє своїми даними (Solid) та контентом (IPFS), які розподілені по мережі,

а не зосереджені на серверах єдиної компанії. Клієнтський додаток соціальної мережі в цій моделі функціонує як інтерфейс, який читає дані з Подів користувачів (за їхнім дозволом) та отримує контент з мережі IPFS для відображення стрічки новин, профілів тощо.

Представлена робота має на меті дослідити можливості та виклики реалізації соціальної медіа платформи на базі інтеграції Solid та IPFS. Дослідження включатиме розробку архітектури такого додатку, його практичну реалізацію та оцінку ефективності. Зокрема, буде проаналізовано технічну архітектуру системи, її функціональні можливості, аспекти зручності використання та практичну реалізованість в реальних умовах.

Додатково, в рамках дослідження буде розглянуто ширший контекст застосування цих децентралізованих технологій для трансформації ландшафту соціальних медіа. Будуть обговорені потенційні переваги, які надають Solid та IPFS (підвищена приватність, стійкість до цензури, розширення прав користувачів), а також існуючі виклики (масштабованість мереж Solid та IPFS, забезпечення ефективної модерації в розподіленому середовищі, складність розробки децентралізованих додатків, адаптація користувачів до нової парадигми).

Внесок даної роботи полягає у сприянні розвитку децентралізованого соціального нетворкінгу шляхом аналізу та практичної реалізації рішень на базі передових децентралізованих технологій Solid та IPFS. Дослідження дизайну, реалізації та оцінки ефективності запропонованої платформи допоможе краще зрозуміти потенціал цих технологій для створення більш справедливих, прозорих та орієнтованих на користувача соціальних медіа, сприяючи еволюції цифрового ландшафту у напрямку більшої автономії та контролю користувачів над їхньою цифровою присутністю.

РОЗДІЛ 2. АРХІТЕКТУРНІ ПІДХОДИ ТА ТЕХНОЛОГІЇ ВІД ЦЕНТРАЛІЗАЦІЇ ДО ДЕЦЕНТРАЛІЗАЦІЇ В СОЦІАЛЬНОМУ НЕТВОРКІНГУ

2.1. Традиційні практики соціальних медіа

Домінуюча архітектура традиційних платформ соціальних медіа базується на централізованій моделі. У такій конфігурації єдиний централізований сервер є основним вузлом обробки та зберігання даних, до якого підключаються всі користувачі. Взаємодія з контентом, включаючи створення та управління публікаціями та асоційованими даними, здійснюється виключно через інтерфейс та функціонал, що надається цією централізованою платформою. Незважаючи на потенційну ефективність такого підходу для масового обміну даними, виникають значні складнощі, коли користувач намагається повністю видалити свої дані з платформи. Часто дані не підлягають повному незворотному видаленню, натомість вони можуть ставати лише недоступними для публічного перегляду користувачем. Така практика потенційно дозволяє власникам платформ зберігати, використовувати або навіть маніпулювати цими даними без явного контролю з боку їх первісного власника.

Однією з фундаментальних проблем, притаманних централізованим платформам соціальних медіа, є обмежений контроль користувачів над власними даними та їхньою власністю. Коли користувачі завантажують цифровий контент, такий як зображення, відео або текстові публікації, вони, по суті, передають певний обсяг прав на використання та управління цими даними адміністраторам платформи. Хоча інтерфейс платформи може надавати інструменти для управління або видалення опублікованого контенту, базові дані, що лежать в його основі, часто продовжують зберігатися на серверах платформи протягом невизначеного терміну.

Процедура видалення даних з централізованих соціальних медіа платформ, як правило, характеризується непрозорістю та відсутністю чітких гарантій повного знищення інформації. Навіть коли користувачі ініціюють видалення своїх публікацій або цілих облікових записів, немає абсолютної впевненості в тому, що їхні дані будуть повністю та фізично стерті з серверів платформи. У багатьох випадках запити на видалення даних здебільшого призводять до маркування даних як "видалених" або "неактивних" у базах даних, без повного фізичного видалення з основних сховищ даних [7]. Ця практика породжує значні ризики для приватності даних, оскільки зберігається потенційна доступність інформації для адміністраторів платформи або афілійованих третіх сторін, навіть після того, як користувач вважає дані видаленими.

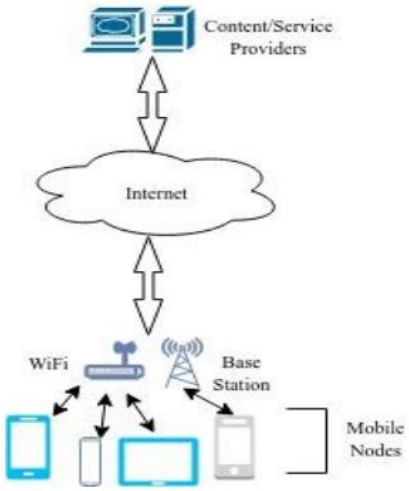

Рисунок 2.1 - Архітектура традиційних мобільних соціальних мереж

Рисунок 2.1 показує архітектуру традиційних мобільних соціальних мереж (МСМ). У цій моделі мобільні вузли обмінюються інформацією/даними з іншими мобільними вузлами за допомогою мережі Інтернет, використовуючи WiFi або базові станції як точки доступу. Інформація зберігається у провайдера контенту/сервісів, який виступає як централізований сервер. У деяких випадках архітектура МСМ також

функціонуватиме у децентралізований спосіб. Тут замість централізованих серверів роль відіграють локальні сервери, а мобільні вузли комунікують між собою через інтранет.

Централізований характер традиційних платформ також робить їх вразливими до цензури та маніпуляції контентом. Концентрація контролю надає адміністраторам платформ повноваження здійснювати модерацію та цензуру контенту, базуючись на їхніх внутрішніх політиках. Це може спричиняти упередженість у модерації, призводити до дискримінації та обмежувати свободу вираження поглядів.

Крім того, централізована інфраструктура є більш схильною до інцидентів витоку даних та вразливостей безпеки. Єдина точка відмови або компрометація центрального сервера може поставити під загрозу персональну інформацію величезної кількості користувачів одночасно.

Таким чином, незважаючи на те, що традиційні платформи соціальних медіа кардинально змінили парадигму онлайн-комунікації та взаємодії, їх централізована природа генерує суттєві проблеми у площині володіння даними, забезпечення приватності та безпеки. Відповідно, спостерігається зростаючий інтерес до альтернативних, децентралізованих архітектурних підходів до соціальних медіа, зокрема платформ, що базуються на технологіях блокчейну та однорангових (peer-to-peer) мереж. Метою цих підходів є надання користувачам розширеного контролю над їхніми даними та елімінація потреби у централізованих посередниках.

2.2. Досягнення та виклики соціальних мереж на основі блокчейну

У сфері децентралізованого соціального нетворкінгу відбулися значні досягнення, зокрема у розробці платформ, що базуються на технологіях блокчейну та імплементують хостинг контенту через блокчейн. Однією з таких концепцій є проєкт Ushare [8]. Архітектура Ushare базується на

використанні дозвільного (permissioned) блокчейну. У такій моделі доступ користувачів до мережі є обмеженим і вимагає проходження верифікації від уповноваженої третьої сторони для отримання дозволу на взаємодію з блокчейном [8]. Цей підхід контрастує з недозвільними (permissionless) блокчейнами, які є публічними, дозволяють анонімним користувачам вільно брати участь та надавати свою обчислювальну потужність для підтримки мережі. Рішення використовувати дозвільний блокчейн у Ushare обґрунтовується необхідністю забезпечення підзвітності та відстежуваності даних, що поширюються в мережі.

Крім використання блокчейну, додаток Ushare застосовує розподілені хеш-таблиці (Distributed Hash Tables - DHT) для забезпечення ефективного отримання даних. Зазначено, що ефективність цього механізму значно зростає зі збільшенням кількості користувачів та вузлів у мережі. Використання DHT для зберігання та отримання контенту, тоді як блокчейн використовується, ймовірно, для метаданих, контролю доступу або реєстрації транзакцій, переводить проєкт від моделі суто децентралізованого хостингу даних до більш розподіленої мережі, де вузли розташовані розподілено, а хеш-таблиці забезпечують зв'язування та адресацію даних.

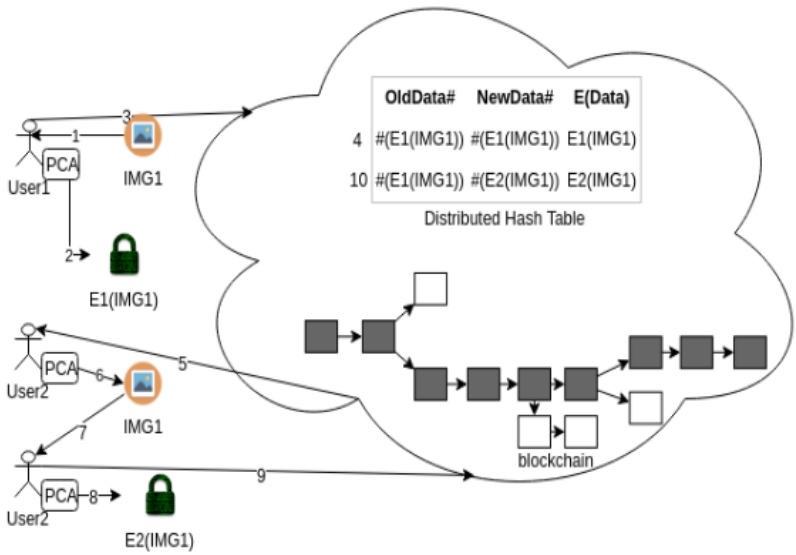

Рисунок 2.2 - Робочий потік для створення хеш-таблиці

Рисунок 2.2 показує робочий потік для створення хеш-таблиці. Користувач, User1, отримує зображення IMG1(1) і шифрує його(2) публічним ключем кола, з яким воно буде поширено, використовуючи PCA. Потім користувач зберігає зашифроване зображення у хеш-таблиці Ushare(3), використовуючи хеш свого зашифрованого зображення як OldData# та NewData#(4). Оскільки користувач є власником зображення, старий і новий хеш-ключі є однаковими. Інший користувач, User2, є членом кола, який має доступ до поширеного зображення. Після проходження блокчейну, користувач може отримати доступ до зображення з хеш-таблиці(3) та дешифрувати його(6). Цей користувач тепер повторно поширює це зображення(7) зі своїм колом, шифруючи його публічним ключем кола(8). Це зашифроване зображення зберігається у хеш-таблиці(10) з хеш-ключем попереднього шифрування як oldData# та його новим хешем як NewData#.

Підтримка окремої хеш-таблиці для зберігання даних поза блокчейном дозволяє забезпечити передбачуване зростання блокчейну. Оскільки валідація блоків зазвичай потребує завантаження повного ланцюга вузлами, що беруть участь у мережі, більші блокчейни створювали б значні обчислювальні навантаження.

Користувач, який бажає поширити елемент даних зі своїм колом, створює першу трансляцію як транзакцію, вказуючи свою ідентичність як адресу "відправника" (from) та хеш-ключ зашифрованого елемента даних як адресу "отримувача" (to). Транзакція також містить "значення токена", яке визначає кількість дозволених поширень для цього елемента даних.

Далі користувач транслює кілька транзакцій, кожна з яких містить хеш-ключ зашифрованих даних як адресу відправника та ідентичність членів кола як адресу отримувача. Значення токена також присутнє в цих транзакціях. Будь-які поширення, здійснені з цим елементом даних, створюють іншу транзакцію з ідентичністю нового користувача як адресою відправника та хеш-ключем даних, зашифрованих ключем кола цього нового користувача,

як адресою отримувача. Після цього знову здійснюється кілька транзакцій для членів кола цього користувача, вказуючи новий хеш-ключ як відправника та ідентичність членів кола користувача як адресу отримувача.

Рисунок 2.3 показує переходи станів. Транзакція містить адресу відправника, адресу отримувача, значення даних та цифровий підпис. Стан містить криптографічний одноразовий код (nonce), який є довільним числом, що може бути використане лише один раз. Виконуваний код, що використовує систему взаємозв'язків, перевіряє, чи є транзакція дійсною з точки зору дозволеної кількості поширень. Якщо транзакція дозволена, він зменшує значення токена на одиницю. Також присутнє сховище даних, яке містить сліди поширень та його значення токена.

