КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет комп'ютерних наук та кібернетики
Кафедра інтелектуальних програмних систем

Кваліфікаційна робота

на здобуття освітнього рівня бакалавра
за спеціальністю 121 Інженерія програмного забезпечення
на тему:

РОЗРОБКА API ДЛЯ СЕРВІСІВ З ВИКОРИСТАННЯМ БЛОКЧЕЙН, НА ПРИКЛАДІ ДЕЦЕНТРАЛІЗОВАНОЇ СИСТЕМИ ГОЛОСУВАННЯ

Виконав студент 4-го курсу
Кирило РЯБОВ

(підпис)

Науковий керівник:
доцент, кандидат фіз.-мат. наук
Максим ВЕРЕС

(підпис)

Засвідчую, що в цій роботі немає запозичень з праць інших авторів без відповідних посилань.

Студент

(підпис)

Роботу розглянуто й допущено до захисту на засіданні кафедри інтелектуальних програмних систем
«29» травня 2023 р.,
протокол № 11
Завідувач кафедри
Олександр ПРОВОТАР

(підпис)

Київ – 2023

РЕФЕРАТ

Обсяг роботи 48 сторінок, 9 ілюстрацій, 15 джерел посилань, 3 додатки.

ДЕЦЕНТРАЛІЗОВАНА АВТОНОМНА ОРГАНІЗАЦІЯ, ДЕЦЕНТРАЛІЗОВАНИЙ API, СИСТЕМА ОРГАНІЗАЦІЇ РОЛЕЙ, ПІДГРАФ, ETHEREUM, EVM-СУМІСНИЙ БЛОКЧЕЙН, SOLIDITY, WEB3

Об’єктом роботи є децентралізований API для сервісів на основі EVM-сумісного блокчейну. Предметом роботи є децентралізована автономна організація та API для неї.

Метою роботи є створення децентралізованої автономної організації та на її прикладі дослідження API для сервісів на EVM-сумісному блокчейні.

Методи розроблення: розробка програмного продукту з обчисленнями на блокчейні та поза ним. Інструменти розроблення: інтегроване середовище програмування та розробки JetBrains IntelliJ IDEA 2023.2, контрактно-орієнтована мова програмування Solidity, The Graph протокол для розробки API, AssemblyScript та TypeScript.

Результати роботи: виконано загальний огляд розробки сервісів на основі EVM-сумісному блокчейні, досліджено основні складнощі розробки таких сервісів, їх переваги та недоліки. Створено децентралізовану автономну організацію з єдиною точкою входу в систему та розроблено децентралізований API, який надає можливість отримувати дані, які складно дізнатися напряму з блокчейну.

Рекомендації щодо використання роботи: може застосовуватися в навчальному процесі для вивчення основної структури децентралізованої автономної організації та розробки API для неї за допомогою The Graph протоколу.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ 5

ВСТУП 6

РОЗДІЛ 1 ВСТУП ДО ETHEREUM, ДАО ТА THE GRAPH 9

1.1 Основи Ethereum 9

1.1.1 Поняття про Ethereum блокчейн 11

1.1.2 Значущість Ethereum 12

1.1.3 Переваги та недоліки централізованих рішень 13

1.2 Основи смарт-контрактів 15

1.2.1 Поняття про смарт-контракт 15

1.2.2 Необхідність та приклади використання смарт-контрактів 16

1.2.3 Проблеми та обмеження смарт-контрактів 17

1.3 Вступ до децентралізованих автономних організацій (ДАО) 19

1.3.1 Основи ДАО. Проблематика 19

1.3.2 Приклади сучасних ДАО 21

1.4 Розгляд протоколу The Graph 22

1.4.1 Поняття про The Graph 22

1.4.2 Мета та можливості The Graph 23

РОЗДІЛ 2 РОЛЬ THE GRAPH API В ДАО ТА GOVERNANCE HACKS 25

2.1 The Graph API в ДАО 25

2.1.1 Приклади синергії ДАО та API: Uniswap, Aave та Decentraland 26

2.3 Хакерські атаки на ДАО 26

2.3.1 Наслідки та проблеми хакерських атак на ДАО 27

РОЗДІЛ 3 АРХІТЕКТУРА РОЗРОБЛЕНОГО ДАО 28
3.1 Основна парадигма створеного ДАО 29
3.1.1 Модуль відстеження експертів 30
3.1.2 Модуль зберігання параметрів 31
3.1.3 Модуль для зберігання цифрових активів учасників ДАО 32
3.1.4 Модуль управління ролями в системі ДАО 32
3.2 Масштабованість та адаптованість 33
3.2.1 Переваги Diamond Pattern (ERC 2535) 33
РОЗДІЛ 4 РОЗРОБКА ДЕЦЕНТРАЛІЗОВАНОГО API ДЛЯ ДАО 36
4.1 Архітектура DAO-API 36
4.1.1 Огляд DAO-API 36
4.1.2 GraphQL – значення та переваги 37
4.2 Архітектура Permission-API 38
4.2.1 Огляд Permission-API 38
4.2.2 Майбутнє та переваги даного API 40
ВИСНОВКИ 42
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ 44
ДОДАТОК А. ДІАГРАМА РОБОТИ DAO API 46
ДОДАТОК Б. ПРИКЛАД ЗАПИТУ ТА РЕЗУЛЬТАТУ З ДОПОМОГОЮ DAO API 47
ДОДАТОК В. ПРИКЛАД ЗАПИТУ ТА РЕЗУЛЬТАТУ З ДОПОМОГОЮ PERMISSION API 48

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

ДАО – децентралізована автономна організація;

API – Application Programming Interface, прикладний програмний інтерфейс;

DApps – Decentralized applications, децентралізовані додатки;

DeFi – Decentralized finance, децентралізовані фінанси;

EVM – Ethereum Virtual Machine, віртуальна машина Ethereum;

ICO – Initial coin offering, первинна пропозиція монет;

IPFS – InterPlanetary File System, міжпланетна файлова система;

NFT – Non-fungible token, невзаємозамінний токен;

SBT – Soulbound Token, токен цифрової ідентифікації.

ВСТУП

Оцінка сучасного стану об’єкта розробки. Об’єктом роботи є децентралізований API для сервісів на основі EVM-сумісному блокчейні, що є новою та перспективною сферою діяльності. Технологія блокчейн, що стрімко розвивається, відкрила нові можливості для створення децентралізованих систем та сервісів. Однією з розповсюджених інновацій є концепція децентралізованої автономної організації (ДАО).

ДАО — це тип організації, представлений правилами, зафіксованими у вигляді комп'ютерної програми, яка є прозорою, не може бути зміненою, контролюється членами організації й не підпадає під вплив зовнішніх регуляторів. ДАО мають потенціал для революції у формуванні та управлінні організацій, забезпечуючи детерміновані та незмінні правила, які гарантують підзвітність та прозорість. Практична реалізація ДАО не є тривіальною проблемою, через новизну та специфічність засобів розробки.

ДАО з технічної точки зору найчастіше є сервісом розміщеним на EVM-сумісному блокчейні, з великою кількістю функціонала, що породжує проблему доступу до даних. Для подолання даної проблеми, ми можемо надати API. Розробка даного API для децентралізованих сервісів, являє собою унікальний перелік труднощів, всупереч наявності різноманітних бібліотек та фреймворків. Хоча ці інструменти можуть спростити процес отримання необхідної інформації з блокчейну, але вони не забезпечують правдивість отриманої інформації.

Однією з проблем в процесі розробки — є визначення місця для розміщення API. Централізовані хостингові рішення створюють значні ризики, оскільки вони є централізованими, що означає єдину точку відмови, вразливість до цензури та потенціал для компрометування інформації з боку централізованої сторони. Що докорінно суперечить принципам технології блокчейн: цілісності, безпеці та конфіденційності.

Для вирішення вище згаданих проблем необхідна розробка децентралізованого API. Такий API забезпечить зв'язок між користувачами й

блокчейном, дозволяючи їм ефективно і безпечно взаємодіяти з сервісом, та зменшити ризики пов'язані з централізованими рішеннями.

Актуальність роботи та підстави для її виконання. Актуальність цієї роботи підкреслюється висхідною потребою в децентралізації, прозорості та безпеці в цифровому світі, що в традиційних централізованих системах та організація часто ігнорується. Своєю чергою, технологія блокчейн і в окремих випадках, ДАО, мають рішення для багатьох з цих проблем.

Розробка децентралізованого API для сервісів на EVM-сумісному блокчейні може значно спростити процес взаємодії користувача з ДАО. А отже більша кількість людей почне використовувати децентралізовані додатки (DApps).

Розроблений API забезпечує ефективний спосіб отримання агрегованих даних з блокчейну, оминаючи необхідність використання власних ресурсів для індексації блокчейну для отримання даних.

Крім того, потенційне застосування цієї технології виходить за рамки ДАО. Вона може бути використана для створення широкого спектра децентралізованих додатків і сервісів на блокчейні, в таких галузях, як фінанси, ланцюги постачання, системи голосування тощо.

Мета й завдання роботи. Метою роботи є створення децентралізованої автономної організації та на її прикладі дослідження API для сервісів на EVM-сумісному блокчейні.

Основні напрямки роботи є:

- Розробка програмного продукту з обчисленнями на блокчейні та поза ним.
- Створення ДАО з єдиною точкою входу в систему.
- Розробка децентралізованого API для забезпечення ефективного доступу до даних.

Об’єкт, методи й засоби дослідження або розроблення. Об'єктом дослідження є децентралізований API для сервісів на основі EVM-сумісному блокчейні та ДАО.

Використані методи включають розробку програмного забезпечення з обчисленнями на блокчейні та поза ним, з використанням інтегрованого середовища програмування та розробки, такого як JetBrains IntelliJ IDEA 2023.2. Solidity - контрактно-орієнтованої мови програмування, протоколу The Graph для побудови API, а також AssemblyScript і TypeScript як засоби розробки.

Можливі сфери застосування. Результати цієї роботи можуть бути застосовані в декількох сферах. Перш за все, вони можуть слугувати чудовим освітнім інструментом для осіб та установ, які прагнуть зрозуміти структуру ДАО та розробити API для неї з використанням протоколу The Graph.

Крім того, в ширшому контексті, цей API може сприяти створенню нового покоління додатків і сервісів на основі блокчейну.

