Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____Комп'ютерних наук_____
(повна назва)

Кафедра _____Інформаційних управляючих систем_____
(повна назва)

КВАЛІФІКАЦІЙНА РОБОТА
Пояснювальна записка

рівень вищої освіти _____другий (магістерський)_____

Дослідження та використання технології розподіленого реєстру при розробці ІС в сфері фінансів
(тема)

Виконав:
студент _2_ курсу, групи ____ІУСТм-22-1_____
_____Млієвський Кирило Юрійович_____
(прізвище, ім'я, по батькові)

Спеціальність ___122 Комп'ютерні науки____
(код і повна назва спеціальності)
Тип програми ____освітньо-професійна_____
Освітня програма ___Інформаційні управляючі системи та технології_____
(повна назва освітньої програми)

Керівник ___зав. кафедри ІУС Костянтин_____
ПЕТРОВ_____
(посада, власне ім'я, прізвище)

Допускається до захисту

Зав. кафедри _____ _____Костянтин ПЕТРОВ___
(підпис) (власне ім'я, прізвище)

2024 р.

Харківський національний університет радіоелектроніки

Факультет_____Комп'ютерних наук_____

Кафедра_____Інформаційних управляючих систем_____

Рівень вищої освіти_____другий (магістерський)_____

Спеціальність_____122 Комп'ютерні науки_____
(код і повна назва)

Тип програми _____освітньо-професійна_____

Освітня програма_____Інформаційні управляючі системи та технології
(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____
(підпис)
« 20 » листопада 20 23 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

студентові_____Мліївському Кирилу Юрійовичу_____
(прізвище, ім'я, по батькові)

1. Тема роботи _____Дослідження та використання технології розподіленого реєстру при розробці ІС в сфері фінансів_____
_____
затверджена наказом університету від _16_ __листопада__2023_ р. № _1359Ст_
2. Термін подання студентом роботи до екзаменаційної комісії _14_ січня 2024 р.
3. Вихідні дані до роботи __науково-технічні публікації та інтернет джерела з тематики кваліфікаційної роботи, матеріали передатестаційної практики_____
_____
4. Перелік питань, що потрібно опрацювати в роботі
провести аналіз предметної області та виконати постановку задачі дослідження; визначити особливості використання технології розподіленого реєстру при розробці інформаційних систем у сфері фінансів; провести експериментальну перевірку можливостей використання технології розподіленого реєстру на прикладі розробки ІС благодійного фонду та функціональне порівняння з використанням клієнт-серверної архітектури; провести апробацію отриманих результатів._____
_____

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Термін виконання етапів роботи	Примітка
1	Отримання завдання на кваліфікаційну роботу	20.11.2023	Виконано
2	Аналіз літератури та Інтернет-джерел	21.11.2023 – 25.11.2023	Виконано
3	Аналіз предметної області	26.11.2023 – 30.11.2023	Виконано
4	Постановка задачі дослідження	01.12.2023	Виконано
5	Аналіз особливостей використання технології розподіленого реєстру	02.12.2023 – 04.12.2023	Виконано
6	Формування вимог до структури блоку, транзакцій та цифрового підпису	05.12.2023 – 08.12.2023	Виконано
7	Дослідження характеристик бізнес-процесів некомерційних фінансових організацій	09.12.2023 – 13.12.2023	Виконано
8	Експериментальна перевірка можливостей використання технології розподіленого реєстру	14.12.2023 – 20.12.2023	Виконано
9	Аналіз діяльності благодійного фонду для оцінки можливостей впровадження технології	21.12.2023 – 24.12.2023	Виконано
10	Апробація отриманих результатів	25.12.2023 – 30.12.2023	Виконано
11	Підготовка пояснювальної записки	31.12.2023	Виконано
12	Підготовка презентації	01.01.2024 – 10.01.2024	Виконано
13	Надання роботи на рецензію	10.01.2024	Виконано
14	Надання роботи для перевірки на плагіат	12.01.2024	Виконано
15	Надання роботи на підпис науковому керівникові	14.01.2024	Виконано
16	Надання підписаної завідувачем кафедрою роботи в ЕК	15.01.2024	Виконано
17	Захист	16.01.2024	Виконано