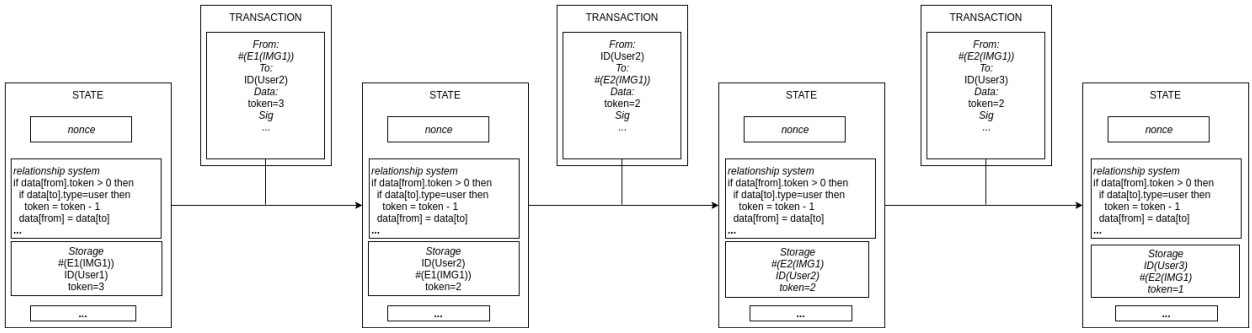

Рисунок 2.3 - Переходи станів

Ushare успадкує технічні аспекти від усталених блокчейнів для валідації блоків, реалізації Доказу виконаної роботи (Proof of Work) або Доказу частки володіння (Proof of Stake), підтримки децентралізованої бази даних, стимулів для майнінгу та інших процедур.

Незважаючи на інноваційні архітектурні рішення, проєкт Ushare має певні недоліки, зокрема пов'язані з залежністю від блокчейну Ethereum для створення або підтримки функціонування мережі. Необхідність використання Ethereum, що передбачає сплату комісій за транзакції (т.зв. "gas fees"), створює значну перешкоду для широкого впровадження платформи.

Багато користувачів можуть бути неохочими сплачувати за розміщення свого контенту, особливо в епоху, коли існує безліч централізованих платформ для безкоштовного обміну контентом. Вимога прямих платежів за використання функціоналу блокчейну може відлякувати потенційних користувачів, особливо тих, хто звик до моделі безкоштовних онлайн-сервісів.

Крім того, залежність від Ethereum може спричинити проблеми масштабованості та непередбачуваності витрат. Комісії за транзакції Ethereum та перевантаження мережі можуть призвести до затримок в обробці операцій та збільшення витрат для користувачів, що може перешкодити широкому впровадженню платформи. Також, волатильність ціни Ethereum може внести невизначеність для користувачів щодо довгострокової доступності та вартості хостингу їхнього контенту на Ushare.

Однак, важливо визнати потенційні переваги інтеграції блокчейну, зокрема Ethereum, у децентралізовану екосистему соціальних медіа. Вимагаючи від користувачів сплати за розміщення контенту, Ushare потенційно може стимулювати створення контенту вищої якості. Користувачі з більшою ймовірністю інвестуватимуть свій час та зусилля у створення цінного контенту, коли вони мають фінансову зацікавленість у самому процесі публікації. Ця модель "плати за публікацію" (pay-to-publish) може слугувати механізмом фільтрації низькоякісного контенту або спаму, сприяючи підтримці вищого стандарту матеріалів на платформі. Проте, як зазначено, цей механізм також може утримати значну частину потенційних користувачів від приєднання до платформи.

2.3 Переваги застосування системи InterPlanetary File System (IPFS)

IPFS являє собою набір однорангових (peer-to-peer) протоколів, що використовуються для адресації, маршрутизації та передачі даних у рамках

децентралізованої файлової системи. IPFS оперує децентралізованою мережею, яка складається з комп'ютерів, що беруть участь у мережі, виконуючи реалізацію протоколів IPFS. Ці комп'ютери іменуються вузлами. Ключовою особливістю IPFS є використання адресації на основі контенту, на відміну від традиційної адресації на основі розташування. Це означає, що дані, збережені за допомогою IPFS, отримують унікальний ідентифікатор контенту (Content Identifier - CID). Цей CID є криптографічним хешем вмісту, він описує самі дані та використовується для безпосереднього доступу до опублікованих даних, незалежно від їх фізичного розташування. У традиційному підході, що базується на розташуванні, дані зберігаються з урахуванням їх прив'язки до конкретного місця, наприклад, за посиланням HTTP, яке вказує на сервер та шлях до файлу.

Оскільки IPFS використовує децентралізовану мережу вузлів, система не має єдиної точки відмови. Це робить її ідеальним рішенням для хостингу даних соціальних медіа мереж, оскільки навіть якщо один або кілька вузлів відключаться від мережі, це не призводить до порушення функціонування всієї системи. За умови, що хоча б один інший вузол у мережі зберігає копію даних, які знаходилися у відключеного вузла, система не втратить доступ до цих даних.

Децентралізована природа IPFS також сприяє підвищенню продуктивності. Вузли мережі можуть бути розташовані в різних географічних місцях, що дозволяє користувачам отримувати доступ до даних з найближчого доступного вузла. Це можливо завдяки тому, що дані ідентифікуються та можуть бути отримані на основі їхнього вмісту (через CID), а не їх розташування на централізованих серверах. Таким чином, децентралізована мережа може забезпечувати швидший час отримання даних, якщо в мережі поблизу запитувача є інші вузли, що мають запитувані дані. Використання цієї децентралізованої можливості дозволяє зменшити загальні проблеми продуктивності, такі як затримка (latency) та

перевантаження мережі, зі збільшенням кількості вузлів. Це дає змогу вірусним постам у соціальних медіа завжди залишатися доступними та масштабуватися відповідно до попиту на їх перегляд, на відміну від традиційних централізованих платформ, де пост може швидко стати недоступним через перевантаження єдиного сервера.

Щоб опублікувати або отримати об'єкт, необхідно створити відображення (мапінг) між CID та PeerID, який може надати цей об'єкт (включаючи його Multiaddress). Для функціонування у децентралізований спосіб та підтримки виявлення контенту й вузлів, ці відображення індексуються у Розподіленій хеш-таблиці (DHT), яка надає прості примітиви PUT та GET. DHT IPFS базується на алгоритмі Kademlia, подібно до тієї, що використовується BitTorrent Mainline DHT. CID та PeerID знаходяться у спільному 256-бітному просторі ключів шляхом використання SHA256-хешів їхніх бінарних представлень (див. Рисунок 1) як ключів індексування.

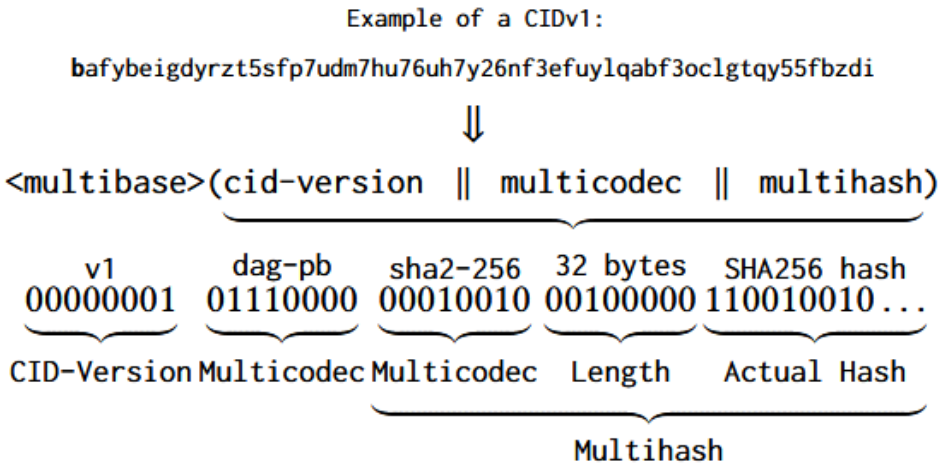

Рисунок 2.4 – Структура CID

На основі нашого практичного досвіду з робочою мережею, ми внесли низку коригувань порівняно з оригінальною специфікацією Kademlia. Вузли у DHT використовують 256-бітні SHA256-ключі замість 160-бітних SHA1-ключів. Це зроблено для передбачення прогресу у навмисних хеш-колізіях.

Ми також підтримуємо i=256 сегментів (buckets) по k вузлів кожен (де k=20) для розділення хеш-простору. Нарешті, ми використовуємо надійні транспортні протоколи, такі як TCP та QUIC (замість UDP), оскільки це робить управління з'єднаннями у реалізації більш прямолінійним.

Нові вузли приєднуються до DHT або як DHT-сервери, якщо вони мають публічне IP-підключення, або як DHT-клієнти, якщо вони не є публічно досяжними, наприклад, тому що вони знаходяться за пристроєм перетворення мережевих адрес (NAT). IPFS розрізняє DHT-клієнтів та серверів за допомогою простої техніки під назвою Autonat. Autonat працює наступним чином: нові вузли за замовчуванням приєднуються як клієнти і негайно просять інші вузли в мережі ініціювати з'єднання у відповідь до них. Якщо більше трьох вузлів можуть підключитися до вузла, що щойно приєднується, тоді новий вузол підвищує свою участь, щоб діяти як серверний вузол. Якщо більше трьох вузлів не можуть підключитися, вузол продовжує працювати як клієнт.

DHT-сервери виконують усі мережеві операції, тобто зберігання контенту, зберігання записів відображення та надання їх вузлам-запитувачам. На відміну від цього, DHT-клієнти лише запитують записи або контент з мережі, але не зберігають і не надають жодних з них. Розрізнення DHT-клієнт/сервер запобігає включенню недосяжних вузлів до таблиць маршрутизації інших вузлів, таким чином прискорюючи процеси публікації та отримання.

На рисунку 2.5 показано як публікується контент та як інші вузли можуть його знаходити та отримувати.

Процес публікації включає:

1. Імпорт контенту у локальний IPFS-процес Провайдера та присвоєння CID;

2. Обхід DHT для пошуку найближчих вузлів до CID;

3. Зберігання запису про провайдера у найближчих вузлів. Процес отримання включає п.4 і п.5.

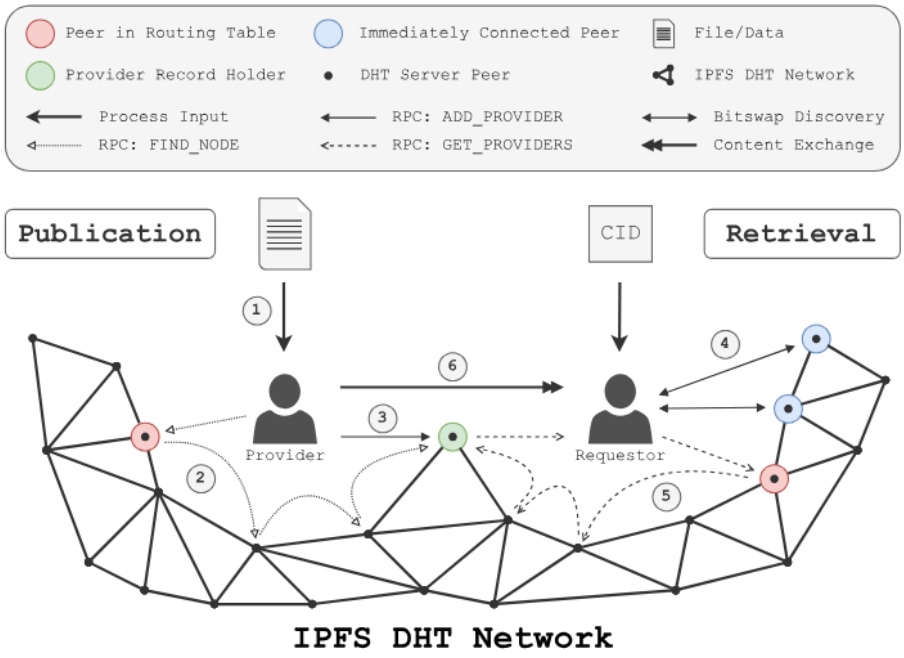

Рисунок 2.5 - Публікація та отримання в IPFS

4. Опортуністичні запити Bitswap до вже підключених вузлів для CID;

5. Обхід DHT для пошуку вузла, що зберігає запис про провайдера;

6. Запитувач підключається до Провайдера та отримує контент. На діаграмі опущено другий Обхід DHT Запитувача для розв'язання PeerID провайдера до його мережевої адреси.

Ще однією важливою проблемою, яку вирішує IPFS, є залежність від постачальника (vendor lock-in). У традиційному підході ця залежність може зробити перехід до іншого постачальника послуг непрактичним або надзвичайно складним. IPFS усуває ці труднощі при переході між платформами та міграції даних. Відповідно, якщо користувач захоче мігрувати публікації з іншої платформи соціальних медіа, яка також використовує протокол IPFS, він може зробити це дуже легко, що значно розширює автономію користувача щодо його цифрових даних.