Наприклад, фінтех-компанії можуть використовувати його для створення децентралізованих фінансових продуктів. Він також може знайти застосування в управлінні ланцюгами постачання, системах голосування, зберіганні даних і платформах обміну, де прозорість, безпека і децентралізація мають першочергове значення.

РОЗДІЛ 1 ВСТУП ДО ETHEREUM, ДАО ТА THE GRAPH

У першому розділі закладено основу для розуміння тем, що розглядаються в наступних розділах. Ми розглянемо основи Ethereum, децентралізованих автономних організацій (ДАО) та протоколу The Graph. Ці технології слугують каркасом для основного предмета дослідження: децентралізованого інтерфейсу прикладного програмування (API), пристосованого для сервісів, побудованих на блокчейні, адаптованому до віртуальної машини Ethereum (EVM), з акцентом на ДАО. У цьому розділі висвітлюється природа цих технологій, їхні можливості, проблеми, які вони вирішують, а також їхні переваги та недоліки порівняно з централізованими альтернативами.

1.1 Основи Ethereum

Створена у 2015 році, Ethereum - це платформа з відкритим вихідним кодом, заснована на технології блокчейн, яка полегшує розробку та хостинг децентралізованих додатків (DApps). Її революційною особливістю є віртуальна машина Ethereum (EVM). Ця система, що є повною за Тюрінгом, дозволяє виконувати будь-які додатки, незалежно від мови програмування, за умови наявності необхідних ресурсів та пам'яті.

Блокчейн Ethereum функціонує як розподілена облікова система, що реєструє транзакції. Однак, на відміну від блокчейну Bitcoin, він дозволяє розміщувати та виконувати програми, так звані "смарт-контракти". Це автономні договори, умови виконання яких прописані в коді.

Смарт-контракти впорядковують, аутентифікують або забезпечують виконання контрактів без необхідності залучення третіх сторін. Вони автоматизують складні процедури, усуваючи потребу в посередниках і зменшуючи ризик людської помилки. Це сприяє підвищенню ефективності та

довіри, що особливо корисно в цифровому світі, де договірні відносини часто виходять за географічні межі.

Як говорив один із засновників проєкту Ethereum Gavin Wood: «Є багато цілей цього проєкту [Ethereum]; одна з ключових цілей — спростити процес здійснення транзакцій між особами, які погодилися на це, оскільки в іншому випадку вони не мали б можливості довіряти один одному.» [1, с.1].

Ethereum набув популярності як основоположна технологія, оскільки може сприяти піринговим транзакціям, позбавленим довірених посередників, таких як банки чи фінансові установи. Така децентралізація пропонує численні переваги, серед яких стійкість до цензури, прозорість, безпека і потенційно знижені витрати на комісію.

Проте, як і будь-яка технологія, Ethereum не позбавлений недоліків. Основним недоліком є масштабованість, оскільки мережа Ethereum наразі здатна обробляти лише обмежену кількість транзакцій в секунду. Через це можуть виникати перевантаження і підвищуватися комісії за транзакції, особливо в періоди високого попиту. Іншим потенційним недоліком є складність, пов'язана з розробкою смарт-контрактів, що вимагає високого рівня технічної експертизи та розуміння мови програмування Solidity чи Vyper.

Попри ці проблеми, потенціал Ethereum для революції в різних галузях є загальновизнаним. Його здатність замінити традиційні контракти, програмованими смарт-контрактами, здатна трансформувати такі сектори, як фінанси, ланцюги постачання, нерухомість та інші, роблячи процеси більш ефективними, прозорими та безпечними. Таке елементарне розуміння Ethereum відкриває шлях до глибшого обговорення децентралізованих автономних організацій (ДАО) та The Graph, які побудовані на платформі Ethereum і використовують її.

«Загалом, ми хочемо створити таку систему, щоб користувачі могли бути впевнені, що незалежно від того, з якими особами, системами чи організаціями

вони взаємодіють, вони можуть робити це з абсолютною впевненістю в можливих результатах і в тому, як ці результати можуть бути досягнуті.»[1, с.1].

1.1.1 Поняття про Ethereum блокчейн

Ethereum, децентралізована блокчейн-система з відкритим вихідним кодом, була вперше презентована наприкінці 2013 року Віталіком Бутеріним, відомим програмістом і співзасновником журналу Bitcoin Magazine. Офіційно запущена у 2015 році, Ethereum започаткувала нову цифрову парадигму, яка вийшла за рамки фінансових додатків, що підтримуються першою криптовалютою — біткоїном.

Блокчейн Ethereum має схожість з блокчейном Bitcoin, слугуючи загальним реєстром історії всіх транзакцій, де кожен вузол мережі зберігає її копію. Однак блокчейн Ethereum відрізняється своєю здатністю зберігати програмований комп'ютерний код, так звані смарт-контракти.

«Ethereum в цілому можна розглядати як машину станів, засновану на транзакціях: ми починаємо зі стану зародження і поступово виконуємо транзакції, щоб перетворити його в деякий поточний стан.»[1, с.2].

Смарт-контракт — це програма, яка виконуються тільки тоді, коли задовольняється заздалегідь визначений набір правил. Ці контракти слугують фундаментальними елементами для побудови додатків на Ethereum і лежать в основі децентралізованих автономних організацій (ДАО), децентралізованих фінансів (DeFi) та інших додатків.

Ethereum використовує власну криптовалюту Ефір (ETH), яка захищає систему і використовується для оплати комісій за транзакції та послуги в екосистемі Ethereum.

По суті, Ethereum - це програмований блокчейн, що дозволяє розробникам створювати власні операції різного ступеня складності. Ця універсальність стала каталізатором сплеску інновацій, що призвело до розробки тисяч децентралізованих додатків (DApps) на платформі Ethereum. Ці DApps

пропонують широкий спектр послуг, починаючи від ігор і закінчуючи цифровими колекціями, фінансовими послугами та не тільки.

Новаторська технологія Ethereum створила основу для нової ітерації Інтернету, яку часто називають "Web3" - Інтернету, де користувачі зберігають повний контроль, посередники зведені до мінімуму, а взаємодія ґрунтується на довірі та прозорості.

1.1.2 Значущість Ethereum

Багатогранна цінність Ethereum пронизує різні сфери, підкреслюючи його потенціал до трансформації.

Одним з найзначніших внесків Ethereum є створення платформи для децентралізованих додатків (DApps). Ці додатки працюють у мережі Ethereum, використовуючи її децентралізовану архітектуру для підвищення прозорості, протистояння цензурі та усунення потреби в центральному органі влади. Ця інновація змінила цифрову екосистему, сприяючи новій парадигмі розробки та розгортання додатків.

Іншою важливою інновацією, яку запровадив Ethereum, є концепція смарт-контрактів. Ці самодостатні контракти фіксують умови договору між сторонами в рядках коду, автоматизуючи транзакції без потреби в посередниках. Така автоматизація прискорює швидкість транзакцій, знижує витрати й виключає можливість маніпуляцій.

Ethereum також демократизував доступ до фінансів, спростивши проведення первинної пропозиції монет (ICO). Підтримка платформою створення токенів прискорила зростання кількості ICO, надавши компаніям унікальний механізм збору коштів, а інвесторам — доступ до раніше недоступних інвестиційних можливостей.

Гнучкість і розширюваність Ethereum привернули увагу активної спільноти розробників. Вона постійно впроваджує різноманітні інновації, розробляючи

безліч DApps, ДАО та інших децентралізованих конструкцій. Їхня активна участь робить значний внесок у постійний розвиток та успіх Ethereum.

На відміну від децентралізованого духу Ethereum, централізовані рішення, такі як традиційні бази даних і централізовані сервери, вже давно стали стандартом, особливо в контексті зберігання та управління даними. Однак поява технології блокчейн кинула виклик цій централізованій парадигмі.

Централізовані системи мають низку переваг, зокрема, швидкість транзакцій та обробки даних, оскільки за цими операціями стежить єдиний орган. Вони, як правило, більш зручні та доступні для нетехнічних користувачів, часто надають кращі користувацькі інтерфейси та клієнтську підтримку. Крім того, контроль з боку центрального органу може підвищити безпеку та запобігти зловмисній діяльності. Цей центральний орган також може легко оновлювати й модернізувати систему. Також саме завдяки централізованим сервісам, інтернет існує сьогодні в тому вигляді в якому він є. Про що йдеться в наступній цитаті.

«Централізація допомогла мільярдам людей приєднатися до Інтернету і створила стабільну, надійну інфраструктуру, на якій він живе. Водночас жменька централізованих організацій контролює значну частину Всесвітньої павутини, одноосібно розпоряджаючись тим, що дозволено, а що ні.»[2].

Однак поява Ethereum і подібних децентралізованих технологій підкреслила обмеження централізованих систем – про які ми згадаємо в наступному розділі – стимулюючи перехід до децентралізації.

1.1.3 Переваги та недоліки централізованих рішень

Протягом багатьох років централізовані рішення, включаючи традиційні бази даних і централізовані сервери, були стандартом, насамперед у контексті зберігання та управління даними. Однак поява технології блокчейн, уособленням якої є Ethereum, поставила під сумнів цю централізовану модель. У цьому розділі ми критично розглянемо переваги та недоліки централізованих систем, порівнюючи їх з децентралізованим духом блокчейну та Ethereum.

Централізовані системи мають кілька переваг:

- вони часто демонструють вищу ефективність з точки зору швидкості транзакцій і обробки даних, оскільки ці операції організовує єдиний орган.
- ці системи, як правило, більш зручні та зрозумілі, що робить їх більш доступними для нетехнічних користувачів. Вони також часто забезпечують кращий користувацький інтерфейс і підтримку клієнтів.
- нарешті, контроль з боку центрального органу влади може підвищити безпеку і запобігти зловмисній діяльності. Центральний орган також може легко оновлювати та модернізувати систему.

Однак централізовані системи мають і суттєві недоліки:

- вони мають єдину точку відмови; якщо центральний сервер виходить з ладу, вся система може стати недоступною.
- якщо центральний орган скомпрометований, безпека всієї системи опиняється під загрозою.
- таким системам бракує прозорості, оскільки центральний орган управляє даними й процесами, що потенційно може призвести до проблем з довірою.
- нарешті, централізовані системи схильні до цензури, оскільки центральний орган влади має повноваження регулювати інформацію в системі.