Дата видачі завдання 20.11.2023

Студент _____
(підпис)

Керівник роботи _____ зав. кафедри ІУС Костянтин ПЕТРОВ
(підпис) (посада, власне ім'я, прізвище)

РЕФЕРАТ

Пояснювальна записка до магістерської кваліфікаційної роботи містить: 95 с., 3 розділи, 30 рисунків, 2 таблиці, 21 джерело, 2 додатки.

БЛАГОДІЙНИЙ ФОНД, БЛОКЧЕЙН-ПРОТОКОЛИ, ДЕЦЕНТРАЛІЗАЦІЯ, ЛАНЦЮЖОК БЛОКІВ, СМАРТ-КОНТРАКТИ, СФЕРА ФІНАНСІВ, ТЕХНОЛОГІЯ РОЗПОДІЛЕНОГО РЕЄСТРУ, ТРАНЗАКЦІЇ, ЦИФРОВИЙ ПІДПИС

Об’єктом дослідження є процес розробки інформаційних управляючих систем з використанням технології розподіленого реєстру.

Предметом дослідження є моделі та технології, що базуються на використанні децентралізованих облікових систем та блокчейн-протоколів у сфері надання фінансових послуг.

Метою дослідження є аналіз існуючих підходів, виявлення особливостей та розробка рекомендацій щодо підвищення безпеки та надійності для визначення факторів аналізу та впровадження розподілених технологій в інформаційних облікових системах, а також вивчення економічного впливу технології розподіленого реєстру на сферу фінансів та визначення переваг та наслідків її впровадження.

Наукова новизна теми дослідження полягає в аналізі, розробці та виявленні нових теоретичних та практичних підходів для забезпечення надійності та підвищення безпекової складової інформаційних систем (ІС) на основі узагальнення досвіду використання децентралізованих облікових систем в сфері фінансів. Дослідження в цій області також дозволить вирішити складні проблеми, що пов'язані з анонімністю, цілісністю та незалежністю даних, а також забезпечить ефективну комунікацію між різними компонентами ІС у контексті розподіленого реєстру.

ABSTRACT

The explanatory note to the master's qualifying work contains: 95 pages, 3 chapters, 30 figures, 2 tables, 21 sources, 2 appendices.

BLOCK CHAIN, BLOCKCHAIN-PROTOCOLS, CHARITABLE FOUNDATION, DECENTRALIZATION, DIGITAL SIGNATURE, DISTRIBUTED REGISTER TECHNOLOGY, FINANCE, SMART CONTRACTS, TRANSACTIONS

The object of research is the process of developing information management systems using distributed ledger technology.

The subject of the study is models and technologies based on the use of decentralized accounting systems and blockchain protocols in the field of financial services.

The purpose of the research is to analyze existing approaches, identify features and develop recommendations for improving security and reliability to determine the factors of analysis and implementation of distributed technologies in information accounting systems, as well as to study the economic impact of distributed ledger technology on the financial sector and determine the benefits and consequences of its implementation.