Як було зазначено раніше, IPFS є набором peer-to-peer протоколів. Це означає, що для взаємодії з мережею та обробки й поширення даних необхідні реалізації цих протоколів. Одна з таких реалізацій, яка використовується в цьому проєкті, — це Helia [15]. Ця реалізація дозволяє використовувати функціонал IPFS у серверних та браузерних середовищах, що працюють на базі JavaScript.

Helia — це свіжий погляд на те, як може виглядати повнофункціональна реалізація IPFS, написана на JavaScript. З самого початку js-IPFS задумувався як API-сумісний клон go-IPFS, і з того часу його перейменували на Kubo. З часом це почало втрачати сенс, оскільки до Kubo додавалося все більше і більше функцій, js-IPFS насилу встигав за оновленнями, і навіть тоді не всі додані функції мали сенс у контексті JS.

Хоча Helia сумісна з Kubo та іншими інтерфейсами на дроті, вона не обмежена Kubo API та може вільно впроваджувати інновації, що розширюють можливості використання JS. Наприклад, немає потреби дотримуватися універсального підходу Kubo/js-IPFS, натомість користувачі можуть залишати свої програми легкими, але використовувати лише ті функції, які вони використовують.

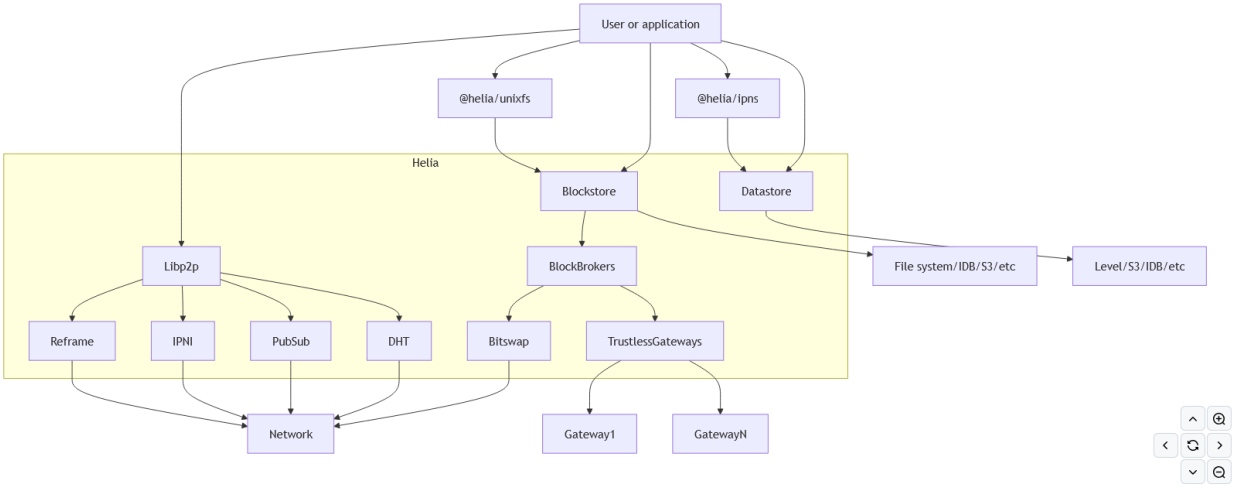

Рисунок 2.6 – Системна діаграма проекту

Реалізація Helia заснована на бібліотеці libp2p, яка є проєктом з відкритим вихідним кодом. libp2p часто використовується для побудови реалізацій розподілених систем, зокрема IPFS, завдяки його спрямованості на забезпечення однорангової комунікації.

2.4. Концепція виявлення неправдивих відгуків на основі блокчейн та IPFS

Для виявлення неправдивих онлайн-відгуків (фейкових відгуків) представлена система, показана на рисунку 2.7.

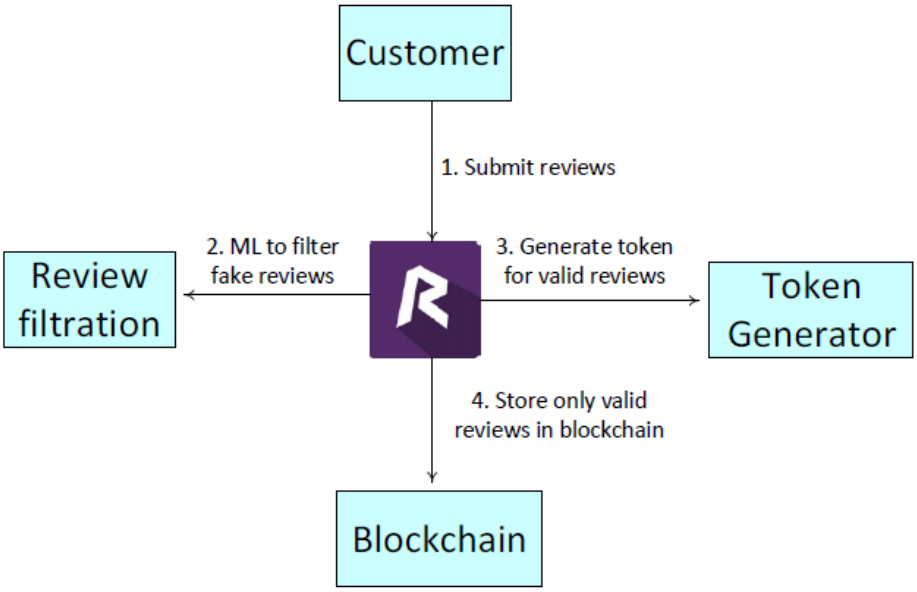

Рисунок 2.7 – Метод виявлення неправдивих онлайн-відгуків

На першому етапі клієнт подає відгук до додатку виявлення, відомого як аналізатор Watson. Інструмент спершу фільтрує неправдиві відгуки на основі аналізу емоційного стану, що визначається за способом написання відгуку. Це завдання виконується з використанням методів штучного інтелекту. Після цього відфільтровані відгуки проходять додаткову перевірку (перехресну перевірку) компанією або власником продукту. Перехресна перевірка відгуків власником є важливою, оскільки здебільшого ринкові

конкуренти реєструють негативні відгуки про продукт з метою зниження його ринкової вартості. Власник може відхилити або прийняти відгуки після верифікації. У випадку, якщо відгуки відхиляються як інструментом фільтрації, так і компанією, причини відхилення повідомляються клієнтам для забезпечення прозорості. Прийняті відгуки зберігаються у блокчейні та не можуть бути змінені через їхню незмінність.

Різні випадки реєстрації неправдивих відгуків представлені наступним чином:

- Маніпуляція оцінками (Vote Manipulation).

Ситуація, коли власник просить своїх друзів або співробітників розмістити неправдиві відгуки щодо якості продукту.

- Купівля відгуків (Vote Buying).

Коли компанії наймають спеціальних співробітників для розміщення відгуків, використовуючи фейкові електронні адреси та імена користувачів, сплачуючи їм за це значну суму грошей.

- Відгук про невідповідний об'єкт (Vote on Wrong Thing).

Ця категорія не стосується неправдивих відгуків як таких, але здебільшого клієнти фактично залишають відгук не про той продукт, щодо якого вони мали б його зареєструвати. Наприклад, клієнт забронював номер у готелі, і йому може не сподобатися пляж поблизу готелю. Тоді клієнт, найімовірніше, залишить негативні відгуки про готель, хоча його незадоволення пов'язане з зовнішнім чинником.

Кожного разу, коли клієнт отримує дані, він надсилає запит усім робочим вузлам. Запит супроводжується цифровими підписами клієнтів. Новий запитувач спершу ідентифікується робочими вузлами за допомогою RSA-підписів. RSA — це асиметрична криптографічна система, де генеруються пари публічного та приватного ключів, де приватний ключ є захищеним і зберігається користувачем, а публічний ключ поширюється серед інших.

```
// Генерація пари ключів RSA
key0.generateKeyPair(2048, 65537);
fs.writeFileSync("privateKey_0.pem", key0.exportKey('pkcs8-private-pem'));
fs.writeFileSync("publicKey_0.pem", key0.exportKey('pkcs8-public-pem'));
console.log("0 - RSA Key Pairs saved.");
```

Рисунок 2.8 – Генерація ключів

Після того, як ідентичність отримувача верифікована робочими вузлами, отримувач має внести ефір як плату за цифровий контент. Після здійснення оплати, робочі вузли приступають до пошуку зашифрованих частин (шарів) у смарт-контракті, як було запитано. Якщо отримувач неавторизований або його ідентичність не підтверджена як дійсного запитувача, тоді смарт-контракт перериває транзакцію, і робочі вузли ігнорують отриманий запит.

З іншого боку, у випадку успішної верифікації та оплати, робочі вузли отримують n зашифрованих частин з блокчейну та дешифрують їх, використовуючи свої приватні ключі. Важливо відзначити тут, що робочий вузол може дешифрувати частини від імені отримувача лише, якщо його публічний ключ був обраний власником під час поширення контенту в IPFS (як було описано в попередніх розділах). Після дешифрування отримувач може отримати оригінальний хеш IPFS файлу шляхом його реконструкції (відновлення) з дешифрованих частин.

Після того, як дані були завантажені клієнтом, очікується реєстрація оцінки даних від цього клієнта. Оцінка даних (дата-рейтинг) відображає якість даних та автентичність власника. Майбутні клієнти можуть легко перевірити якість, переглядаючи відгуки про конкретні дані.

Щоб забезпечити отримання клієнтом оновлених версій цифрового контенту, розгортається смарт-контракт з функцією підписки. Ця функція дозволяє клієнту підписатися на послуги власника. Кожного разу, коли власник даних модифікує існуючий контент або випускає нові редакції файлу, усі попередні клієнти, що мають підписку, отримують сповіщення про

вихід нової версії. Перевага контролю версій полягає в тому, що існуючі клієнти, які вже придбали контент від того ж власника та сплатили певну суму у вигляді цифрової валюти, не повинні знову сплачувати повну вартість. Вони або сплачують певний відсоток від фактичної ціни, або не сплачують нічого, залежно від умов підписки. Таким чином, клієнти отримують оновлений контент, не сплачуючи повну вартість повторно. Ці послуги підписки допомагають існуючим клієнтам отримувати найновіші редакції документів.

Для мотивації клієнтів до реєстрації автентичних відгуків запроваджено механізм стимулювання. Існує пропозиція для клієнтів, що 10% від фактично внесених коштів (депозиту) відшкодовуються тим, хто подає дійсні відгуки. Коли смарт-контракт отримує сповіщення про реєстрацію відгуку певним клієнтом, тоді відгуки валідуються (перевіряються), як показано на рисунку 2.7. Смарт-контракт відшкодовує 10% від фактично внесеної суми, якщо відгук не є неправдивим. У разі реєстрації неправдивого відгуку жодна сума клієнту не відшкодовується. Подібним чином, новий клієнт може шукати відгуки про дані у системі, що базується на відгуках. Відгуки зберігаються у блокчейні та можуть бути отримані за запитом клієнта. Схема стимулювання клієнта показана на рисунку 2.9.

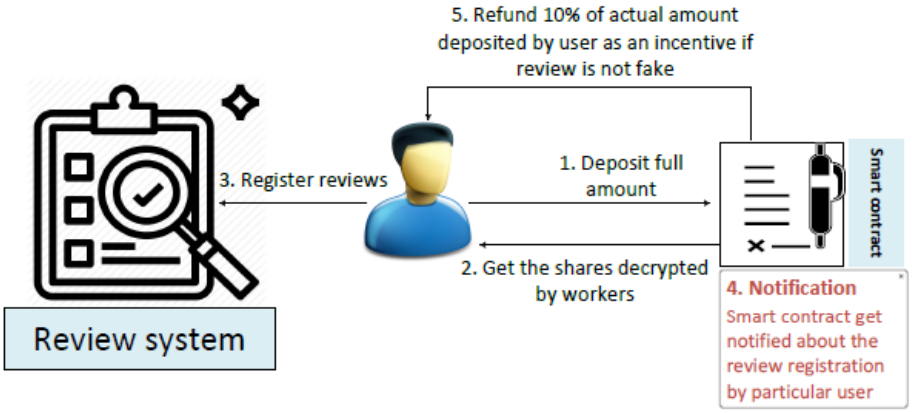

Рисунок 2.9 - Стимулювання користувача за реєстрацію відгуків

Діаграма послідовності для всіх задіяних суб'єктів мережі та їхньої взаємодії в сценарії обміну даними показана на рисунку 2.10.