«Комерція в Інтернеті стала майже повністю покладатися на фінансові установи, які виступають довіреними третіми сторонами для обробки електронних платежів. Хоча система працює досить добре для більшості транзакцій, вона все ще страждає від недоліків, притаманних моделі, заснованій на довірі.»[3, с.1].

Зіставлення цих переваг і недоліків з децентралізованою моделлю Ethereum і технологією блокчейн показує, що, хоча централізовані системи мають певні переваги, вони також демонструють значні недоліки. Ethereum усуває ці недоліки, надаючи прозору, стійку до цензури платформу, позбавлену єдиної точки відмови, позиціонуючи її як переконливу альтернативу традиційним централізованим системам.

Проте, Ethereum і технологія блокчейн також стикаються з власними проблемами, такими як масштабованість і складність, які варто враховувати при їх використанні.

1.2 Основи смарт-контрактів

Смарт-контракти є фундаментальним компонентом технології блокчейн, що лежить в основі Ethereum та багатьох інших EVM-сумісних блокчейн-мереж. Вони дозволяють децентралізованим додаткам (DApps) та децентралізованим автономним організаціям (ДАО) функціонувати незалежно та прозоро. У цьому підрозділі ми розглянемо концепцію смарт-контрактів, механізми їхньої роботи та їхнє значення в екосистемі Ethereum.

1.2.1 Поняття про смарт-контракт

Смарт-контракт, в контексті технології блокчейн, — це самовиконуваний контракт, в якому умови угоди між сторонами безпосередньо записані в коді. Цей код зберігається в децентралізованій мережі блокчейн і є прозорим для всіх залучених сторін.

Ось таке визначення терміну смарт-контракт дає G. Wood, автор книги «Mastering Ethereum»: «… ми використовуємо термін "смарт-контракти" для позначення незмінних комп'ютерних програм, які детерміновано виконуються в контексті віртуальної машини Ethereum як частини мережевого протоколу Ethereum — тобто на децентралізованому світовому комп'ютері Ethereum.»[4, с.127].

Смарт-контракти автоматизують виконання угоди, гарантуючи, що вона виконується відповідно до заздалегідь визначених правил та умов. Після виконання умов контракт самостійно виконує узгоджені дії, які можуть включати передачу цифрових активів або запис інформації.

По суті, смарт-контракти — це програмовані транзакції, які не потребують довіри та здійснюються на блокчейні. Вони діють як своєрідний цифровий

посередник, усуваючи необхідність у сторонньому посереднику для нагляду або забезпечення виконання контракту.

Хоча смарт-контракти існують на різних блокчейн-платформах, вони найчастіше асоціюються з Ethereum, яка була першою блокчейн-мережею, що реалізувала мову Solidity, яка є повною за Тюрінгом, що дозволяє створювати складні та повнофункціональні смарт-контракти.

«Смарт-контракти зазвичай пишуться мовою високого рівня, наприклад, Solidity. Але для того, щоб їх можна було запустити, вони повинні бути скомпільовані в низькорівневий байт-код, який виконується в EVM.»[4, с.128].

Смарт-контракти Ethereum стали основою для широкого спектра додатків, починаючи від простих обмінів токенів і закінчуючи складними ДАО.

1.2.2 Необхідність та приклади використання смарт-контрактів

Смарт-контракти сприяють взаємодії без довіри, автоматизуючи виконання транзакцій на основі заздалегідь визначених умов. Це усуває потребу в посередниках і сприяє підвищенню довіри до системи. Вони також забезпечують прозорість і незмінність, оскільки код відкритий для всіх учасників системи блокчейн і незмінний після розміщення.

Необхідність в системі без участі посередників, ще наголошувалась засновником протоколу Bitcoin, Satoshi Nakamoto: «Що потрібно, так це електронна платіжна система, заснована на криптографічному доказі, а не на довірі, що дозволить будь-яким двом охочим сторонам здійснювати транзакції безпосередньо одна з одною без необхідності залучення довіреної третьої сторони. Транзакції, які неможливо скасувати з обчислювальної точки зору, захистять продавців від шахрайства, а для захисту покупців можна легко впровадити звичайні механізми тимчасового зберігання коштів на рахунку (routine escrow mechanisms).»[3, с.1].

Ефективність і швидкість смарт-контрактів перевершують звичайні контракти, оскільки вони виконуються автоматично при виконанні певних умов.

Цьому сприяє віртуальна машина Ethereum (EVM), яка забезпечує середовище виконання смарт-контрактів. Крім того, усуваючи посередників, смарт-контракти також забезпечують економічну ефективність, що особливо актуально для транскордонних транзакцій.

Практичне застосування смарт-контрактів є різноманітним і трансформаційним. Вони є невіддільною частиною роботи платформ децентралізованих фінансів (DeFi), уможливлюючи такі функції, як кредитування, запозичення, стейкінг і децентралізовані обміни. Вони керуються децентралізованими автономними організаціями (ДАО), сприяючи демократичному прийняттю рішень і роботі без централізованої влади.

Смарт-контракти також дозволяють створювати токени, сприяючи демократизації доступу до фінансів та уможливлюючи нову форму збору коштів через первинну пропозицію монет (ICO). Вони підвищують прозорість та ефективність управління ланцюгами постачання, автоматизуючи процеси та незмінно фіксуючи транзакції. У секторі нерухомості смарт-контракти спрощують транзакції, автоматизуючи передачу прав власності та платежі. Вони також можуть бути використані для створення прозорих і захищених від підробки систем голосування, підвищуючи довіру та участь у виборчих процесах.

1.2.3 Проблеми та обмеження смарт-контрактів

Хоча смарт-контракти пропонують безліч переваг, вони не позбавлені проблем та обмежень. На Рис. 1.1 ці проблеми продемонстровані.

Рисунок 1.1 – Проблеми смарт-контрактів та їх наслідки.

Розберемо кожну вищезгадану проблему.

Безпека та вразливості: Смарт-контракти — фрагменти коду, і, як і будь-яке програмне забезпечення, вони схильні до помилок і вразливостей. Незмінна природа блокчейну означає, що після розміщення смарт-контракту його код не може бути змінений, що робить будь-які присутні помилки постійними. У минулому це призводило до значних інцидентів, пов'язаних з безпекою, таких як злом ДАО у 2016 році[5].

Масштабованість та плата за газ (комісії): Мережа Ethereum, як і багато інших блокчейн-платформ, стикається з проблемами масштабування. Кожна транзакція в смарт-контракті вимагає обчислювальних ресурсів, за які стягується плата, відома як газ. У міру того, як мережа стає більш перевантаженою, ці збори можуть стати непомірно високими. Це особливо актуально для додатків DeFi та ДАО, які часто використовують складні смарт-контракти й велику кількість транзакцій.

Дана проблема також згадується в роботі, де був описаний протокол Ethereum: «Масштабованість залишається вічною проблемою. З узагальненою функцією переходу стану стає важко розділити та розпаралелити транзакції, щоб застосувати стратегію "розділяй і володарюй". Без розв'язання цієї проблеми динамічний діапазон значень системи залишається по суті фіксованим, і зі збільшенням середньої вартості транзакцій менш цінні з них ігноруються, оскільки їх економічно безглуздо включати в облікову базу.»[1, с.17].

Нормативно-правова невизначеність: Правовий статус смарт-контрактів не є однозначно визначеним і може відрізнятися в різних юрисдикціях. Виконання умов смарт-контракту поза межами блокчейну або вирішення спорів може вимагати втручання традиційних правових систем, що додає ще один рівень складності.

Хоча смарт-контракти дозволяють здійснювати безперешкодні та прозорі транзакції, вони можуть бути складними та важкими для розуміння нетехнічними

користувачами. Це стосується як розуміння того, як працюють смарт-контракти, так і того, як з ними взаємодіяти.

Очікується, що постійні дослідження і розробки в таких сферах, як методи програмування смарт-контрактів, рішення для масштабування (наприклад, Ethereum 2.0 і Layer 2), а також правова і регуляторна ясність, з часом пом'якшать ці проблеми.

1.3 Вступ до децентралізованих автономних організацій (ДАО)

В останні роки децентралізовані автономні організації отримали визнання як суттєвий здобуток у сфері блокчейну. ДАО впроваджують новаторський підхід до структурування та управління онлайн-спільнотами, пропонуючи відкритий, прозорий та ефективний шлях до децентралізованого колективного прийняття рішень. Ми розглянемо концепцію ДАО, з'ясуємо їхні основоположні принципи, переваги, недоліки та невіддільну роль в екосистемі Ethereum.

1.3.1 Основи ДАО. Проблематика

По суті, ДАО — структура, яка полегшує координацію людської діяльності в цифровому просторі. Тому управління є найважливішим елементом ДАО. Хоча початкова ідея полягала в тому, що все, що робить ДАО, визначається виключно кодом, але незабаром ДАО прийшли до висновку, що це обмежує обсяг того, що вони можуть робити. Тож сьогодні багато ДАО вийшли за рамки простої логіки "код — це закон" і впроваджують процеси прийняття рішень, орієнтовані на потреби людини.

Однак цим процесам часто бракує витонченості; часто вони обмежуються простими голосуваннями власників токенів або схемами мультипідпису. Це по суті обмежує можливості й може призвести до вразливості до атак на управління.

Для того, щоб ДАО могли вийти за рамки поточного стану речей, необхідна більш досконала архітектура управління. Зокрема, ДАО потребують:

- здатність децентралізовано ухвалювати, а також оскаржувати прийняті рішення
- забезпечення дотримання "верховенства права" у відповідному ДАО
- механізми врегулювання для врегулювання конфліктів
- гарантії крипто економічної безпеки, які можна застосовувати до критично важливих рішень та виборів параметрів
- структурований процес зміни правил самої ДАО
- забезпечення дотримання прав власності.[6]

По суті, це функції управління, які в централізованому світі були б забезпечені правовими рамками, заснованими на державній владі. Однак сьогодні ці функції управління не доступні для ДАО в децентралізованому світі. А як ми знаємо, покладання на централізовані традиційні правові рамки не дуже добре працює у світі криптовалют.

Попри ці виклики, потенціал ДАО до революційних змін у різних секторах продовжує стимулювати їх прийняття та розвиток. Фундаментальні принципи ДАО являють собою відхід від традиційних централізованих процесів прийняття рішень, пропонуючи нову парадигму організаційного управління.