The scientific novelty of the research topic lies in the analysis, development and identification of new theoretical and practical approaches to ensure the reliability and enhancement of the security component of information systems (IS) based on the generalization of the experience of using decentralized accounting systems in the field of finance. Research in this area will also help to solve complex problems related to the anonymity, integrity and independence of data, as well as ensure effective communication between different IS components in the context of a distributed ledger.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАКИ

БД – база даних

ІС – інформаційна система

МЗ – математичне забезпечення

ОС – операційна система

ПЗ – програмне забезпечення

СУБД – система управління базами даних

API – інтерфейс програмного забезпечення (англ., Application Programming Interface)

DFD – діаграма потоку даних (англ., Data Flow Diagram)

DLT – технологія розподіленого реєстру (англ., Distributed Ledger Technology)

ECDSA – алгоритм цифрового підпису на еліптичних кривих (англ., Elliptic Curve Digital Signature Algorithm)

P2P – однорангова мережа (англ., Peer-to-Peer)

RLP – протокол радіоканалу (англ., Recursive Length Prefix)

RPC – віддалений виклик процедури (англ., Remote Procedure Call)

SQL – мова структурованих запитів (англ., Structured Query language)

TCP – протокол управління передачами (англ., Transmission Control protocol)

UML – уніфікована мова моделювання (англ., Unified Modeling Language)

ВСТУП

Актуальність теми дослідження обумовлена тим, що у сучасному цифровому світі велику роль відіграє автоматизація та розвиток технологій зберігання, обробки, узгодження та синхронізації даних в процесах управління, в яких беруть участь один або декілька незалежних сторін. З появою методів та технологій, які формують мережевий рівень взаємодії між учасниками, з'являється перспектива забезпечення рівноправних та безпечних каналів передачі та обміну даними, що в першу чергу сформує та надасть нові можливості для розвитку інформаційних систем з використанням методик блокчейну, які базуються на принципах розподіленості та децентралізації. Важливість та затребуваність дослідження також зумовлена фактором анонімності контролю та моніторингу облікової системи, концепція якого полягає в усуненні необхідності затвердження вповноваженого посередника або оракула, такого як спеціалізована установа або структура. Крім того, дослідження обраних методів та технологій може сприяти створенню більш ефективних та швидких засобів обробки операцій. Отримані модулі зможуть прискорити процеси підтвердження та верифікації операцій, зменшуючи затримки та надмірність мережі.

Об'єктом дослідження є процес розробки інформаційних управляючих систем з використанням технології розподіленого реєстру.

Предметом дослідження є моделі та технології, що базуються на використанні децентралізованих облікових систем та блокчейн протоколів у сфері надання фінансових послуг.

Метою дослідження є аналіз існуючих підходів, виявлення особливостей та розробка рекомендацій щодо підвищення безпеки та надійності для визначення факторів аналізу та впровадження розподілених технологій в інформаційних облікових системах, що пов'язані з фінансовими структурами та установами, а також вивчення економічного впливу технології розподіленого реєстру на сферу фінансів та визначення переваг та наслідків її впровадження. Зокрема, метою

роботи є розширення загальних принципів і вдосконалення методологій мережевої взаємодії та пірінгових протоколів, які наразі недостатньо досліджені через замалу існуючу базу знань.

Методика проведення дослідження. Структура і логіка дослідження підпорядковані рішенню поставлених завдань [1]. При вирішенні конкретних завдань використовуються методи логічного, функціонального та системного аналізу із застосуванням економіко-математичного апарату, статистичні методи, методологія використання та впровадження блокчейн технологій, методи порівняння алгоритмів по критеріям безпеки, надійності, масштабованості та доцільності при реалізації ІТ-проєктів [2-3].

Наукова та практична новизна дослідження. Наукова новизна теми дослідження полягає в аналізі, розробці та виявленні нових теоретичних та практичних підходів для забезпечення надійності та підвищення безпекової складової інформаційних систем (ІС) на основі узагальнення досвіду використання децентралізованих облікових систем в сфері фінансів. Дослідження в цій області також дозволить вирішити складні проблеми, що пов'язані з анонімністю, цілісністю та незалежністю даних, а також забезпечить ефективну комунікацію між різними компонентами інформаційних систем у контексті розподіленого реєстру. Практична новизна даної теми виходить з можливості застосування отриманих та розроблених методів та моделей за різними призначеннями та потребами, що пов’язані з реалізацією процесів обліку та моніторингу у сфері фінансів. Облікові інформаційні системи, які побудовані на основі розподіленого реєстру, можуть знайти широке застосування в таких інфраструктурах, як платіжні системи, банківський сектор, благодійні фонди тощо.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ТА ПОСТАНОВКА ЗАДАЧІ ДОСЛІДЖЕННЯ