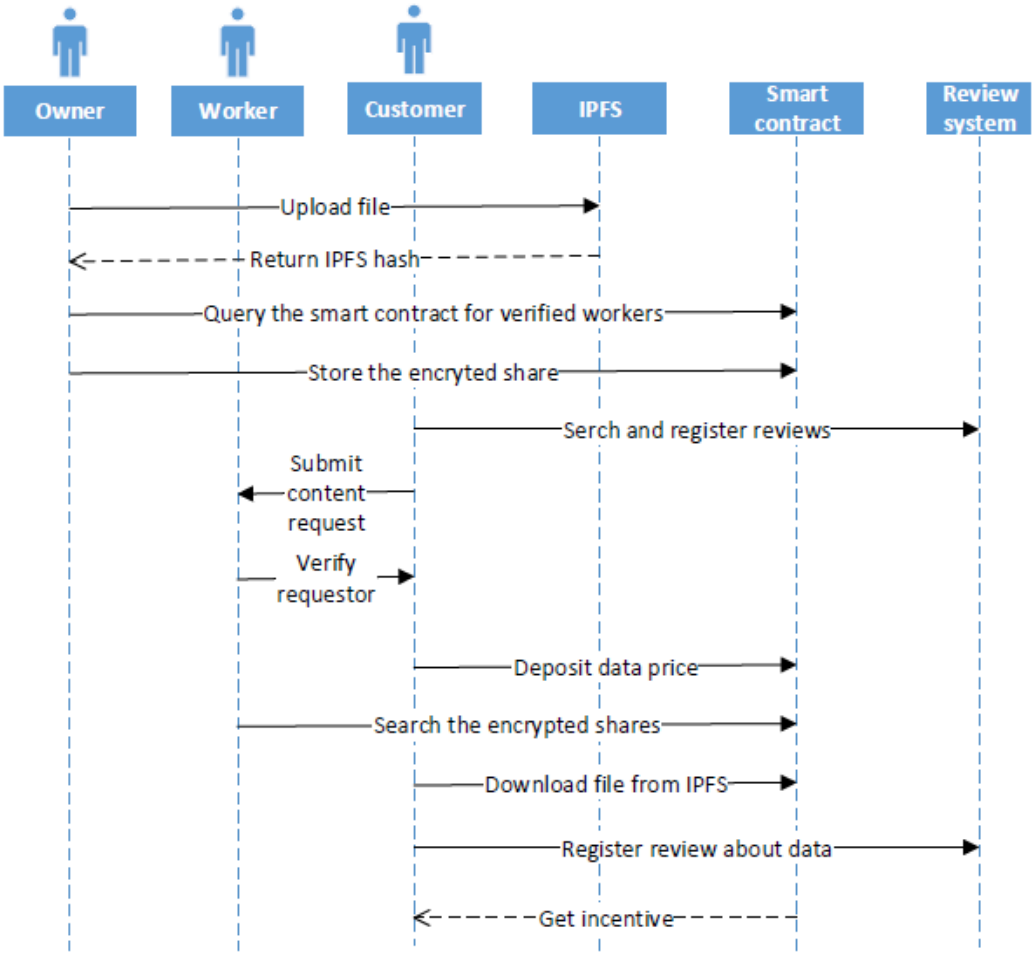

Рисунок 2.10 - Діаграма послідовності сценарію обміну даними

Автентичність даних забезпечується шляхом додавання системи, що базується на відгуках, де клієнти можуть записувати свої коментарі та оцінки щодо даних. Таким чином, нові клієнти можуть оцінити якість даних, отже, можна заощадити кошти на стороні клієнта. Різні смарт-контракти розроблені для різних цілей; таких як, смарт-контракт власника для завантаження файлу в IPFS та шифрування хешів. Інший смарт-контракт призначений для сторони користувача. Користувач може отримати доступ до хешів файлів зі смарт-контракту після аутентифікації від робочих вузлів.

2.5. Проєкт Solid як децентралізована парадигма даних у соціальному нетворкінгу

Проєкт Solid є проєктом з відкритим вихідним кодом Solid і призначений для зменшення рівня контролю, що здійснюється великими корпоративними структурами, такими як Meta, Amazon, Apple, Netflix та Google, над діяльністю та контентом, які користувачі бачать і з якими взаємодіють онлайн, шляхом контролю над їхніми персональними даними.

Ключовою перевагою проєкту Solid є надання користувачам можливості контролювати місце зберігання їхніх даних. Використовуючи децентралізовані сховища даних, які називаються Подами (Pods), користувачі можуть зберігати власні дані та управляти доступом до них, визначаючи, хто може отримувати доступ та переглядати їхню інформацію. Це кардинально змінює парадигму контролю над даними користувачів, що здійснюється великими корпораціями, оскільки сховище даних підтримується користувачем, і дані можуть дійсно належати цьому користувачеві.

Ресурси даних у цих Подах зберігаються та представляються як файли Resource Description Framework (RDF), ідентифіковані певним Уніфікованим ідентифікатором ресурсу (URI). Усі дані в межах Подів можуть бути представлені в цьому форматі пов'язаних даних (linked data).

У контексті соціальних медіа цей формат використовується для представлення сутностей, таких як користувачі (люди), контент (публікації, медіа) та інших потенційних ідентифікаторів, які можуть зберігатися в екосистемі Solid. До цих ресурсів можна отримати доступ за допомогою URI, який називається WebID. WebID — це URI, що вказує на документ RDF профілю WebID. Цей профіль містить інформацію, таку як посилання на сховище Поду WebID та кінцеві точки аутентифікації, як показано на рисунку 2.11.

Рисунок 2.11 - Схема функціонування Solid у додатках соціальних медіа

Solid вирізняється не лише ефективним зберіганням даних у Подах, а й надає детальний контроль доступу, визначаючи, хто може переглядати та змінювати певні контейнери (папки) та документи. Використовуючи списки контролю доступу (ACL - Access Control Lists), користувачі можуть управляти дозволами на читання (read), запис (write) та додавання (append). Дозвіл на запис (write) дозволяє уповноваженому суб'єкту створювати нові ресурси, оновлювати існуючі ресурси та видаляти ресурси. Дозвіл на читання (read) дозволяє лише переглядати або отримувати ресурс. Нарешті, дозвіл на додавання (append) дозволяє додавати контент до існуючого ресурсу, що є особливо корисним у контексті соціальних медіа, де необхідно дозволити іншим користувачам додавати коментарі до ваших публікацій та взаємодіяти з ними без надання повного права на запис.

Реалізації проєкту сприяє значна підтримка у вигляді бібліотек JavaScript. Існують різноманітні методи та функції в їхній бібліотеці JavaScript, які вже реалізують базовий функціонал аутентифікації та контролю доступу, що є важливим для підключення користувачів до екосистеми та управління їхніми дозволами. З використанням цих бібліотек було реалізовано низку проєктів, таких як додатки для планування та календарів, менеджери контактів та реалізації мікроблогінгу. Однак при роботі з публікаціями (постами) мікроблогів виникали певні проблеми, пов'язані, зокрема, з особливостями редагування та синхронізації змін у

розподіленому середовищі, оскільки власник зберігає повний контроль над вихідними даними публікації.

Завдяки значній підтримці у вигляді бібліотек JavaScript, цей проєкт може бути успішно завершений без необхідності реалізовувати базовий функціонал Solid з нуля, а натомість зосередитися на реалізації та специфічних проблемах, які можуть виникнути при застосуванні Solid для розробки платформи соціальних медіа.

РОЗДІЛ 3. ПРОГРАМНА ІМПЛЕМЕНТАЦІЯ МЕТОДІВ ДЕЦЕНТРАЛІЗОВАНИХ ОБЧИСЛЕНЬ В СОЦІАЛЬНОМУ НЕТВОРКІНГУ

Розробка цього додатку соціальних медіа базувалася на використанні сучасних веб-фреймворків. Для реалізації фронтенду (клієнтської частини), призначеної для взаємодії з користувачем, застосовувалися TypeScript та React. Серверна частина (бекенд), що обробляє основну інтерактивність додатку, була реалізована за допомогою Express. Функціонал інтерактивності включає створення публічних або приватних публікацій (постів), взаємодію з публікаціями (наприклад, шляхом позначки "подобається" або додавання коментарів), а також можливість підписуватися на інших користувачів додатку. Децентралізовані аспекти реалізації, що охоплюють імплементацію протоколів Solid та IPFS для зберігання публікацій та взаємодії з ними, були реалізовані за допомогою зовнішніх бібліотек, зокрема Helia. Використання цих бібліотек суттєво спростило процес інтеграції децентралізованих компонентів та забезпечило безперешкодну взаємодію з обраним провайдером Solid та сервісами IPFS.

Комбінація зазначених фреймворків та технологій дозволила ефективно продемонструвати можливість реалізації децентралізованої платформи соціальних медіа та підтвердити її переваги для забезпечення приватності даних користувачів шляхом використання Solid та IPFS.

3.1 Імплементація Solid у додатку соціальних медіа

Як було зазначено раніше, Solid є специфікацією, що дозволяє користувачам зберігати дані у децентралізованих сховищах даних, які називаються Подами (Pods). Для використання розробленого додатку користувач повинен мати обліковий запис у провайдера Подів або ж

самостійно розгорнути та хостити власний Под. Для цілей цього демонстраційного проєкту було обрано варіант використання Поду від одного з наявних провайдерів (перелік деяких провайдерів представлено на рисунку 3.1), оскільки він є більш типовим для більшості кінцевих користувачів. Зокрема, для реалізації проєкту було обрано провайдера Подів Inrupt, завдяки наявності численних ресурсів та бібліотек JavaScript, що полегшили налаштування аутентифікації у веб-сервері на базі Node.js. Використання цих ресурсів виявилося цінним, оскільки це дозволило успішно виконати один з перших етапів налаштування протоколу Solid для доступу до даних користувачів у додатку.

Постачальник	Відповідальний за доменне ім'я та умови	Відповідальний за хостинг	Місце хостингу	Реалізація Solid
Inrupt Pod Spaces	Inrupt Inc.	Amazon	Німеччина	ESS
solidcommunity.net	Проєкт Solid	Digital Ocean	Великобританія	NSS
solidweb.org	Solid Grassroots	Hosteurope	Франція	NSS
trinood.us	Grasshmetrix Inc.	Amazon	США	TrInPod
use.id	Dietta	Digital Ocean	ЄС	CSS
solidweb.me	Meisdata	Hosteurope	Франція	CSS
Data Pod	iGrant.io Sweden	RedPill Linero, AWS, GCP або Azure	ЄС	NSS
redpencil.io	redpencil.io	Hetzner	Фінляндія	CSS
teamid.five	Meisdata	Hosteurope	Франція	CSS

Рисунок 3.1 - Список постачальників Pod

Після успішної аутентифікації користувача у додатку, у його Solid Pod автоматично ініціалізується певна структура контейнерів (папок). Ці контейнери містять інформацію, яка є релевантною для функціоналу соціальних медіа додатку, включаючи, зокрема, усі публікації, створені користувачем, та перелік користувачів, на яких він підписаний.

Для управління доступом до певних контейнерів та документів генеруються списки контролю доступу (ACL - Access Control Lists). Для забезпечення видимості публічних публікацій іншим користувачам створюється спеціальний контейнер "public", який налаштовується за допомогою ACL Solid. Цей публічний контейнер автоматично створюється для користувача, якщо він бажає створити публічну публікацію. Після створення публікації її унікальна адреса (URI) зберігається в його Solid Pod. Якщо користувач бажає опублікувати приватну публікацію, яка видима лише певній групі користувачів, додаток автоматично налаштовує відповідні ACL для надання доступу конкретним WebID, визначеним користувачем. Використання функціоналу Solid та ACL дозволяє додатку забезпечити приватність та контроль над даними користувача, не передаючи права власності на ці дані ні серверу додатку, ні іншим користувачам платформи.

Після успішної публікації користувачем у своєму Solid Pod, публікація стає видимою для суб'єктів, яким користувач надав відповідний дозвіл через ACL. Це також означає можливість створення повністю приватних публікацій, які не є видимими для інших за замовчуванням. Ця можливість реалізована з урахуванням сценарію, коли користувач може бажати опублікувати щось приватно, а потім згодом надавати або відкликати доступ іншим користувачам. Однак через певні обмеження ця функціональність не була реалізована у повному обсязі в межах поточного проєкту. Ще однією реалізованою функцією, що відповідає популярним практикам сучасних соціальних медіа платформ, є можливість користувачів змінювати свої імена користувачів (нікнейми). Імена користувачів часто виступають як персоналізовані ідентифікатори. Реалізована інтеграція забезпечує зв'язок WebID користувачів з їхніми обраними іменами користувачів, що дозволяє користувачам за необхідності верифікувати справжні ідентичності інших суб'єктів.