На Рис. 1.2 наведено спрощений приклад взаємодії користувача з ДАО та її значення.

Рисунок 1.2 – Взаємозв’язок користувача та ДАО.

1.3.2 Приклади сучасних ДАО

ДАО в Ethereum існують у різних формах, кожна з яких виконує різні функції. Вони варіюються від венчурних фондів до децентралізованих бірж і органів управління спільнотами.

Ось кілька найпомітніших прикладів:

MakerDAO: децентралізована кредитна платформа на Ethereum, яка підтримує Dai, стейблкоїн, прив'язаний до долара США. Управління MakerDAO здійснюється спільнотою за допомогою токена MKR. Власники токенів беруть участь у голосуванні щодо різних аспектів, включаючи параметри ризику.

Aragon: Ця платформа надає всеосяжну основу для створення та управління ДАО. Вона пропонує гнучке і середовище з можливістю конфігурації, що дозволяє будь-кому створити децентралізовану організацію на Ethereum і керувати нею.

Compound: Децентралізований протокол, який дозволяє надавати та позичати криптовалюту. Протокол Compound керується ДАО, а власники токенів COMP голосують за зміни в протоколі.

Ці приклади підкреслюють універсальність Ethereum для створення та управління ДАО. Вони також демонструють, як ДАО сприяють розширенню екосистеми Ethereum, стимулюючи інновації та полегшуючи децентралізоване управління.

Однак, управління ДАО виходить за рамки коду. Управління ДАО часто вимагає більш комплексного підходу, який може впоратися з непередбачуваними ситуаціями, які не можуть бути закладені в смарт-контракти. Це вимагає системи управління, яка забезпечує впевненість у технічно обов'язкових транзакціях, але також пропонує свободу дій, нюанси та підзвітність.

По суті, управління ДАО не повинно обмежуватися бінарною логікою смарт-контрактів, а має також включати людське судження та інтерпретацію[7]. Це особливо актуально в ситуаціях, що вимагають прийняття контекстно-

залежних рішень, які не можуть бути адекватно відображені логікою смарт-контрактів.

Отже, в той час, як Ethereum забезпечує надійну платформу для створення і роботи ДАО, управління цими організаціями вимагає збалансованого підходу, який поєднує в собі точність коду і гнучкість людського судження. Це забезпечить ефективне управління діяльністю ДАО та адаптацію до складної та динамічної природи людських взаємодій.

1.4 Розгляд протоколу The Graph

У сфері технології блокчейн можливість доступу до даних та управління ними має першорядне значення. The Graph, децентралізований протокол для індексації та запитів даних з блокчейнів, відіграє вирішальну роль у цьому контексті. Він призначений для ефективного пошуку даних на блокчейні, що стає все більш важливим, оскільки децентралізовані додатки (DApps) стають дедалі складнішими.

1.4.1 Поняття про The Graph

Граф слугує проміжною ланкою між блокчейн-додатками та даними, з блокчейну, які їм потрібні. Він дозволяє розробникам створювати та публікувати відкриті API, так звані підграфи, до яких додатки можуть звертатися за допомогою GraphQL. Це позбавляє розробників необхідності вручну витягувати дані з блокчейну, що може бути трудомістким і неефективним процесом.

Децентралізована природа The Graph гарантує, що дані залишаються відкритими та жодна сторона не контролює інформацію. Це відповідає принципам технології блокчейн, що сприяють прозорості та надійності роботи. Крім того, власний токен The Graph, GRT, використовується для заохочення індексаторів, кураторів і делегатів, які роблять внесок у протокол, сприяючи розвитку екосистеми, керованої спільнотою.

По суті, The Graph виступає життєво важливою інфраструктурою для блокчейн-додатків, дозволяючи їм функціонувати більш ефективно і результативно. Це ключовий компонент в постійному розвитку DApps і в розширені екосистеми блокчейну.

Станом на 2023 рік The Graph має глобальну спільноту, що включає понад 438 вузлів-індексаторів у мережі та понад 3 000 кураторів. Ця спільнота допомогла The Graph стати одним з найбільш використовуваних протоколів у блокчейн-індустрії.[8]

Таким чином, The Graph слугує децентралізованим протоколом для індексації та запитів даних з блокчейнів, подібно до того, як Google індексує вебсторінки. Це дозволяє запитувати дані, які важко отримати безпосередньо.

1.4.2 Мета та можливості The Graph

Мета The Graph - зробити дані на блокчейні легкодоступними, і вона досягається шляхом надання фреймворку для індексації даних з різних мереж, включаючи Ethereum, IPFS і PoA. Підтримка The Graph декількох мереж підвищує його універсальність, що робить його цінним інструментом для розробників, які працюють на різних блокчейн-платформах.

Функціональність The Graph поширюється на широкий спектр випадків використання, особливо у сфері децентралізованих додатків (DApps). Ці програми часто потребують доступу до складних наборів даних, що зберігаються на блокчейні. Граф полегшує цей доступ, дозволяючи DApps функціонувати ефективно і надійно.

Для ілюстрації розглянемо DApp, який додає дані в Ethereum за допомогою смарт-контракту. Цей контракт генерує одну або декілька подій, які підхоплюються вузлом графіку. Вузол сканує Ethereum на наявність нових блоків, кожен з яких може містити події для підграфа. Далі вузол використовує обробник «відображень», які перетворюють дані в сутності, див. Рис. 1.3. DApp може запитувати ці дані у графічного вузла, який отримує їх, перетворюючи запити

GraphQL у запити до свого базового сховища даних. Потім DApp може відображати ці дані в користувацькому інтерфейсі для кінцевих користувачів, які можуть створювати нові транзакції на Ethereum.

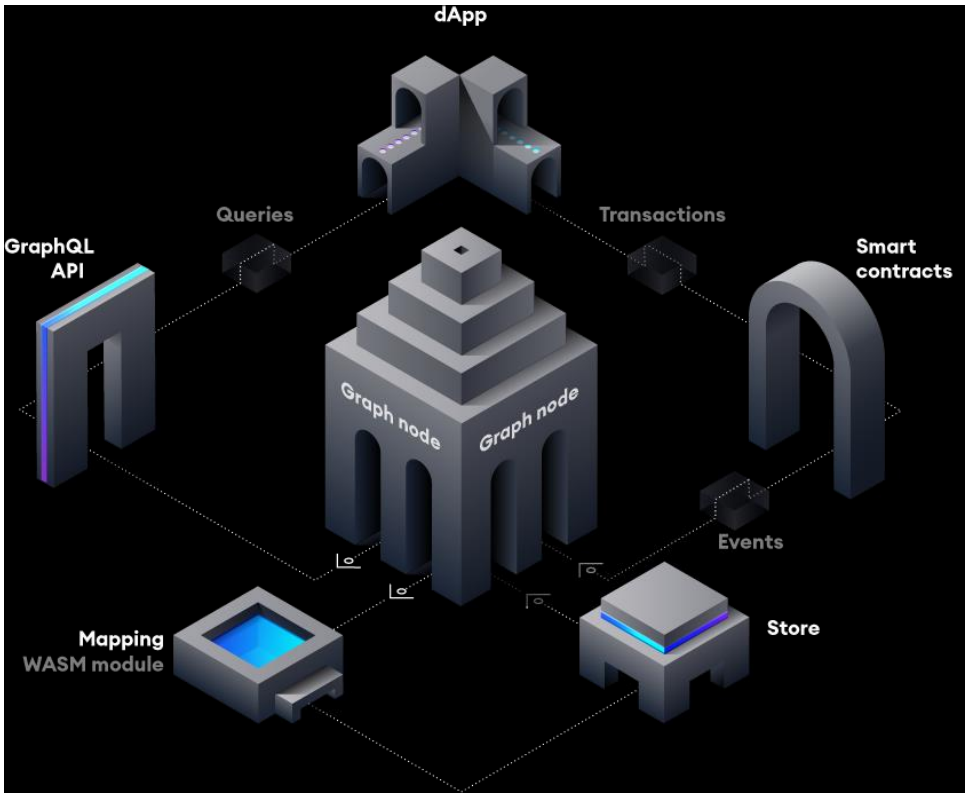

Рисунок 1.3 – Архітектура роботи Вузла графу[8]

По суті, The Graph служить життєво важливим мостом між даними блокчейну і додатками, спрощуючи доступ до даних і полегшуючи розробку і роботу DApps. Це важливий компонент екосистеми Ethereum, який стимулює інновації та сприяє децентралізованому управлінню.

Його можливість швидко індексувати події – головна його сила. Це стало можливим завдяки особливій реалізації заголовка блоку в Ethereum. Детально це питання розглядається у статті [9].

Важливо відзначити, що хоча The Graph є потужним інструментом, він не позбавлений своїх проблем. Як і у випадку з будь-якою технологією, забезпечення точності даних, підтримання безпеки та управління витратами є постійними проблемами.

РОЗДІЛ 2 РОЛЬ THE GRAPH API В ДАО ТА GOVERNANCE HACKS

The Graph API допомагає децентралізованим автономним організаціям (ДАО) здійснювати свою діяльність ефективніше. Здатність The Graph API отримувати та організовувати дані блокчейну у структурований та доступний спосіб покращує прозорість, безпечність та ефективність ДАО.

2.1 The Graph API в ДАО

ДАО працюють на блокчейні, що означає, що всі їхні дані зберігаються децентралізовано. Ці дані містять інформацію про учасників, пропозиції, голоси та будь-яку іншу необхідну інформацію. Зазвичай ці дані в сирому вигляді не надають значної користі. Саме тут і потрібен The Graph API.

The Graph API дозволяє ДАО запитувати ці дані у більш структурований спосіб, і проводити їх аналіз. Даний аналіз на самому блокчейні є неймовірно дорогим та обмеженим, оскільки з контексту EVM не можливо отримати доступ до історичних даних.

Наприклад, ДАО може потребувати інформації про те, скільки голосів отримала певна пропозиція. Використовуючи The Graph API, ДАО може легко запитувати ці дані та будувати складну статистику.

The Graph API є потужним інструментом для ДАО, що дозволяє їм запитувати та маніпулювати своїми даними так, як це було б складно або неможливо в інший спосіб. Оскільки популярність ДАО продовжує зростати, такі інструменти, як The Graph API, ставатимуть все більш важливими у світі децентралізованих організацій.