1.1 Аналіз технології розподіленого реєстру та блокчейн протоколів

Технологія розподіленої реєстру (DLT) – це сукупність технологічної інфраструктури та ряду протоколів, які забезпечують одночасний доступ, перевірку й оновлення записів у мережевій базі даних. Це технологія, на основі якої створено блокчейни, а інфраструктура дозволяє користувачам переглядати будь-які зміни та хто їх вніс, зменшує потребу в перевірці даних, гарантує надійність даних і надає доступ лише тим, кому вони потрібні [4].

Насправді технологія розподіленого реєстру і ринок криптовалюти демонструють активний розвиток і викликають значний інтерес у суспільстві. Ця технологія вносить та регламентує певні нововведення, які полягають в тому, що інформація про транзакції вже не зберігається в централізованій базі даних, а передається на комп'ютери всіх учасників мережі, які зберігають дані локально. Першим прикладом застосування подібної технології став Bitcoin (описаний у 2008 році та реалізований у 2009 році) – криптовалюта, яка працює на його основі та бухгалтерська книга для всіх операцій та транзакцій. Вона надала потужний поштовх для створення інших блокчейн-додатків, багато з яких активно розробляються та використовуються в фінансовій сфері.

За визначенням блокчейн – це розподілена база даних, яка містить впорядкований ланцюжок записів, відомих як блоки, що постійно та безперервно збільшується у розмірі у міру додавання блоків у ланцюжок, що значно підвищує безпеку реєстру. Блоки в цій базі пов’язані між собою та захищені від підробки та спотворення за допомогою засобів криптографії. Кожен блок містить часову позначку, геш-суму попереднього блоку і дані транзакцій, що представлені у вигляді геш-дерева. Термін "блокчейн" використовується для позначення такої розподіленої бази даних.

Проте, можна стверджувати, що для визначення терміну «блокчейн» існує дві варіації:

– розподілена база даних;

– безперервний послідовний ланцюжок блоків, що містить та зберігає інформацію облікової системи.

Обидва визначення вірні у своїй суті, але не дають повної відповіді на питання про значення даної технології. Якщо розглядати протокол з точки зору обліку, то блокчейн – це спільна незмінна книга, яка полегшує процес запису транзакцій і відстеження активів у бізнес-мережі. Актив може бути матеріальним (будинок, автомобіль, готівка, земля) або нематеріальним (інтелектуальна власність, патенти, авторські права, брендинг). Практично все, що має цінність, можна відстежувати, зберігати та продавати в мережі блокчейн, що знижує ризики та скорочує витрати для всіх учасників. Також можна зазначити, що блокчейн є влучним засобом для обміну даними стосовно бізнес-вимог та бізнес-процесів, необхідних для прийняття проєктних рішень, оскільки він надає негайну спільну та повністю прозору інформацію, що зберігається в незмінній книзі, до якої можуть отримати доступ лише авторизовані учасники мережі. Блокчейн-мережа може відстежувати замовлення, платежі, рахунки, виробництво та багато іншого. А оскільки кожен учасник володіє повною та актуальною версією облікової системи, тоді з’являється можливість переглянути всі деталі транзакції від початку до кінця, що зможе надати нові можливості для аудиту та оцінки проєкту.

Можна виділити такі ключові атрибути блокчейну.

1. Розподілена книга. Усі учасники мережі мають доступ до розподіленої книги та її незмінного запису транзакцій. За допомогою цієї спільної книги транзакції реєструються лише один раз, усуваючи дублювання даних, що є типовим для традиційних бізнес-мереж.