Щоб зберегти дані публікації в Pod користувача, дані повинні бути збережені у файл Resource Description Framework (RDF). Причина використання формату RDF замість інших альтернатив, таких як JSON або простий текстовий файл, полягає в наявності контексту та взаємозв'язків між елементами даних, що є ключовим для представлення соціальних зв'язків та контенту. Використання RDF забезпечує можливість передачі даних до інших додатків соціальних медіа без необхідності враховувати специфічні формати даних, що сприяє інтероперабельності. Використовуючи загальноприйнятий словник ідентифікаторів RDF, такий як Schema.org, було зіставлено наступні частини даних в межах кожної публікації з певними ідентифікаторами (наприклад, для автора, дати створення, тексту публікації), як показано на рисунку 3.2.

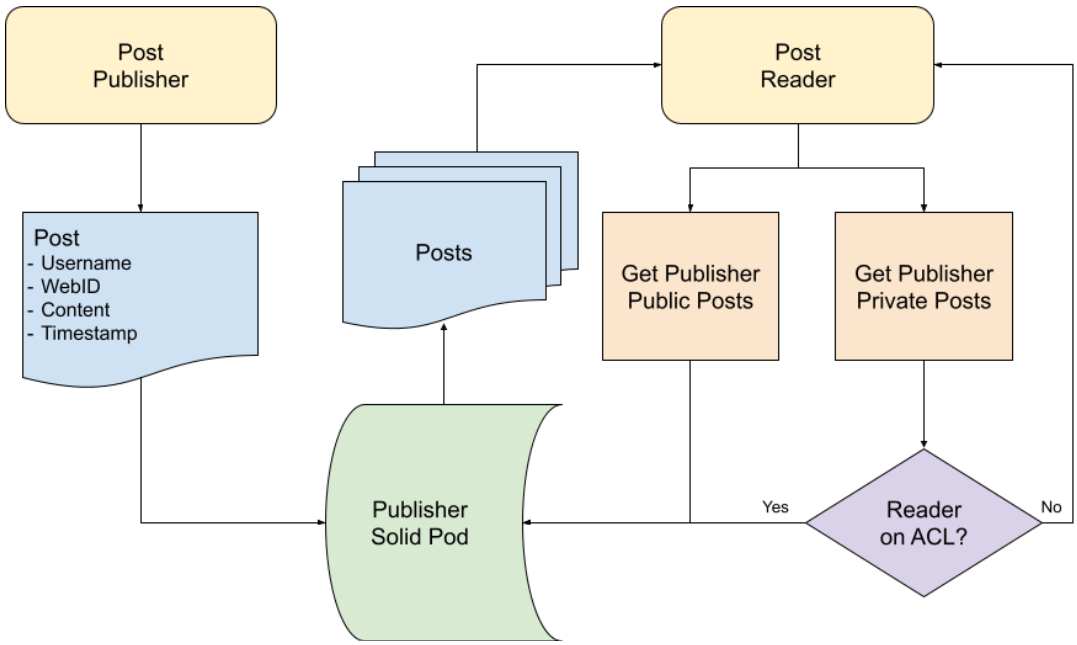

Рисунок 3.2 - Схема функціонування Solid у додатках соціальних медіа

3.2 Застосування IPFS у додатку соціального нетворкінгу

Використання файлової системи IPFS у контексті реалізованого додатку соціальних медіа є відносно простим. Завдяки особливостям

архітектури протоколу, IPFS може ефективно забезпечувати хостинг та отримання даних, а також гарантувати їхню перевіреність (verifiability), оскільки сам вміст даних хешується в унікальний ідентифікатор контенту (CID - Content Identifier). Ці властивості протоколу були використані для того, щоб зробити публікації загальнодоступними та видимими для широкого кола користувачів. IPFS дозволяє автору публікації зробити її видимою та кешованою усіма суб'єктами мережі. Теоретично, зі збільшенням кількості користувачів, які звертаються до публікацій, швидкість їх отримання може бути значно вищою порівняно з традиційними методами хостингу за рахунок розподіленого кешування.

Для імплементації функціоналу IPFS у цьому проєкті було використано бібліотеку JavaScript Helia. Ця бібліотека спеціалізується на реалізації протоколу IPFS у середовищах JavaScript, включаючи серверні та браузерні середовища. Використовуючи Helia, реалізовано функціональність додавання вмісту публікацій до мережі IPFS та їх отримання з неї. У цьому демонстраційному проєкті застосовувався режим офлайн-зберігання, проте реалізація сумісна з онлайн-режимом IPFS, який для забезпечення постійного зберігання даних незалежно від звернень користувачів протягом тривалого часу потребує використання сервісу піннінгу (закріплення даних).

Процес збереження даних в IPFS передбачає створення унікального ідентифікатора CID. Цей CID генерується шляхом криптографічного хешування вмісту, що забезпечує незмінність даних. Після збереження даних в IPFS їх не можна змінити. Крім того, дані, що зберігаються в IPFS, є публічно доступними за їхнім CID, навіть для користувачів, які не використовують початкову платформу, через яку дані були завантажені.

Для забезпечення можливості інтеракцій, таких як позначки "подобається" та коментарі до публікацій, що вимагають змінності даних, було використано систему імен IPNS - InterPlanetary Naming System. Використання IPNS дозволяє забезпечити змінність шляхом надання

змінного вказівника, який може вказувати на різні, але незмінні фрагменти контенту, збережені в IPFS. Таким чином, коли користувач публікує пост, у IPNS створюється унікальний вказівник на цю публікацію. Цей вказівник потім зберігається у 'Поді' автора публікації (Solid Pod). Коли інший користувач бажає переглянути опубліковану публікацію, він запитує контейнер публічних публікацій автора у його Поді, який містить вказівники IPNS на найновішу версію публікації, збережену в IPFS. Оскільки IPNS забезпечує механізм змінності, щоразу, коли користувач взаємодіє з опублікованою публікацією (наприклад, ставить позначку "подобається" або додає коментар), додаток створює копію поточних даних публікації в IPFS, а потім додає до неї запитувану інтеракцію. Ці оновлені дані потім зберігаються у новому файлі в IPFS (з новим CID), і додаток присвоює вказівник IPNS цій новій адресі (новому CID), таким чином оновлюючи вказівник на останню версію публікації.

3.3 Інтеграція та комбіноване застосування Solid та IPFS у соціальному нетворкінгу

Враховуючи переваги та сценарії застосування Solid та файлової системи IPFS, було прийнято рішення використати ці технології у тандемі для досягнення цілей проєкту. Solid, будучи переважно орієнтованим на користувача протоколом, надає користувачам детальний контроль над тим, хто саме та в якому обсязі може отримувати доступ до даних, що зберігаються у їхньому Поді. Цей функціонал є аналогічним до можливостей контролю видимості активності облікового запису у традиційних додатках соціальних медіа, але реалізований на рівні власності даних. Використання Solid як основного механізму зберігання та отримання публікацій є високоефективним, оскільки архітектура додатку спроєктована з метою

мінімального вторгнення у дані кінцевого користувача, залишаючи їх під повним контролем власника.

Ще однією суттєвою перевагою використання Solid є можливість безперешкодної міграції даних між різними платформами, що підтримують цей протокол. Якщо користувач вирішить перейти на іншу платформу, сумісну з Solid, йому достатньо буде здійснити аутентифікацію за допомогою свого наявного Поду на новій платформі, і всі його публікації та дані стануть там доступними. Це зумовлено тим, що дані публікацій користувача зберігаються виключно у його власному Поді, а не на сервері, яким керує провайдер конкретного додатку.

У тандемі з Solid, використання IPFS та системи імен IPNS надає значні переваги, особливо для роботи з публічним та змінним контентом. Замість того, щоб користувачі безпосередньо отримували інформацію (зміст публікації) з Поду автора, вони отримують вказівник IPNS. Цей вказівник, що зберігається у Поді видавця (наприклад, у контейнері публічних публікацій), перенаправляє їх до найновішої версії публікації, збереженої в IPFS, яка може містити оригінальний пост, а також асоційовані інтеракції, такі як позначки "подобається" та коментарі (реалізовані як оновлення даних, на які вказує IPNS). Важливою особливістю використання IPNS є також те, що автор публікації зберігає можливість видалити або оновити вказівник IPNS. Це може зробити стару версію контенту менш легкодоступною через IPNS, однак сам ресурс контенту в IPFS залишатиметься доступним, доки інші вузли мережі здійснюють його піннінг (закріплення даних). Ця властивість IPFS сприяє зменшенню цензури, оскільки інформація, одного разу опублікована в IPFS, потенційно може зберігатися в мережі незалежно від дій початкового видавця, доки її підтримують інші вузли.

Таким чином, використання IPFS (через IPNS) для публічних публікацій у додатку дозволяє відображати справжню динамічну взаємодію з аудиторією, оскільки інтеракції відображаються через оновлення версій

контенту, на які вказує змінний вказівник IPNS. Це відрізняється від моделі, де автор публікації мав би повний і незмінний контроль над представленим контентом після його початкової публікації. Проте, для випадків, де повний контроль над контентом та доступом є пріоритетним, існує опція приватної публікації, що реалізується через механізми Solid ACL та обмежує можливість публічної інтеракції.

3.4. Оцінка продуктивності реалізованої системи

У розробленому додатку було використано протоколи Solid та IPFS для зберігання та отримання контенту, включаючи публікації та інтеракції користувачів. Була проведена оцінка ефективності використання зазначених протоколів шляхом аналізу часу відгуку (response time) для різних типів викликів API, які забезпечують основний функціонал додатку. Отримані результати класифіковано за протоколом, що використовується (Solid або IPFS). До переліку тестованих викликів API включено операції створення публічних та приватних публікацій, надсилання коментарів до публічних публікацій, позначки "подобається" для публічних публікацій, управління доступом для приватних переглядачів, отримання публічних та приватних публікацій, а також підписка на інших користувачів та доступ до списку користувачів, на яких здійснено підписку. Статистичні дані, представлені нижче, зібрано на основі 100 запитів для кожного типу виклику API.

Таблиця 3.1 - Час відгуку GET API для протоколу Solid

API	Час відповіді (мс)
Приватний пост	688.15
Приватний переглядач	985.39
Підписка	820.41

Таблиця 3.2 - Час відгуку POST API для протоколу Solid

API	Час відповіді (мс)
Приватний пост	1041.93
Приватний переглядач	1938.94
Підписка	1034.47

Таблиця 3.3 - Час відгуку API для протоколу IPFS

API	Метод HTTP	Час відповіді (с)
Публічний пост	POST	36.64
Публічний пост	GET	1.27
Коментар до поста	POST	30.22
Лайк до поста	POST	30.23

Як свідчать дані, представлені у таблицях вище, час відгуку API для операцій отримання даних (GET запитів) з Pods (Solid) або мережі IPFS є відносно низьким. Зокрема, отримання приватного посту через Solid займає менше секунди, а отримання публічного посту через IPFS – трохи більше секунди. Це демонструє практичність даного рішення для сценаріїв отримання контенту у платформах соціальних медіа.

Незважаючи на те, що час, необхідний для виконання запитів на публікацію (POST запитів), є значним (наприклад, понад 30 секунд для публікацій в IPFS), це зумовлено природою протоколу IPFS, зокрема необхідністю закріплення даних (pinning) на кількох різних вузлах мережі для забезпечення їх постійного зберігання та доступності. Хоча тривалість цих операцій може здаватися високою порівняно з традиційними централізованими системами, цей аспект не становить критичної проблеми для загальної зручності використання платформи, оскільки забезпечення доступності даних менш ніж за хвилину є прийнятним показником з точки

зору користувацького досвіду, особливо враховуючи переваги децентралізації та стійкості до цензури.

3.5. Функціональність реалізованого додатку соціального нетворкінгу

Розроблений додаток успішно демонструє базові функціональні можливості, що є фундаментальними для функціонування платформи соціальних медіа в децентралізованому середовищі. Ці можливості включають створення публікацій (постів), взаємодію з публікаціями, а також механізм підписки на користувачів для отримання доступу до їхніх публікацій. Додатково реалізовано функцію приватних публікацій, доступ до яких обмежений користувачами, включеними до списку контролю доступу (ACL) видавця.

Початковий етап взаємодії користувача з додатком – це процес аутентифікації через провайдера Solid Pod, екран якого показано на рисунку 3.3.