2.1.1 Приклади синергії ДАО та API: Uniswap, Aave та Decentraland

The Graph API відіграє важливу роль в роботі різних ДАО, включаючи Uniswap, Aave і Decentraland. Ці ДАО використовують The Graph API для ефективної обробки даних з блокчейну.

Uniswap: Широко визнана децентралізована біржа, використовує The Graph API для побудови маршрутів обміну двох токенів. Ця можливість дозволяє Uniswap надавати своїм користувачам інформацію про ціни в режимі реального часу, покращуючи задоволеність користувачів і загальну ефективність платформи.

Aave: Платформа децентралізованого кредитування, також використовує API-інтерфейси The Graph для аналізу загальної ситуації в системи. Це дозволяє Aave надавати актуальні кредитні ставки та іншу фінансову інформацію своїм користувачам, що сприяє підвищенню прозорості та надійності платформи.

Decentraland: Платформа віртуальної реальності, використовує The Graph API для швидкого оновлення даних. Це дозволяє Decentraland надавати в режимі реального часу інформацію про продаж віртуальної землі та інші ігрові події, підвищуючи інтерактивність платформи та залучення користувачів.

Ці тематичні дослідження ілюструють життєво важливу роль The Graph API в ДАО. Дозволяючи цим організаціям легко отримувати доступ до даних у мережі та використовувати їх, The Graph API сприяє швидкому функціонуванню ДАО та наданню актуальної й точної інформації їхнім учасникам.

2.3 Хакерські атаки на ДАО

Хакерські атаки на ДАО — це випадки, коли демократичним процесом прийняття рішень в ДАО маніпулюють або використовують, що часто призводить до негативних наслідків. Ці атаки можуть приймати різні форми, від використання вразливостей в смарт-контрактах до маніпулювання механізмами голосування.

2.3.1 Наслідки та проблеми хакерських атак на ДАО

Хакерські атаки на ДАО становлять значну проблему в децентралізованому світі. Зазвичай ці інциденти використовують вразливості в смарт-контрактах або механізмах управління ДАО, що призводить до маніпуляцій з результатами голосування або незаконного привласнення коштів. Відомим прикладом такої атаки був злам ДАО у 2016 році[5], коли вразливість в смарт-контракті ДАО була використана для витоку значної кількості ефіру.

Зовсім недавно мою увагу привернув злом “Tornado Governance Hack”[10].

Tornado Governance Hack, полягав у тому, що зловмисник висунув пропозицію, на виконання коду з його контракту. Пропозиція була затверджена голосуванням і прийнята, оскільки код контракту виглядав коректним та надійним. Однак потім хакер ліквідував контракт і розгорнув шкідливий за тією ж адресою. Цей другий контракт був виконаний через прийняту пропозицію, що призвело до того, що хакер отримав повний контроль над ДАО і забрав всі кошти, що зберігалися в ньому.

Такі інциденти підкреслюють важливість надійних заходів безпеки та ретельного нагляду в управлінні ДАО. Вони підкреслюють необхідність постійного моніторингу смарт-контрактів і механізмів управління для виявлення та усунення потенційних вразливостей. Крім того, вони підкреслюють необхідність всебічного розуміння коду, що лежить в основі операцій ДАО, особливо при голосуванні за пропозиції, які виконують код зовнішніх контрактів.

Ці атаки на ДАО мають значні наслідки для децентралізованого світу. Вони не лише призводять до фінансових втрат, але й підривають довіру до ДАО та екосистеми в цілому. Тому розв'язання цих проблем має першочергове значення для подальшого зростання та успіху ДАО.

РОЗДІЛ 3 АРХІТЕКТУРА РОЗРОБЛЕНОГО ДАО

Проаналізувавши кілька проєктів, реалізованих для екосистеми Ethereum, стає очевидним, що функціональний потенціал децентралізації є високим. Це особливо актуально для безлічі організацій і розробок, в першу чергу завдяки таким невіддільним властивостям технології блокчейн, як прозорість, доступність, конфіденційність і можливість запобігти цензурі даних.

Розглянемо, наприклад, систему, в якій право прийняття рішень належить не лише користувачам системи, але й групі експертів. Ці експерти слугують запобіжником, гарантуючи, що рішення приймаються у спосіб, який, найімовірніше, призведе до успіху. Їхній досвід дозволяє запобігти ситуаціям, подібним до тієї, що сталася під час "Tornado Governance Hack", адже вони можуть відмовити у виконанні тієї чи іншої пропозиції, наклавши вето.

Експерт у цьому контексті — це учасник системи, який продемонстрував свою компетентність у певній сфері, в основному в WEB3. Такий децентралізований підхід, який поєднує внесок усіх користувачів та експертні знання обраних, може забезпечити значні переваги для будь-якої організації.

Однак важливо зазначити, що хоча потенційні переваги є значними, впровадження такої системи вимагає ретельного планування. Виклики, з якими зіткнулися Governance Tornado і DAO Hack, слугують нагадуванням про потенційні ризики та складнощі, пов'язані з управлінням децентралізованими системами. Тому вкрай важливо винести уроки з цих інцидентів і вжити заходів для запобігання подібним ситуаціям у майбутньому.

Насамкінець, інтеграція технології блокчейн і залучення експертів, які приймають рішення, до децентралізованої системи може забезпечити численні переваги. Однак, важливо підходити до цього з обережністю, вчитися на минулих інцидентах, щоб забезпечити успішне впровадження та функціонування таких систем.

3.1 Основна парадигма створеного ДАО

Децентралізована автономна організація, яку я розробив, має унікальний набір функцій, включаючи інтеграцію експертів, єдину точку входу, масштабованість і сумісність з The Graph API. Такий дизайн дозволяє ДАО розпізнавати типи голосувань, які потребують залучення експертів. Ці експерти можуть брати участь у процесі голосування за необхідності, тим самим покращуючи механізм прийняття рішень.

Архітектура цієї системи складається з декількох модулів, кожен з яких виконує свою функцію:

- модуль зберігання параметрів (DAOParameterStorage): Цей модуль відповідає за зберігання параметрів системи.
- модуль відстеження експертів (DAOMemberStorage): Цей модуль керує експертами в системі.
- модуль зберігання цифрових активів (DAOVault): Цей модуль відповідає за зберігання цифрових активів учасників системи.
- модуль для голосування (DAOVoting): Цей модуль потрібний для проведення голосувань в ДАО.
- модуль управління ролями (PermissionManager): Цей модуль контролює управління ролями в системі.

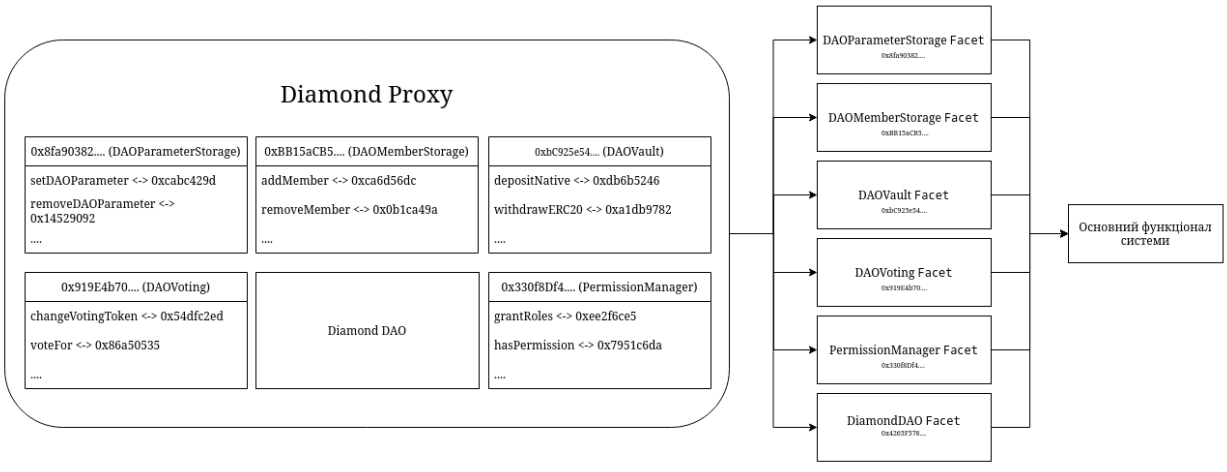

Рисунок 3.1 – Архітектура розробленого мною ДАО

Основною структурою системи є проксі (DiamondDAO), який діє як єдина точка входу в систему. Така структура забезпечує спрощену та ефективну взаємодію з платформою. На Рис. 3.1 продемонстровано архітектуру вищезгаданої системи.

Система ДАО розроблена таким чином, щоб бути надійною та гнучкою. Залучення експертів додає ще один рівень прийняття обґрунтованих рішень. Модульна архітектура забезпечує ефективне управління параметрами, експертними оцінками, цифровими активами та ролями. Проксі забезпечує єдину точку входу, що підвищує зручність використання системи, оскільки користувачу не потрібно робити запити окремо до кожного модуля в системі, як продемонстровано на Рис. 3.2.

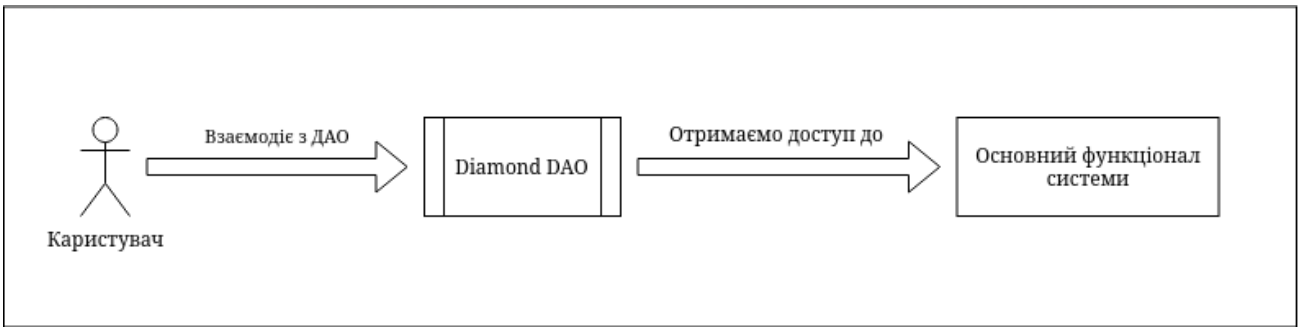

Рисунок 3.2 – Єдина точка входу в ДАО

3.1.1 Модуль відстеження експертів

Модуль відстеження експертів призначений для учасників, які прагнуть підвищити свій статус і стати експертами в ДАО та отримати право вето. Для цього вони повинні створити пропозицію на включення себе до списку експертів системи.