2. Незмінні дані. Жоден учасник не може змінювати або втручатися в транзакцію після її запису до спільної книги. Якщо запис транзакції містить помилку, необхідно додати нову транзакцію, щоб скасувати помилку, і тоді обидві транзакції стануть видимими.

3. Смарт контракти. Щоб пришвидшити транзакції, набір правил – так званий смарт-контракт – зберігається в блокчейні та виконується автоматично. Смарт-контракт, наприклад, може визначати умови для переказу корпоративних облігацій, містити умови для оплати туристичної страховки та багато іншого.

4. Консенсус. Це механізм, за допомогою якого вузли в мережевих застарілих системах узгоджують порядок і дійсність даних, які додаються до блокчейну. Це важлива частина блокчейну, оскільки вона гарантує, що дані в блокчейні надійні та їм можна довіряти. Він є необхідним, оскільки в децентралізованій мережі немає центрального органу, який міг би забезпечити виконання правил і забезпечити цілісність даних на повних вузлах [4]. Натомість мережа покладається на консенсус своїх учасників для перевірки транзакцій і додавання їх до блокчейну як системного потоку на інших вузлах.

Механізми досягнення консенсусу розподіляються за наступними рівнями.

1. За доказом виконаної роботи (Proof-of-Work):

– загальна кількість валідаторів, що відповідають за прийняття рішень невідома;

– валідатори анонімні та не мають репутації серед інших;

– голос закріплюється доказом роботи валідатора;

– консенсус досягнуто, якщо особи, що контролюють більшість потужності системи, узгодили усі умови.

2. За доказом частки володіння (Proof-of-Stake):

– валідатор визначається алгоритмом;

– шанс голосу пропорційний балансу валідатору;

– при помилці на валідатора накладається штраф;

– консенсус досягнуто, якщо особи, що володіють більшим фінансовим балансом, затвердили стан бази даних облікової системи.

3. За підтвердженням повноважень:

– валідатор визначається за рахунок репутації та значних внесків або покращень роботи системи;

– ідентичність валідатора має бути офіційно підтверджена;

– особа, що бажає стати валідатором, повинна вкладати певні ресурсі – фінансові або трудові.

4. BFT (Byzantine Fault Tolerance):

– загальна кількість валідаторів відома та визначена заздалегідь;

– кожен валідатор розголошує свою особистість;

– використовується для приватних та обмежених за правами систем.

5. FBA (Federated Byzantine Agreement):

– має високу пропускну здатність та масштабованість;

– складається з кворумів – мінімальної кількості вузлів, необхідних для того, щоб рішення було правильним [4].

Для коректної роботи облікової системи, що використовує технологію розподіленого реєстру, важливим аспектом є побудова надійної мережі.

Надійність мережі визначається за рахунок наступних критеріїв.

1. Публічність. Публічна мережа – це та, до якої може приєднатися та брати участь будь-яка система. Недоліки можуть включати значну необхідну обчислювальну потужність, низьку або повну відсутність конфіденційності транзакцій і слабкий захист. Для корпоративних систем можуть нести великий ризик.

2. Приватність. Приватна мережа є подібною до публічної, але є децентралізованою одноранговою мережею. Проте, одна організація керує мережею, контролюючи, кому дозволено брати участь, виконувати консенсусний протокол і підтримувати спільну книгу. Залежно від варіанту використання це може значно підвищити довіру та впевненість між особами, що взаємодіють один з одним [5].

3. Рівень дозволеності. Системи, які забезпечують приватність, зазвичай повинні містити обмеження за рівнем дозволеності. Також варто зауважити, що публічні мережі також можуть містити подібні обмеження. Обмеження за рівнем дозволеності накладається за рахунок визначення прав та ролей осіб, кому дозволено брати участь у мережі та в яких операціях.