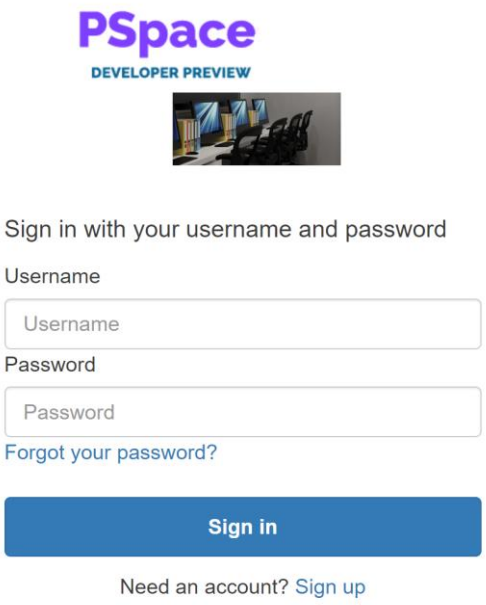

Рисунок 3.3 - Екран аутентифікації

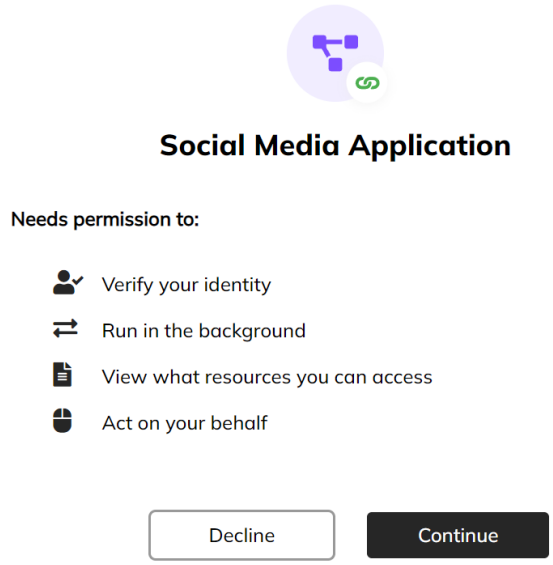

Рисунок 3.4 - Надання дозволу через інтерфейс

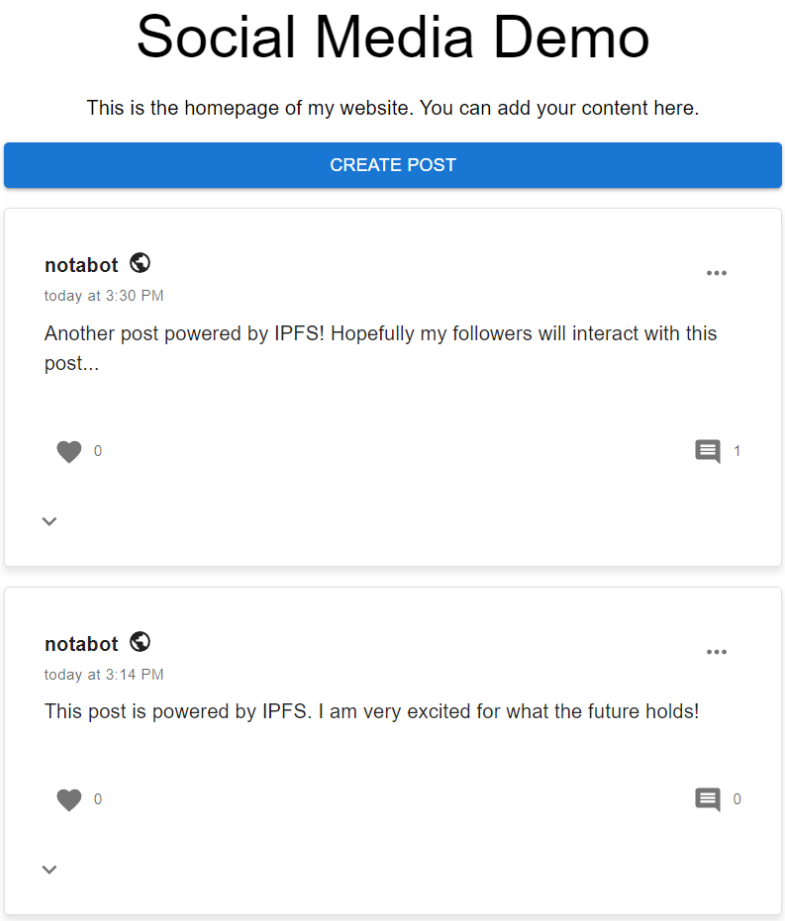

Рисунок 3.5 - Головна сторінка додатку соціальних медіа

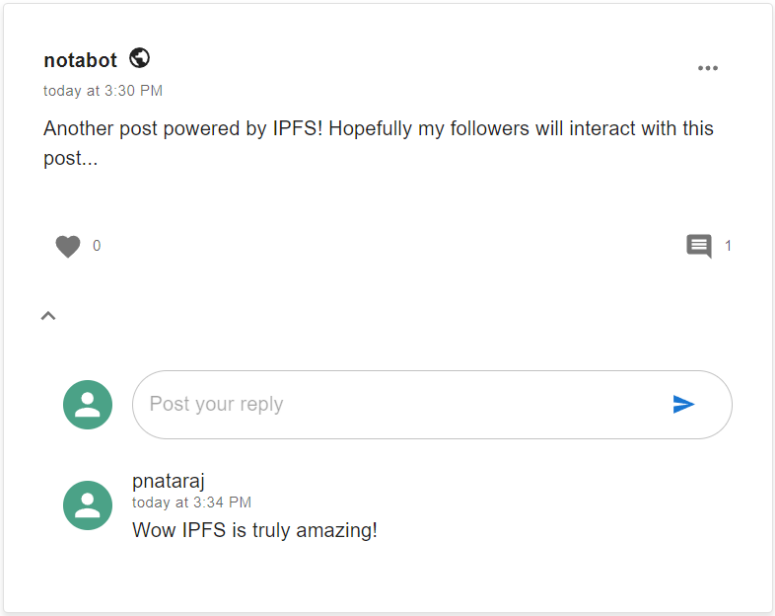

Рисунок 3.6 - Коментування публікації

Процес створення публічної публікації проілюстровано на рисунку 3.7, де показано створення посту з текстом "Hello, World!". Після публікації, цей пост стає видимим на стрічці новин підписника, як відображено на рисунку 3.16. З цією публікацією також можлива взаємодія. Наприклад, як показано на рисунку 3.17, підписник може поставити позначку "подобається". Ця інтеракція (лайк) буде відображена для всіх інших переглядачів публікації після успішного оновлення вказівника IPNS на новий ресурс IPFS, що містить актуалізовані дані публікації. Користувачі також можуть залишати коментарі, як ілюструється на рисунку 3.6, що демонструє процес взаємодії користувача з публікацією шляхом додавання коментаря.

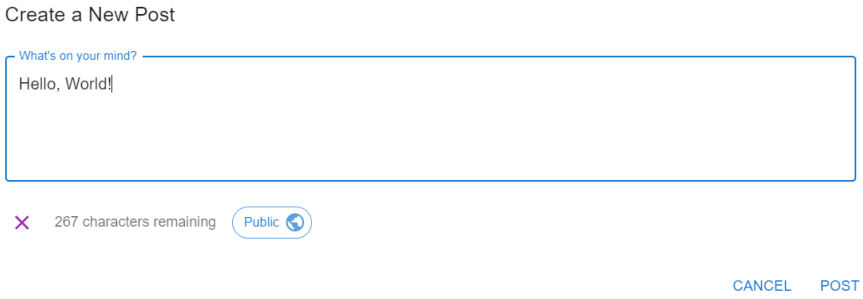

Рисунок 3.7 - Створення публічного поста (публікації)

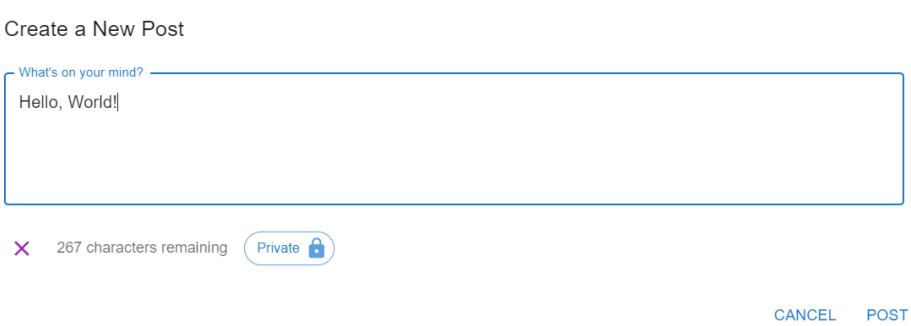

Рисунок 3.8 - Створення приватної публікації

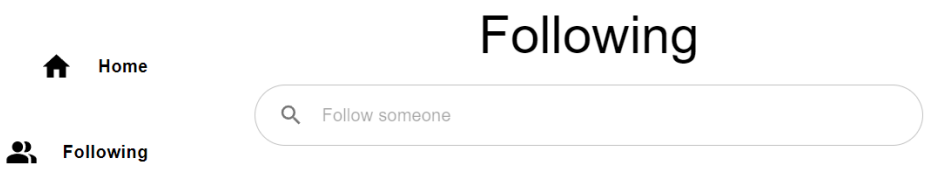

Рисунок 3.9 - Відсутність користувачів у списку підписок

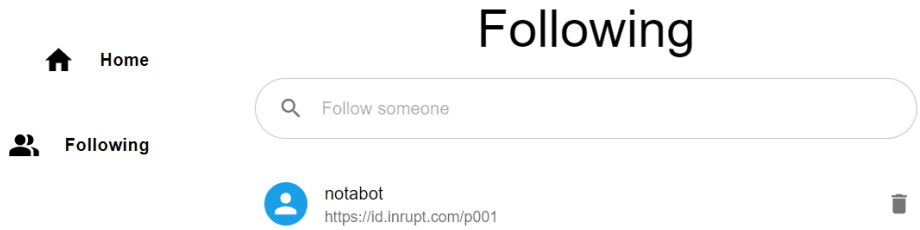

Рисунок 3.10 - Успішна підписка на користувача

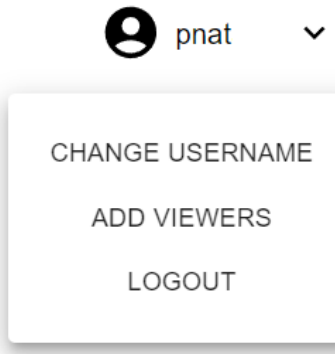

Рисунок 3.11 - Параметри користувача

Ще однією реалізованою функцією, що відповідає популярним практикам сучасних платформ соціальних медіа, є можливість користувачів змінювати свої імена користувачів (нікнейми) (див. рисунок 3.12). Імплементація цієї функції зумовлена тим, що імена користувачів часто виступають як персоналізовані ідентифікатори. Крім того, реалізована інтеграція забезпечує зв'язок WebID користувачів з їхніми обраними іменами користувачів, що дозволяє користувачам за необхідності верифікувати справжні ідентичності інших суб'єктів.

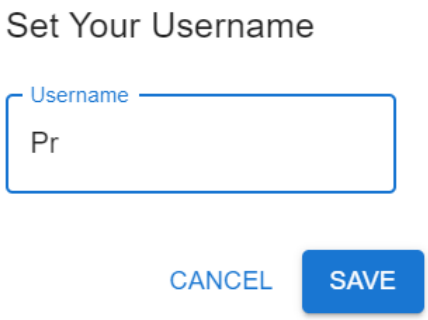

Рисунок 3.12 - Зміна імені користувача

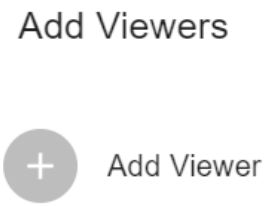

Рисунок 3.13 - Список переглядачів порожній

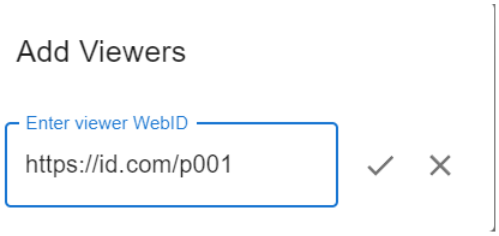

Рисунок 3.14 - Додавання переглядача

У контексті приватних публікацій, видавець спершу повинен визначити WebID користувачів, які матимуть право доступу до публікації. Цей процес ілюструється на рисунку 3.15. Доступ до приватних публікацій надається лише користувачам, які мають відповідний дозвіл, встановлений видавцем через механізми Solid ACL. Дозвіл на доступ може бути відкликаний у будь-який час, що забезпечує видавцю повний контроль над його приватними даними.