Пропозиція повинна містити вичерпну інформацію про учасника, включаючи його досягнення в галузі децентралізованих технологій, його професійний статус, базу знань та фізичне місце проживання. Потім ця інформація ретельно вивчається учасниками системи, щоб визначити чи

підходить кандидат для того, щоб бути експертом у даній сфері. Оцінка зосереджується на тому, чи має заявник необхідну кваліфікацію і, чи заслуговує він на роль експерта ДАО.

Після схвалення пропозиції заявника додають до списку експертів системи й призначають роль "Експерта".

Цей процес є невіддільною частиною підтримки цілісності та правдивості списку експертів у децентралізованій системі.

Майже кожна операція в системі, наприклад, додавання або видалення експерта, викликає подію: MemberAdded, MemberRemoved. Ці події мають велике значення для подальшого аналізу даних з використанням протоколу The Graph.

3.1.2 Модуль зберігання параметрів

Модуль зберігання параметрів відіграє ключову роль у роботі децентралізованої автономної системи, виступаючи її конституційною основою. Цей модуль відповідає за зберігання всіх параметрів, які безпосередньо впливають на процеси в системі. Наприклад, він регулює процес голосування за включення експертів, тривалість періоду вето на додавання параметрів системи й мінімальну кількість токенів, необхідну для того, щоб учасник міг ініціювати видалення експерта.

Ці параметри часто називають "параметрами ризику" через те, що вони можуть призвести до порушення функціональності системи, якщо їх неналежним чином модифікувати. Таким чином, система потребує присутності експертів, обраних спільнотою, для забезпечення її цілісності. Таким чином, даний модуль слугує конституцією ДАО, що забезпечує безперебійну та успішну роботу децентралізованої автономної системи.

3.1.3 Модуль для зберігання цифрових активів учасників ДАО

Екосистема децентралізованих автономних організацій спирається на активну участь її учасників, які інвестують в систему і, своєю чергою, мають право голосу в її управлінні. Такій участі сприяє спеціальний модуль, призначений для зберігання цифрових активів учасників ДАО. Цей модуль слугує вхідним шлюзом до ДАО, гарантуючи, що лише ті, хто інвестував у систему, можуть створювати пропозиції та впливати на напрямок її розвитку.

У моїй системі підтримуються чотири типи цифрових активів: Токени ERC20[11], ERC721[12], Soulbound (SBT) відповідно до стандарту ERC5484 та Ефір (ETH). Учасники можуть внести будь-який з цих токенів, щоб отримати роль учасника в ДАО. Однак для створення пропозицій їм потрібен базовий токен ДАО. Цей токен визначається творцем ДАО або спільнотою і може бути, наприклад, модифікованим токеном ERC721.

Цей модуль гарантує, що екосистема ДАО залишається захищеним середовищем, позбавленим потенційних зловживань. Він дозволяє створити справедливу і збалансовану систему, в якій тільки ті, хто має частку в ДАО, можуть впливати на її рішення.

3.1.4 Модуль управління ролями в системі ДАО

Модуль управління ролями є невіддільною частиною системи ДАО, забезпечуючи гнучкий і динамічний механізм управління ролями в рамках ДАО. Цей модуль базується на моделі управління доступом на основі ролей (Role-Based Access Control, RBAC)[13], який дозволяє призначати ролі та керувати ними відповідно до потреб та розвитку ДАО.

На початкових етапах розвитку ДАО може не потребувати широкого спектра ролей або експертів. Однак, у міру зростання та розвитку ДАО може виникнути потреба в більш спеціалізованих ролях та експертизі. Наприклад, ДАО може розширюватися і включати нові модулі, кожен з яких вимагає спеціальних

знань для ефективного управління. У таких випадках модуль управління ролями дозволяє додавати нові ролі та призначати їх відповідним експертам.

Крім того, модуль управління ролями також дозволяє проводити переоцінку та коригування ролей з плином часу. Така гнучкість має вирішальне значення, оскільки дає змогу ДАО адаптуватися до мінливих обставин і потреб. Наприклад, у міру зростання ДАО може виникнути потреба в наданні певним ролям права вето на певні пропозиції. Цим можна легко керувати за допомогою модуля управління ролями.

По суті, модуль управління ролями слугує гнучкою та адаптивною системою для управління ролями в рамках ДАО. Він підтримує динамічну природу ДАО, дозволяючи їй рости та розвиватися, зберігаючи при цьому ефективні механізми управління та прийняття рішень.

3.2 Масштабованість та адаптованість

Основою розробленої мною ДАО є інтеграція діамантового патерну (ERC 2535[14]), який слугує єдиною точкою входу в ДАО. Цей патерн є важливим для досягнення масштабованості системи. Діамантовий патерн — це по суті проксі, який зберігає сигнатури функцій та відповідні адреси імплементацій, де ці функції знаходяться. Поки сигнатура функції є унікальною в рамках системи, її можна додати до ДАО, забезпечуючи тим самим її масштабованість

3.2.1 Переваги Diamond Pattern (ERC 2535)

Масштабованість ДАО є вирішальним аспектом, який визначає її здатність рости й адаптуватися до світу Ethereum що постійно розвивається. ДАО, яку я розробив, використовує діамантовий патерн ERC 2535, що дозволяє (майже) необмежено розширювати його функціональність.

Diamond Pattern (ERC 2535) слугує ядром ДАО, полегшуючи додавання необмеженої кількості функціональних можливостей. Це особливо важливо, враховуючи обмеження на розмір контрактів, які можуть бути розміщені в

блокчейні Ethereum, що становить 24 кілобайти. Діамантовий патерн долає це обмеження, дозволяючи інтегрувати широкий спектр модулів, кожен з яких має окрему функціональність, але поділяє один і той самий стан. Ці та інші характеристики згадуються розробником патерну: «

1) Єдина адреса для необмеженої функціональності контракту. Використання єдиної адреси для функціональності контракту спрощує розгортання, тестування та інтеграцію з іншими смарт-контрактами, програмним забезпеченням та користувацькими інтерфейсами.

2) Ваш контракт перевищує максимальний розмір 24 КБ. Можливо, у вас є пов'язана функціональність, яку має сенс тримати в одному контракті або за однією адресою контракту. Діамант не має максимального розміру контракту.»[14].

Однією з ключових переваг такої масштабованості є можливість інтеграції з безліччю протоколів в екосистемі Ethereum. Наприклад, ДАО може інтегруватися з децентралізованими біржами, такими як Uniswap, протоколами кредитування та іншими популярними протоколами. Кожною інтеграцією можна керувати як окремим модулем всередині ДАО, що дозволяє гнучко впроваджувати правила та право голосу. Можливий перебіг інтеграції наведений на рис. 6.

Масштабованість ДАО також поширюється на структуру управління. Спільнота ДАО може визначати "параметри ризику" для кожної нової інтеграції та призначати експертів, які знаються на цих параметрах. Це гарантує, що ДАО зберігає контроль над новими інтегрованими модулями та може ефективно управляти будь-якими пов'язаними з ними проблемами.

Крім того, ДАО може визначати нові дозволи й додавати їх до теперішніх ролей. Це означає, що ДАО може впливати на нові додані модулі через процес управління. Спільнота також може вирішувати, хто має право накладати вето на пропозиції, що додає ще один рівень контролю та безпеки до функціонування ДАО.

На відміну від наявних систем, які можуть мати кілька точок входу, моя система ДАО має єдину точку входу, що підвищує її зручність та ефективність. Ця єдина точка входу також полегшує розробку підграфів відповідно до протоколу The Graph, роблячи зміни в системі більш доступними для користувачів, адже всі події доступні з однієї адреси.

Підсумовуючи, можна сказати, що розроблена мною ДАО є не тільки масштабованою, але й адаптованою. Вона може розширювати свою функціональність з часом, інтегруватися з різними протоколами Ethereum і коригувати свою структуру управління відповідно до змін у потребах. Вона створена для того, щоб рости та розвиватися з часом, надаючи більше функціональності кінцевим користувачам і забезпечуючи її актуальність в екосистемі блокчейну, що постійно розвивається.

РОЗДІЛ 4 РОЗРОБКА ДЕЦЕНТРАЛІЗОВАНОГО API ДЛЯ ДАО

Останній розділ мого дослідження присвячений розробці API для створеної мною децентралізованої автономної організації. Враховуючи складність ДАО, яка складається з численних модулів і величезної кількості збереженої інформації, дуже важливо надати користувачам ефективні засоби доступу та аналізу цих даних.

Розроблений мною ДАО оснащений модулями, які задовольняють широкий спектр потреб користувачів. Однак користувачі можуть потребувати додаткової функціональності, особливо коли йдеться про доступ до історичних даних для прийняття рішень. Щоб розв'язувати цю проблему, я реалізував API, які дозволяють користувачам взаємодіяти з ДАО та отримувати необхідну інформацію.

По суті, розробка API для ДАО не тільки розширює функціональність ДАО, але також сприяє його масштабованості та адаптивності, роблячи його більш надійною та зручною для користувача системою.

4.1 Архітектура DAO-API

Перший розроблений прикладний інтерфейс обслуговує основні модулі ДАО: модуль для зберігання цифрових активів учасників ДАО, модуль для відстеження експертів ДАО, модуль для зберігання параметрів у ДАО та модуль для відстеження створення пропозицій та активності користувачів.

4.1.1 Огляд DAO-API

За допомогою API ми можемо встановити тривалість участі експерта в ДАО та його участь у голосуваннях, таким чином оцінюючи його активність. Ці дані про активність можуть бути використані для винагороди експертів або

розширення їхніх прав, наприклад, надання їм повноважень керувати новим модулем, коли ДАО інтегрується з новим протоколом.

Крім того, API дозволяє відстежувати зміни параметрів ДАО та їх динаміку, що полегшує побудову статистики. Ці статистичні дані про параметри дозволять нам впевнено прогнозувати майбутнє і побудувати стандартну модель, в якій учасники ДАО почуватимуться комфортно. Наприклад, ми можемо визначити оптимальний період голосування на основі активності користувачів.