4. Організаційна структура. Кілька організацій або підприємств можуть розділити відповідальність за підтримку мережі розподіленого реєстру. Попередньо згруповані організації визначають, хто може проводити транзакції або мати доступ до даних.

Щоб технологія розподіленого реєстру була ефективна в інформаційній системі, вона повинна забезпечувати наступні можливості:

– зберігання та облік;

– передача активів між користувачами;

– якщо в процесі управління системою залучено кілька незалежних сторін які мають довіряють один одному.

Технологія розподіленого реєстру може мати численні варіанти використання в сфері фінансів (наприклад, біржі, фонди, банки, аукціони, страхування, деривативи). Усі ці структури використовують інструменти для взаємодії з фінансами в децентралізованій обліковій системі за рахунок застосування програмного забезпечення у вигляді смарт-контрактів – додатків, що представляють надбудови для визначення алгоритмів роботи мережі.

Ключовим компонентом для функціонування IT-проєктів в сфері фінансів з використанням технології розподіленого реєстру є поняття криптовалюти – певного децентралізованого цифрового активу, який видається за певним правилом та алгоритмом, надсилається та отримується у вигляді транзакцій, містить захист у вигляді приватних та публічних ключів та правила протоколу, які можна змінювати у випадку досягнення консенсусу.

1.2 Огляд мережевих протоколів розподіленого реєстру

Технологія розподіленого реєстру представляє собою унікальне поєднання кількох інформаційних засобів та методів, таких як апаратний рівень, рівень даних, рівень консенсусу та прикладний рівень, кожен з яких забезпечує широкий спектр

можливих застосувань у сфері фінансових послуг. Проте, для того, щоб даний набір моделей міг певним чином здійснювати обмін даними та взаємодіяти один з одним, необхідно мати безпечну, масштабовану та децентралізовану структуру архітектури мережі, що утворює мережевий рівень. З плином часу та оновленням блокчейн технологій мережевий рівень все більше розростався, набір мережевих протоколів покращувався та розширювався, в зв'язку з чим подібні етапи було розподілено на різні стадії в залежності від складності, масштабованості та функціональності.

Розглянемо їх більш детально:

– нульовий рівень. Складається з апаратного забезпечення, набору протоколів та інших компонентів, які формують основу екосистеми блокчейну, Він діє як мережева архітектура, що лежить в основі блокчейну. Цей рівень можна розглядати як «мережу блокчейнів». Працездатність між ланцюжками також забезпечується на даному рівні, який дозволяє блокчейнам спілкуватися один з одним. Він забезпечує важливу роль для вирішення проблеми масштабованості;

– перший рівень. Відповідає за виконання більшості завдань, які підтримують основні операції мережі блокчейн, такі як механізм консенсусу, віртуальна машина, тощо. Фактично, він ідентифікує блокчейн як монолітну структуру. Велика кількість завдань, якими повинен керувати цей рівень, може спричинити проблеми з адаптивністю. У міру того, чим більше людей навантажує реєстр, тим більшу кількість обчислювальної потужності необхідно для вирішення та додавання блоків даних у спільну базу даних, що призводить до значних витрат та довшого часу обробки запиту. Проте, дане питання вирішено за допомогою появи шардингу, тобто поділу обчислювальних операцій на менші частини. Наразі до розподілених реєстрів першого рівня відносяться одні з найвідоміших блокчейн систем – Bitcoin та Ethereum;

– другий рівень. Щоб підвищити продуктивність, необхідно було визначити додаткову потужність обробки. Однак це вимагало включення додаткових вузлів, що певною мірою засмічує мережу. Хоча додавання вузлів має важливе значення для підтримки децентралізованого характеру блокчейну, проблеми з