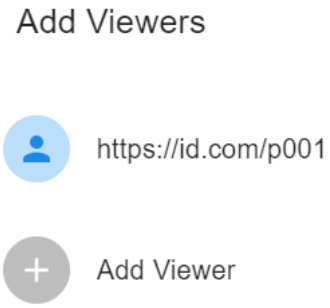

Рисунок 3.15 - Додані переглядачі

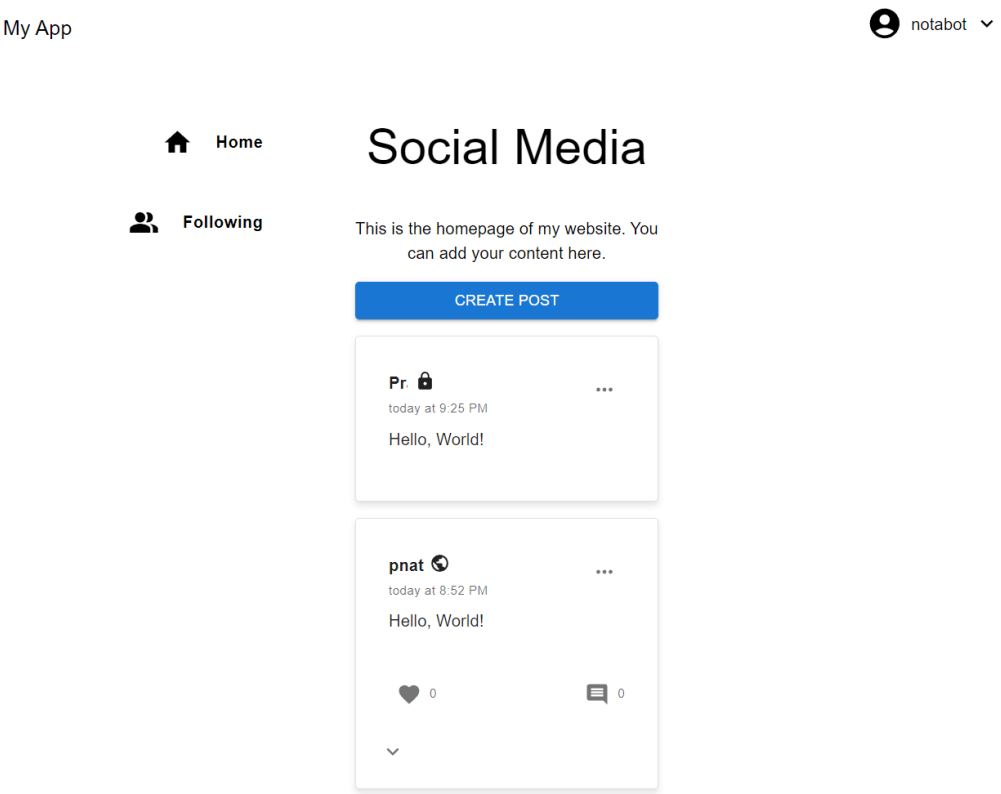

Рисунок 3.16 - Приватні та публічні публікації на головній сторінці

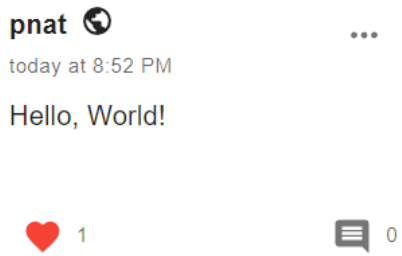

Рисунок 3.17 - Публікація, відзначена користувачем як "подобається".

Однак, слід зазначити потенційну особливість, пов'язану з управлінням даними інтеракцій (таких як коментарі) у контексті приватних публікацій. Якщо дані коментарів повністю зберігаються у Solid Pod видавця, це може створити ситуацію, коли коментарі користувачів потенційно можуть бути змінені або видалені власником Поду, що певним чином суперечить принципам володіння даними коментаторів.

3.6. Представлення обмежень застосування Solid та IPFS у децентралізованому соціальному нетворкінгу

Використання Solid та файлової системи IPFS пропонує численні переваги для підвищення безпеки та надання автономії користувачеві у децентралізованих системах. Проте, існують певні значні виклики та обмеження, пов'язані з їхньою практичною імплементацією та потенційним широким застосуванням.

Одним з головних викликів є проблема масштабованості. Хоча обидві технології пропонують перспективні рішення для децентралізації та володіння даними, вони можуть стикатися з труднощами при обробці значних обсягів трафіку та підтримці масштабного впровадження. IPFS, зокрема, спирається на мережу вузлів для хостингу та розповсюдження контенту, що може призвести до появи вузьких місць продуктивності та збільшення часу отримання контенту зі зростанням мережі.

Крім того, забезпечення цілісності та постійного зберігання даних є вагомим викликом. Хоча IPFS надає розподілене рішення для зберігання, завжди існує ризик втрати даних, якщо достатня кількість вузлів, що зберігають контент, відключаються або не забезпечують ефективну реплікацію даних (наприклад, через відсутність піннінгу). Аналогічно, Solid спирається на принципи пов'язаних даних та децентралізовані ідентифікатори, такі як WebID, для підтримки володіння даними та управління доступом, проте забезпечення надійності та постійної доступності цих децентралізованих ідентифікаторів може бути складним завданням.

Значний виклик у використанні IPFS, особливо в контексті підтримки змінних даних (як-от інтеракції у соціальних медіа), пов'язаний із застосуванням розподілених хеш-таблиць (DHT) в IPNS (InterPlanetary Naming System). Оскільки IPNS використовує DHT для зберігання записів, контент, адресований через нього (тобто, запис IPNS, що вказує на ресурс IPFS), потребує регулярного повторного публікування для забезпечення доступності цього запису. Це вимагає постійної активності вузла, який хостить інформацію IPNS (як правило, сервера піннінгу або іншого спеціалізованого сервісу), для безперервного оновлення записів у DHT. Інакше посилання на публічні публікації можуть стати неактивними протягом короткого періоду, що є неприйнятним для підтримки життєздатності платформи соціальних медіа, яка очікує постійної доступності контенту.

Іншою ключовою проблемою є крива навчання та складність, асоційовані з впровадженням та прийняттям Solid та IPFS. Обидві технології вводять нові парадигми для зберігання даних, доступу та аутентифікації, що може вимагати від розробників та кінцевих користувачів освоєння незнайомих концепцій та інструментарію. Це може слугувати перешкодою для широкого прийняття, особливо серед користувачів з низьким рівнем

технічної грамотності, для яких навігація у децентралізованих системах та управління власними Подами може здаватися складною. Зі збільшенням кількості додатків, що використовують Solid, користувачі стикатимуться зі збільшенням кількості запитів на доступ до даних, що може бути заплутаним, особливо при використанні більш складних додатків, які потребують множинних дозволів.

Крім того, існують занепокоєння щодо потенційного нецільового використання або зловживання децентралізованими технологіями, такими як IPFS та Solid. Децентралізація ускладнює регулювання та реагування на протиправні дії, такі як розміщення незаконного контенту або здійснення нелегальних транзакцій. Це породжує юридичні та етичні питання щодо відповідальності розробників/операторів платформ та ролі управління (governance) у децентралізованих екосистемах. Проте, були запропоновані потенційні рішення, такі як BlockIPFS, які можуть бути застосовані у контексті соціальних медіа для вдосконалення модерації та ідентифікації джерела протиправної діяльності. Слід також зазначити, що, незважаючи на ці виклики, децентралізація обох технологій також може слугувати перевагою, оскільки розподілене зберігання файлів у IPFS на множині вузлів підвищує стійкість даних до втрати під час часткових збоїв мережі.

Підсумовуючи, хоча Solid та IPFS пропонують вагомі рішення для підвищення безпеки та автономії користувача, вони також ставлять низку викликів, які потребують вирішення для повної реалізації їхнього потенціалу та забезпечення широкого прийняття. Подолання проблем масштабованості, забезпечення надійної цілісності та постійного зберігання даних, спрощення процесу адаптації користувачів та розробників, а також вирішення питань, пов'язаних із потенційним зловживанням, є критичними факторами для подальшого розвитку та широкого прийняття цих технологій у контексті децентралізованих соціальних медіа.

3.7. Потенційні застосування IPFS та перспективи децентралізованого вебу

Протокол IPFS має різноманітні потенційні застосування у різних сферах. Однією з таких областей є сфера мереж доставки контенту (CDN). Традиційно CDN спираються на централізовані сервери для доставки контенту кінцевим користувачам. Однак така централізована архітектура може призводити до виникнення вузьких місць продуктивності та єдиних точок відмови. Використання IPFS дозволяє децентралізувати доставку контенту в CDN, забезпечуючи, що популярний контент буде легкодоступним з множини вузлів, розподілених по мережі.

Більш практичним прикладом цього застосування є його використання в контексті соціальних медіа. IPFS може ефективно застосовуватися для отримання та публікації публікацій. Корисність IPFS у цьому сценарії посилюється завдяки наявності великої кількості вузлів у мережі. Таким чином, "вірусні" публікації не спричинятимуть значних перевантажень центральних серверів, оскільки зі збільшенням кількості вузлів, що отримують копії такого контенту, зростає кількість джерел його дистрибуції. Це дозволяє платформі масштабуватися для обслуговування будь-якої кількості користувачів, що звертаються до даних, роблячи отримання публікацій ще швидшим, чим більше людей запитують конкретну публікацію користувача. Масштабованість IPFS, особливо щодо операцій читання даних, є чудовим інструментом для додатків, де кількість користувачів, що отримують дані, значно перевищує кількість тих, хто їх публікує, що є типовим для соціальних медіа. Таким чином, пріоритет зміщується на швидкість отримання контенту, а не швидкість його публікації.

IPFS також може бути ефективно використаний у контексті обміну файлами та однорангових (peer-to-peer) мереж. Платформи для обміну файлами часто стикаються з проблемами, такими як низька швидкість

завантаження та залежність від центральних серверів або трекерів. За допомогою IPFS файли розподіляються по децентралізованій мережі вузлів, що дозволяє здійснювати швидший та стійкіший обмін файлами, подібно до принципу роботи торрентів, але без необхідності використання централізованих трекерів для координації передачі файлів.

Крім того, IPFS може знайти своє застосування у галузі наукового дослідження та обміну даними. Дослідникам часто необхідно обмінюватися великими наборами даних з колегами по всьому світу. Використовуючи IPFS, дослідники можуть розподіляти набори даних децентралізованим способом, забезпечуючи їхню доступність та надлишковість, одночасно зменшуючи залежність від централізованих серверів сховищ. У цьому контексті також може бути інтегрований протокол Solid. Наприклад, при обміні медичними даними користувача з його Поду, учасники можуть легко управляти доступом для будь-яких дослідницьких організацій, які потребують цих даних, використовуючи механізми Solid ACL. Крім того, дані можуть бути анонімізовані для забезпечення повної конфіденційності учасників дослідження.

Протоколи Solid та IPFS пропонують перспективний погляд на майбутнє мережі Інтернет, яке пріоритезує безпеку та контроль користувача над його даними. Хоча виклики, пов'язані з масштабованістю, забезпеченням постійного зберігання даних, процесом прийняття користувачами та потенційним зловживанням, потребують подальшого дослідження та вирішення, потенційні переваги є значними. Solid надає користувачам безпрецедентний контроль над їхніми даними. Користувачі зберігають свою інформацію у власних Подах та надають додаткам доступ на основі гранулярних дозволів, таким чином стаючи повноправними розпорядниками свого цифрового сліду. IPFS, з іншого боку, надає надійне та децентралізоване рішення для зберігання контенту. Контент адресується за його унікальним хешем (CID), що забезпечує цілісність даних та стійкість до

цензури. Працюючи над подоланням існуючих перешкод та використовуючи сильні сторони як Solid, так і IPFS, ми можемо прокласти шлях до більш безпечного, прозорого та орієнтованого на користувача онлайн-досвіду.

Таким чином, комбінація Solid та IPFS представляє перспективну архітектурну основу для побудови децентралізованих додатків соціальних медіа, які пріоритезують конфіденційність користувача, володіння даними та стійкість до цензури. За умови подальших інновацій та розвитку технологій, а також активної співпраці в цій галузі, децентралізовані соціальні медіа мають потенціал здійснити трансформацію цифрового ландшафту, надаючи користувачам розширені можливості та сприяючи формуванню більш відкритого та справедливого онлайн-середовища.

Подальші дослідження та розвиток цього децентралізованого додатку соціальних медіа можуть бути зосереджені на низці ключових напрямків. Майбутні покращення можуть включати:

- Розробка системи модерації.