API також дозволяє відстежувати, хто з користувачів найбільш активний в ДАО, наприклад, за кількістю створених ним пропозицій, а також оцінювати якість цих пропозицій, тобто, чи були ці пропозиції прийняті, чи ні. На додаток, ми можемо отримати загальну інформацію про користувача в одному запиті, що позбавляє від необхідності робити кілька запитів до різних кінцевих точок в ДАО на блокчейні.

В додатку А наведено діаграму роботи DAO API на прикладі створення користувачем пропозиції на додавання нового Експерта в ДАО.

4.1.2 GraphQL – значення та переваги

Розроблений мною програмний інтерфейс слугує важливим компонентом у системі децентралізованої автономної організації. Цей інтерфейс, що є основою для головної сторінки організації, дозволяє запитувати всі необхідні параметри з підграфа.

The Graph API, який використовує GraphQL, є ключовою особливістю цієї системи. GraphQL - це мова запитів для API та середовище виконання для виконання цих запитів з наявними даними. Вона забезпечує повний і зрозумілий опис даних в API, дає клієнтам можливість запитувати саме те, що їм потрібно, і нічого більше, полегшує розвиток API з часом і надає потужні інструменти для розробників.

Гнучкість GraphQL дозволяє налаштовувати запити для отримання конкретної інформації. Це означає, що користувачі можуть адаптувати свої запити

до своїх точних вимог, забезпечуючи більш ефективний і результативний спосіб доступу до даних.

Крім того, The Graph API підтримує широкий спектр типів запитів, включаючи сортування, пагінацію, фільтрацію та повнотекстові пошукові запити. Це дозволяє користувачам сортувати дані за певними атрибутами, переглядати колекції, фільтрувати результати за кількома критеріями та здійснювати текстовий пошук у даних.

Ця можливість налаштування запитів покращує роботу користувачів, забезпечуючи більш точний і ефективний пошук даних. Вона також підтримує масштабованість та адаптивність системи ДАО, що робить її потужним інструментом для управління та взаємодії з даними організації.

Крім того, дизайн системи полегшує створення інформаційної панелі, використовуючи можливості GraphQL для забезпечення зручного інтерфейсу для взаємодії з даними ДАО.

Як приклад роботи даного API, в додатку Б продемонстровано запит на отримання інформації про баланси, токени користувачів в системі та пропозиції, в яких користувачі брали участь.

4.2 Архітектура Permission-API

Permission API, відіграє ключову роль у відстеженні ролей та дозволів у системі децентралізованої автономної організації (ДАО). Цей інтерфейс особливо важливий для доступу до даних, які складно або майже неможливо отримати безпосередньо з блокчейну, до прикладу, таких як ролі в ДАО.

4.2.1 Огляд Permission-API

Ролі в ДАО — це, по суті, абстракція. Вони можуть називатися по-різному, але мати однакові права доступу. Аналіз і зберігання цієї інформації на рівні блокчейну, зокрема на рівні смарт-контрактів, є дуже ресурсомістким. Саме тут API доводить свою цінність.

Слухаючи такі події, як додавання ролі або дозвіл на роль, ми можемо агрегувати всю інформацію і зберігати список користувачів, які мають однакову роль на підграфі. Ми також можемо створити запит, який в результаті поверне нам список користувачів з однаковими або схожими ролями.

Особливо це корисно при інтеграції нового модуля в систему, можливий перебіг даної інтеграції наведено на Рис. 4.1. Адже під час даного процесу створюється багато подій пов’язаних з ролями та дозволами.

«Проєкти зі складними смарт-контрактами, такі як Uniswap, та NFT-ініціативи, такі як Bored Ape Yacht Club, зберігають дані в блокчейні Ethereum, що робить дуже складним зчитування чогось, окрім базових даних, безпосередньо з блокчейну…. вам довелося б обробляти кожну подію передачі даних, зчитувати метадані з IPFS, використовуючи ідентифікатор токена та хеш IPFS, а потім агрегувати їх. Навіть на такі відносно прості запитання децентралізованому додатку (DApp), що працює в браузері, знадобилися б години або навіть дні, щоб отримати відповідь.»[9].

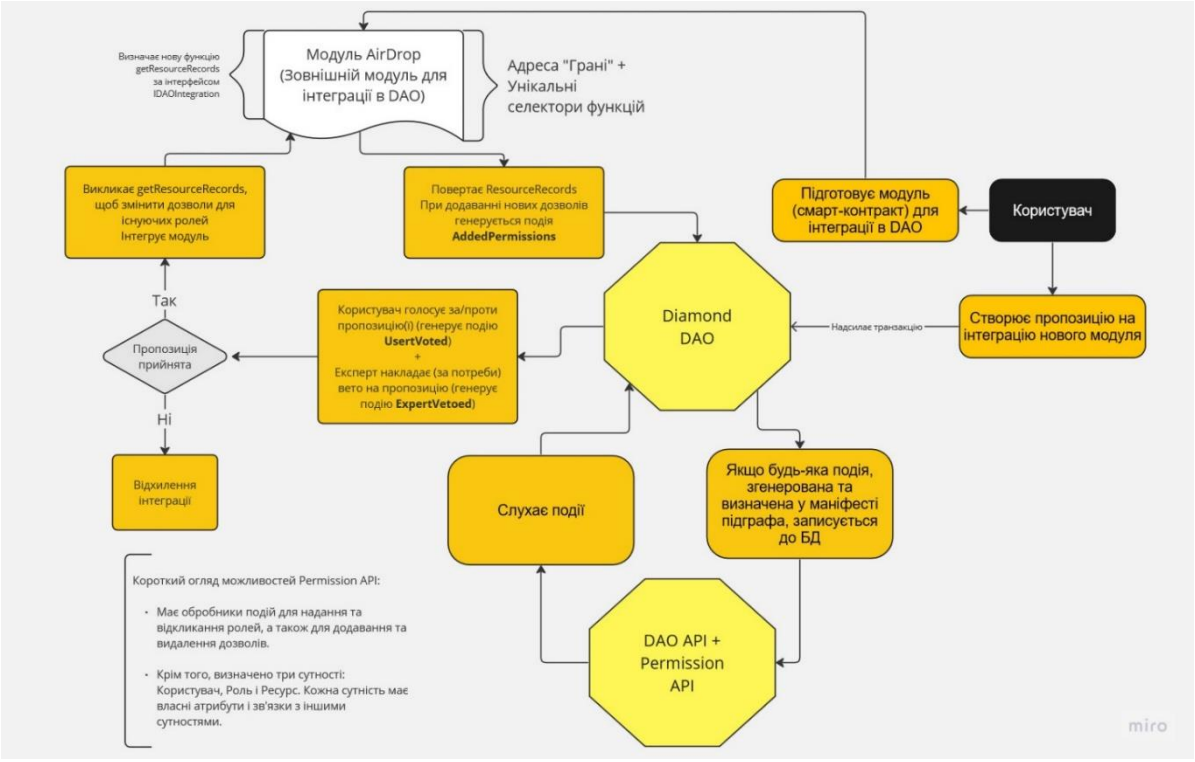

Рисунок 4.1 – Приклад інтеграції нового модуля в ДАО

Без API досягнення цього вимагало б індексації всього блокчейну, а точніше, індексації блокчейну з моменту створення нашого ДАО. Це могло б стати непосильним завданням для звичайного комп'ютера. Крім того, API дозволяє нам уникнути ситуації, коли користувачеві або наявним ролям призначаються схожі дозволи.

Як приклад роботи даного API, в додатку Б продемонстровано запит на отримання користувачів та їх ролей в системі.

4.2.2 Майбутнє та переваги даного API

Однією з ключових переваг підграфа, є те що він розміщений децентралізовано, а користувачі можуть переконатися, що дані отримані з певного підграфа не сфабриковані, перевіривши, що маніфест та/або зіставлення, які в ньому використовуються, відповідають тим, що були зазначені розробником. Процес перевірки включає читання маніфесту підграфа з Міжпланетної файлової системи (IPFS) і перевірку початкового розгортання на The Graph. Така прозорість дозволяє користувачам самостійно перевіряти коректність роботи API.

На мій погляд, це значна перевага протоколу The Graph, який я використовував при створенні API для мого ДАО. Він підвищує гнучкість, безпеку і масштабованість ДАО, виділяючи його серед конкурентів.

Особливу увагу варто приділити сутностям та їх архітектурі. Підграф перестає синхронізуватися у випадку помилки, що робить збої вкрай небажаними. Щоб спростити відновлення, я використав патерн, за яким, якщо сутність не існує в базі даних і ми намагаємося отримати її записи, вона буде автоматично створена. Цей патерн необхідний тільки під час розробки.

Розробка такого програмного інтерфейсу не є звичайною справою. Дуже мало проєктів застосовують цей підхід при розробці своїх ДАО через складність, специфічність технології та її новизну, враховуючи, що самому протоколу тільки п'яти років.

На мою думку, за цим підходом майбутнє розвитку ДАО. Він дозволяє знизити витрати на транзакції, делегуючи аналіз даних The Graph. А також даний протокол відповідає принципам технології блокчейн.

Як говорив один із засновників протоколу Ethereum: «Однією з конкретних причин, близьких мені особисто, є те, що я називаю "підприємницькими суспільними благами": суспільні блага, які нині лише кілька людей вважають важливими, але в майбутньому будуть цінуватися набагато більшою кількістю людей... Проте, нам не потрібно розв'язувати кожну проблему сьогодні.» [15].

Таким чином, інтеграція протоколу The Graph в розробку ДАО є перспективним напрямком для підвищення їх масштабованості, гнучкості та безпеки. Вона забезпечує більш ефективний спосіб управління та взаємодії з даними організації, тим самим покращуючи користувацький досвід.

ВИСНОВКИ

Під час виконання кваліфікаційної роботи, було проведено поглиблене вивчення поточного стану та можливостей децентралізованого API для сервісів на EVM-сумісному блокчейні. Було проведено успішну розробку та тестування ДАО. Цей ДАО був розроблений відповідно до модифікованої версії стандарту EIP 2535, також відомого як "Діамантовий патерн". В результаті система була побудована з єдиною точкою входу, що значно спростило розробку API.

Система управління була вдосконалена шляхом розподілу обов'язків між різними модулями системи. На відміну від нинішніх ДАО, система може бути легко масштабована з великою кількістю зовнішніх модулів та є високо адаптивною. Причиною цього є структура діамантового патерну та модифікована система адміністрування ролей.