Необхідне створення ефективної системи модерації для запобігання етичним проблемам та зловживанням, що можуть виникнути на повністю нерегульованій децентралізованій платформі. Одним з можливих підходів є імплементація системи репутації.

- Функціонал спільнот та рекомендацій.

Розширення функціоналу шляхом створення можливостей для об'єднання користувачів у спільноти та розробка платформи для заохочення взаємних підписок. Важливим завданням є створення рекомендаційної системи для сприяння встановленню зв'язків між користувачами.

- Реалізація функцій реального часу.

Впровадження технологій, таких як Web Sockets, для забезпечення отримання оновлень публікацій у реальному часі, а також реалізація системи сповіщень та справжньої системи запитів на "дружбу" для підвищення інтерактивності та повноти додатку.

- Оптимізація продуктивності та масштабованості.

У більш технічних аспектах, критично важливим завданням є дослідження та реалізація оптимізацій продуктивності та масштабованості додатку, особливо при обробці великих обсягів контенту, створеного користувачами, та інтеракцій. Це може включати дослідження методів для ефективного індексування контенту, стратегій кешування та оптимізації однорангової (peer-to-peer) комунікації в мережі IPFS.

- Сприяння прийняттю та обізнаності.

Зусилля, спрямовані на просування прийняття та підвищення обізнаності про децентралізовані технології, такі як Solid та IPFS, серед як розробників, так і кінцевих користувачів, є критично важливими для розвитку екосистеми. Ініціативи з просування, розробка освітніх ресурсів та співпраця з іншими проєктами в екосистемі децентралізованого Вебу можуть допомогти створити життєздатну спільноту та екосистему навколо децентралізованих додатків соціальних медіа.

ВИСНОВКИ

В дипломній роботі було продемонстровано потенціал децентралізованої платформи соціальних медіа, реалізованої з використанням протоколів Solid та IPFS.

У процесі виконання дослідження було здійснено аналіз сучасних концепцій децентралізованих обчислень у контексті соціального нетворкінгу, що дозволило сформувати цілісне уявлення про можливості, переваги та обмеження впровадження децентралізованих технологій у сфері соціальних медіа.

Перший розділ роботи був присвячений аналізу особливостей побудови децентралізованих соціальних мереж. Було встановлено, що традиційні централізовані соціальні мережі мають низку фундаментальних обмежень, серед яких домінують централізоване зберігання даних, порушення конфіденційності, ризики витоків інформації та обмежена користувацька автономія. У цьому контексті особливу увагу було приділено технологіям Solid та IPFS — як ключовим інструментам побудови децентралізованих систем. Інтеграція Solid, що забезпечує персоналізоване керування даними користувачів, з IPFS як розподіленою файловою системою, створює технічну основу для побудови альтернативної моделі соціального нетворкінгу з акцентом на приватність, масштабованість та довготривале збереження інформації.

У другому розділі було розглянуто архітектурні підходи до переходу від централізованих до децентралізованих рішень у соціальних мережах. Аналіз показав, що блокчейн, Solid і IPFS пропонують принципово нові підходи до зберігання та обміну інформацією, усуваючи потребу в єдиному централізованому контролері. Окрему увагу було приділено перевагам IPFS — таким як мінімізація дублювання даних, забезпечення стійкості до цензури та ефективна адресація контенту за його хешем. Також було розглянуто

практичне застосування цих технологій у контексті боротьби з дезінформацією, зокрема через розробку концепції виявлення неправдивих відгуків на основі блокчейн-технологій у поєднанні з IPFS. Проєкт Solid було ідентифіковано як перспективну парадигму управління персональними даними, що дозволяє користувачам зберігати контроль над своєю інформацією без посередництва платформ.

Третій розділ дослідження містив опис програмної реалізації розробленої системи соціального нетворкінгу з використанням Solid і IPFS. Практична імплементація Solid дозволила реалізувати модель розподіленого зберігання даних у власних контейнерах користувачів (PODs), що підвищує рівень безпеки та приватності. Інтеграція IPFS забезпечила ефективне зберігання великого обсягу мультимедійного контенту та можливість децентралізованого обміну ним. Комбіноване застосування цих технологій у рамках одного додатку продемонструвало високу функціональність, однак під час тестування були також виявлені технічні обмеження — такі як затримки при взаємодії з IPFS-мережею, складність автентифікації у Solid та обмежена підтримка сучасними браузерами. Попри це, система показала задовільну продуктивність у тестовому середовищі та підтвердила потенціал широкого використання.

Загалом, проведене дослідження дозволяє зробити висновок про значний потенціал децентралізованих технологій у трансформації сучасних моделей соціального нетворкінгу. Solid і IPFS не лише забезпечують вищий рівень приватності та безпеки, а й відкривають можливості для створення більш справедливих, прозорих і користувачоцентричних соціальних платформ. Подальший розвиток цих технологій, а також вирішення наявних технічних викликів, можуть стати ключем до формування децентралізованого вебу наступного покоління - Web 3.0 – з фундаментально новими принципами взаємодії між користувачами, платформами та даними.

В дипломній роботі показано, що користувачі розробленого додатку мають можливість створювати публікації та взаємодіяти з ними, а також здійснювати повний контроль над своєю цифровою присутністю в обліковому записі. Це досягається завдяки застосуванню протоколу Solid, який вимагає від користувачів надавати доступ до своїх даних через їхні персональні Поди (Pods), тим самим передаючи власність та контроль над даними від централізованої платформи до користувача

ПЕРЕЛІК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. M. Acosta, M.-E. Vidal, T. Lampo, J. Castillo, and E. Ruckhaus. ANAPSID: an adaptive query processing engine for SPARQL endpoints. In Proc. Int. Semantic Web Conf. (ISWC), pages 18–34, 2011.
2. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: Https://bitco.in/pdf/bitcoin.pdf
3. Smith, J. & Jones, A. "The Privacy Paradox in Centralized Social Networks". Journal of Online Behavior (2021): 45-62.
4. Chen, L. et al. "Data Handling Practices of Tech Giants: Concerns and Critique of Current Protections". Proc. of the Symposium on Internet Accountability (2022): 112-125.
5. Crosby, M.; Pattanayak, P.; Verma, S.; Kalyanaraman, V. Blockchain technology: Beyond bitcoin. Appl. Innov. 2016, 2, 71.
6. Vimitha R Vidhya Lakshmi and Gireesh Kumar T. "Mobile Social Networks: Architecture, Privacy, Security Issues and Solutions," Journal of Communications, vol. 12, no. 9, 524-531, 2017.
7. Davis, S. & Miller, P. "Federated Futures: An Analysis of Mastodon and the Decentralized Social Web". International Journal of Network Science 8.3 (2023): 187-201.
8. Berners-Lee, T. et al. "Solid: A Decentralized Web Platform for Data Ownership and Interoperability". Proceedings of the Decentralized Web Summit (2017): 1-10.
9. C Bommelaer de Leusse and Carl Gahnberg. 2019. The Global Internet Report: Consolidation in the Internet Economy. Internet Society (2019).
10. Benet, J. "IPFS - Content Addressed, Versioned, P2P Filesystem". arXiv preprint arXiv:1409.5180 (2014).

11. L. A. Cutillo, R. Molva, and M. ¨Onen. Safebook: A distributed privacy preserving online social network. In Proc. Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), pages 1–3, 2011.
12. White, R. & Black, K. "The Illusion of Deletion: Data Persistence in Cloud-Based Services". Journal of Data Security 15.1 (2020): 33-48.
13. Green, M. "Beyond the Trash Bin: Investigating Data Retention and Deletion Policies in Practice". Conference on Digital Rights and Privacy (2021): 78-91.
14. B. Dodson, I. Vo, T. J. Purtell, A. Cannon, and M. S. Lam. Musubi: Disintermediated interctive social feeds for mobile devices. In Proc. World Wide Web Conf. (WWW), pages 211–220, 2012.
15. GitHub - ipfs/helia: An implementation of IPFS in TypeScript - https://github.com/ipfs/helia
16. R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners Lee. Hypertext transfer protocol–HTTP/1.1, 1999.
17. Ushare Development Team. "Ushare: A Permissioned Blockchain Approach to Decentralized Content Hosting". Technical Report, Blockchain Lab (2023).
18. Brown, C. & Garcia, E. "Permissioned vs. Permissionless Blockchains: A Comparative Analysis". Blockchain Research Journal 4.2 (2022): 55-70.
19. Shrestha, A.K.; Vassileva, J. Blockchain-Based Research Data Sharing Framework for Incentivizing the Data Owners. In International Conference on Blockchain; Springer: Cham, Switzerland, 2018; pp. 259–266.
20. [10] Wilson, L. et al. "Addressing Mutable Data in IPFS: An IPNS and DHT-Based Solution for Decentralized Applications". Proc. of the Symposium on Distributed Systems (2023): 205-218.
21. Martinez, R. & Lee, H. "Scalability Challenges for Consumer dApps on the Ethereum Network". Cryptocurrency Economics Quarterly 7.4 (2024): 310-325.

22. Lu, Jian-Zhu, Xiuwei Fan, Jipeng Zhou, and Hao Yang. "An Improvement on an Efficient Mobile Authentication Scheme for Wireless Networks," Indonesian Journal of Electrical Engineering and Computer Science (IJEECS) 11, no. 10, pp. 6250-6257. 2013.
23. Benet, J. Ipfs-content addressed, versioned, p2p file system. arXiv 2014, arXiv:1407.3561.
24. Wood, G. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Proj. Yellow Pap. 2014, 151, 1–32.
25. Campbell, A. et al. "Evaluating IPFS Network Performance and Scalability Characteristics". Journal of Peer-to-Peer Networking 12.4 (2024): 401-418.
26. Fischer, E. & Kim, S. "Combating Vendor Lock-in through Decentralized Storage Solutions". Future of Cloud Computing Conference (2021): 180-195.
27. I. Hickson. The WebSocket API. Candidate recommendation, W3C, Sept. 2012. http://www.w3.org/TR/2012/CR-websockets-20120920/.
28. T. Inkster, H. Story, and B. Harbulot. WebID-TLS Specification. 2013. http://www.w3.org/2005/Incubator/webid/spec/tls/.
29. Helia Developers. "Helia: IPFS Implementation for JavaScript Environments". Online Documentation (2023). [Available: https://docs.helia.io/]
30. Mansour, A. V. Sambra, S. Hawke, M. Zereba, S. Capadisli, A. Ghanem, A. Aboulnaga, and T. Berners-Lee. A demonstration of the solid platform for social web applications. Demo at the World Wide Web Conf. (WWW), 2016.
31. Omar Abdullah Lajam and Tarek Ahmed Helmy. 2021. Performance Evaluation of IPFS in Private Networks. In 2021 4th International Conference on Data Storage and Data Engineering (Barcelona, Spain) (DSDE ’21). Association for Computing Machinery, New York, NY, USA, 77–84. https://doi.org/10.1145/3456146.3456159

32. Legrand, J. "The BitTorrent Mainline DHT: Protocol and Implementation". International Journal of Distributed Systems 10.1 (2019): 45-58.
33. Timm Böttger, Gianni Antichi, Eder L Fernandes, Roberto di Lallo, Marc Bruyere, Steve Uhlig, and Ignacio Castro. 2018. The elusive internet flattening: 10 years of IXP growth. RIPE 78 (2018).
34. Inrupt Documentation Team. "Solid Authentication Guide for Node.js Web Servers". Inrupt Developer Resources (2022). [Available: https://www.google.com/search?q=developer.inrupt.com]
35. Gupta, S. & Kumar, A. "Scalability Bottlenecks in IPFS: A Network Simulation Study". Distributed Systems Conference (2024): 288-301.
36. N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, and C. Mortimore. Openid connect core 1.0. The OpenID Foundation, 2014.
37. Wang, Y. et al. "Ensuring Data Integrity and Identity Reliability in Decentralized Ecosystems". Future Internet Journal 16.2 (2024): 150-165.
38. Hopwood, D.; Bowe, S.; Hornby, T.; Wilcox, N. Zcash Protocol Specification; GitHub: San Francisco, CA, USA, 2016.
39. Wu, A.; Zhang, Y.; Zheng, X.; Guo, R.; Zhao, Q.; Zheng, D. Efficient and privacy-preserving traceable attribute-based encryption in blockchain. Ann. Telecommun. 2019, 74, 401–411.

БІБЛІОГРАФІЧНА ДОВІДКА

Тема дипломної роботи: “Застосування методів децентралізованих обчислень в соціальному нетворкінгу ”

Обсяг пояснювальної записки: __77__ аркушів.

Дата закінчення роботи: 10 червня 2025 р.

Підпис студента _____