Однією з ключових переваг цієї системи є її здатність докорінно трансформувати будь-який процес в рамках ДАО, значною мірою завдяки універсальній організації доступу до параметрів конфігурації системи, які зберігаються в одному місці.

В рамках цього дослідження було розроблено API на основі протоколу The Graph. Цей API дозволяє користувачам ДАО швидко та легко отримати доступ до історичних даних системи, що дає змогу проводити комплексний аналіз системних процесів та надавати інформацію про поточний стан компонентів, яку або неможливо, або ресурсомістко отримати безпосередньо з блокчейну.

Порівняно з наявними системами, архітектура сутностей всередині системи була вдосконалена, щоб уникнути збоїв у її роботі. Завдяки цьому відпала потреба в повній ресинхронізації даних у випадку помилки в процесі індексації блокчейну, яка може зайняти кілька годин при великому навантаженні на сервіс.

У сукупності ці вдосконалення призвели до створення більш безпечної, масштабованої, гнучкої, адаптивної та доступної системи.

Результати цього дослідження мають широкий спектр застосування. Перш за все, для тих, хто бажає розібратися в структурі ДАО і навчитися розробляти API з використанням протоколу The Graph, може бути використана в якості відмінного освітнього ресурсу.

У широкому масштабі цей API може прокласти шлях для розробки нового покоління додатків і сервісів, заснованих на блокчейні. Наприклад, його можуть використовувати фінтех-компанії для розробки децентралізованих фінансових продуктів. Він також може бути корисним у сферах, де доступність даних, масштабованість і прозорість мають першорядне значення, таких як управління ланцюгами постачання, системи голосування і платформи для зберігання та обміну даними тощо.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Wood G. Ethereum: A Secure Decentralised Generalised Transaction Ledger [Електронний ресурс] / Gavin Wood // Berlin Version. – 2022. – Режим доступу до ресурсу: https://ethereum.github.io/yellowpaper/paper.pdf.
2. Introduction to Web3 [Електронний ресурс] – Режим доступу до ресурсу: https://ethereum.org/en/web3/.
3. Nakamoto S. Bitcoin: A Peer-to-Peer Electronic Cash System [Електронний ресурс] / Satoshi Nakamoto // 1. – 2009. – Режим доступу до ресурсу: https://bitcoin.org/bitcoin.pdf.
4. M. Antonopoulos A. Mastering Ethereum: Building Smart Contracts and DApps / A. M. Antonopoulos, G. Wood., 2018. – 422 с. – (1-е). – с. 127-128
5. What Was The DAO? [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: https://www.gemini.com/cryptopedia/the-dao-hack-makerdao.
6. Schmidt M. Q for DAOs [Електронний ресурс] / Martin Schmidt. – 2022. – Режим доступу до ресурсу: https://medium.com/q-blockchain/q-for-daos-e41b757b2af4.
7. Biagosch N. Beyond ‘Code is Law’ — Decentralized Governance in the Web3 World [Електронний ресурс] / Nicolas Biagosch. – 2022. – Режим доступу до ресурсу: https://tinyurl.com/beyond-code-is-law.
8. The Graph Documentation [Електронний ресурс] – Режим доступу до ресурсу: https://thegraph.com/docs/en/about/.
9. Noxx EVM Deep Dives: The Path to Shadowy Super Coder - Part 6 [Електронний ресурс] / Noxx – 2022. – Режим доступу до ресурсу: https://noxx.substack.com/p/evm-deep-dives-the-path-to-shadowy-16e.
10. Sarkar A. Tornado Cash attacker to potentially give back governance control, proposal reveals [Електронний ресурс] / Arijit Sarkar. – 2023. – Режим доступу до ресурсу: https://cointelegraph.com/news/tornado-cash-attacker-to-potentially-giveback-governance-control-proposal-reveals.

11. Vogelsteller F. ERC-20: Token Standard [Електронний ресурс] / F. Vogelsteller, V. Buterin // Ethereum Improvement Proposals, №20. – 2015. – Режим доступу до ресурсу: https://eips.ethereum.org/EIPS/eip-20.
12. Bloemen R. EIP-712: Typed structured data hashing and signing [Електронний ресурс] / R. Bloemen, L. Logvinov, J. Evans // Ethereum Improvement Proposals, №712. – 2017. – Режим доступу до ресурсу: https://eips.ethereum.org/EIPS/eip-712.
13. Chystiakov A. Role-Based Access Control module, RBAC [Електронний ресурс] / Artem Chystiakov – Режим доступу до ресурсу: https://github.com/dl-solidity-library/dev-modules/blob/master/contracts/access-control/RBAC.sol.
14. Mudge N. ERC-2535: Diamonds, Multi-Facet Proxy [Електронний ресурс] / Nick Mudge // Ethereum Improvement Proposals, №2535. – 2020. – Режим доступу до ресурсу: https://eips.ethereum.org/EIPS/eip-2535.
15. Buterin V. Quadratic Payments: A Primer [Електронний ресурс] / Vitalik Buterin. – 2019. – Режим доступу до ресурсу: https://vitalik.ca/general/2019/12/07/quadratic.html.

ДОДАТОК А.
Діаграма роботи DAO API

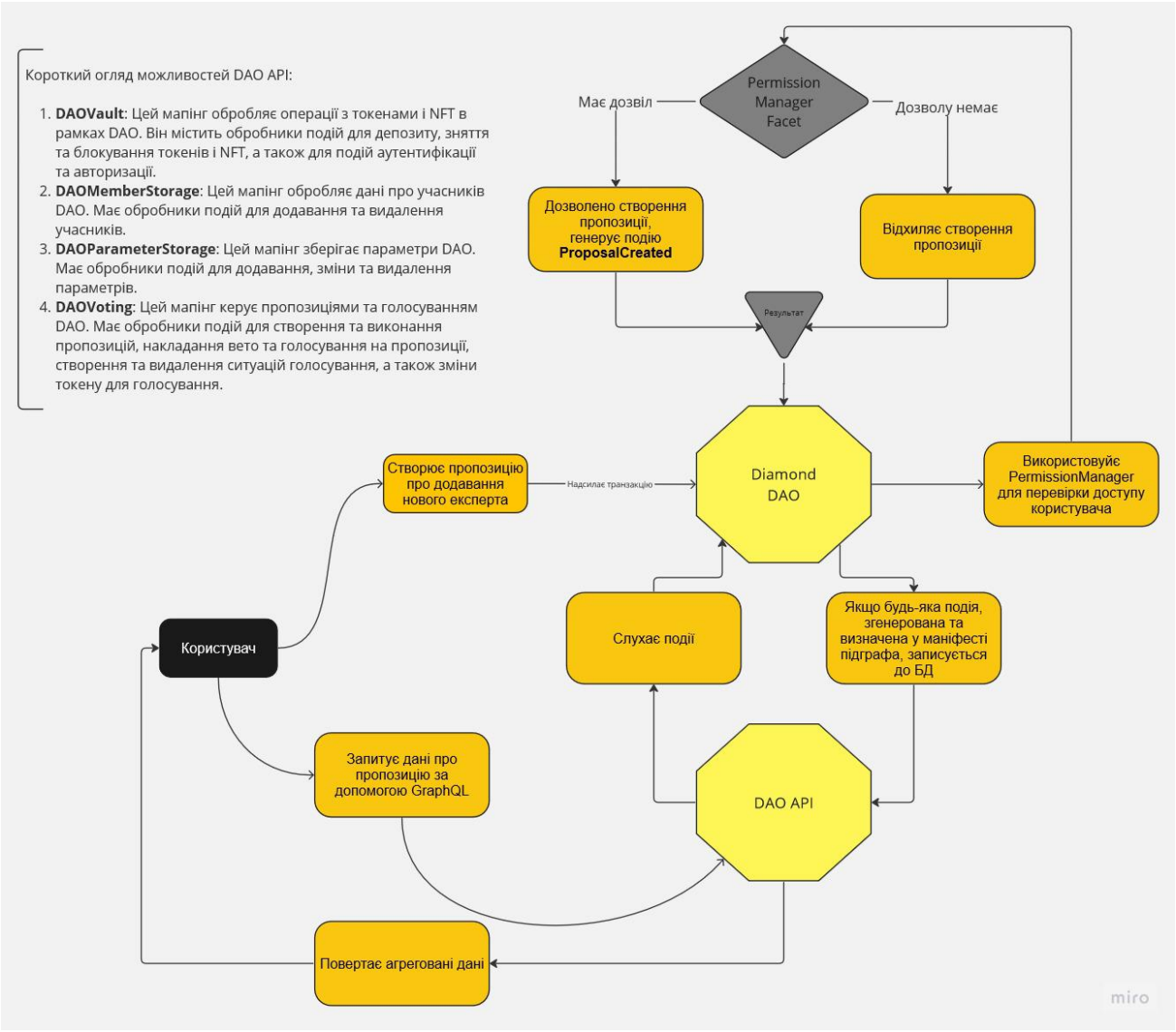

Рисунок А.1 – Діаграма роботи DAO API

ДОДАТОК Б.
Приклад запиту та результату з допомогою DAO API

Це запит на отримання інформації про баланси, токени користувачів в системі та пропозиції, в яких користувачі брали участь:

Рисунок Б.1 – Приклад запиту та результату з допомогою DAO API

ДОДАТОК В.
Приклад запиту та результату з допомогою Permission API

Це запит на отримання користувачів та їх ролей в системі.

```
{
  users {
    id
    roles {
      id
    }
  }
  _meta {
    hasIndexingErrors
    block {
      number
    }
  }
}
```

```
{
  "data": {
    "users": [
      {
        "id": "0x4265f57803fe21c34b86fe5e73a6affbc977bc24",
        "roles": [
          {
            "id": "DAOVotingRole:DAO Token Holder"
          },
          {
            "id": "MASTER"
          }
        ]
      },
      {
        "id": "0xf41cee234219d6cc3d90a6996dc3276ad378cfcf",
        "roles": [
          {
            "id": "MASTER"
          }
        ]
      }
    ],
    "_meta": {
      "hasIndexingErrors": false,
      "block": {
        "number": 6513797
      }
    }
  }
}
```

Рисунок В.1 – Приклад запиту та результату з допомогою Permission API