масштабованістю, безпекою та пропускною здатністю можуть вплинути на інші вузли на першого рівня. Як наслідок, було вирішено розширити перший рівень, перемістивши всю обробку транзакцій на другий рівень, створений у вигляді надбудови. Це стало можливим завдяки інтеграції рішень сторонніх розробників із додатками попереднього рівня. Нова мережа стає оновленою та має змогу керувати всіма перевірками та валідаціями транзакцій. Для коректної роботи, системи, побудовані на цьому рівні, повинні постійно обмінюватися інформацією з системами попередніх рівнів, які відповідають лише за керування та розподіл блоків;

– третій рівень. Визначає останній шар екосистеми блокчейн, інтерфейс користувача. Це рівень, за допомогою якого учасники можуть зрештою взаємодіяти з обліковою системою, шляхом перегляду графічних елементів та екранних форм, а не програмного коду чи алгоритму. На цьому рівні блокчейн використовується для створення комплексних інтегрованих систем та нових бізнес-моделей. Головною перевагою є зручність та зрозумілість. Також має користь для внутрішньої та міжланцюгової фінансової взаємодії, наприклад, децентралізованої біржі, стейкінгу тощо. Представлені у вигляді децентралізованих програм, наприклад, електронного гаманця [5, 6].

Мережеві протоколи розподіленого реєстру надають безпосередні переваги при підтримці та впровадженні ІТ-проєктів зі сфери фінансів у ринок.

Серед них:

– спільне використання ресурсів. У мережі користувачі можуть спільно використовувати різні ресурси, такі як файли, документи, з іншими комп’ютерами в мережі. Це означає, що є можливість отримати доступ до ширшого спектру ресурсів;

– масштабованість. Обумовлюється динамічним зростанням або звуженням кількості підтримуваного апаратного забезпечення, у випадку коли вузли приєднуються або залишають мережу. Це означає, що існують засоби для створення та об’єднання великої кількості вузлів, які не впливають на продуктивність і також можуть продовжувати функціонувати, навіть якщо деякі

вузли виходять з мережі. Подібні системи можна легко розширити, включивши нові функції або послуги, не вимагаючи повного перегляду архітектури;

– анонімність. Обумовлюється забезпеченням високого рівня анонімності, оскільки немає потреби проходити верифікацію або ідентифікувати себе для інших учасників системи;

– безпека. Оскільки у розподіленому реєстрі немає центральної точки контролю, вона може бути більш стійкою до таких атак, як: атака на відмову в обслуговуванні (DoS), «людина посередині», грубий підбір, інтерполяція, інтегральний та диференціальний аналіз, оскільки немає головного та цільового компоненту, що потенційно може мати вразливості. Проте можуть виникати вразливості до атак, які використовують слабкі місця в протоколі;

– автономність. Під автономністю розуміється більший контроль над обчислювальною потужністю та ресурсами інформаційних систем.

Технологія розподіленого реєстру здатна використовувати різноманітні мережеві протоколи для встановлення та подальшого полегшення зв’язку між вузлами в мережі.

Дані протоколи наступні:

– UDP. Це протокол користувацьких датаграм, який дозволяє учасникам знаходити один одного та обмінюватися інформацією по мережі без необхідності попереднього з’єднання для встановлення шляхів даних або спеціальних каналів передачі;

– TCP. Це протокол управління над надсиланням даних, є надійним протоколом, який дозволяє вузлам обмінюватися великими обсягами даних, наприклад заголовками блоків або транзакціями;

– RLPx. Це криптографічний протокол кадрування повідомлень транспортного рівня, який використовується для інкапсуляції та відокремлення повідомлень розподіленого реєстру у форматі JSON-RPC, який легко передавати через мережу [7].

У якості головної концепції, необхідної для дослідження технології розподіленого реєстру при реалізації IT-проєктів в сфері фінансів буде використано

на основі платформи першого рівня – Ethereum, який надає потужні інструменти для створення бізнес-рішень, які представлені у вигляді децентралізованих сервісів, побудованих на основі смарт контрактів.

1.3 Порівняння технології розподіленого реєстру та централізованих систем обліку фінансів

Всього виділяють три типи побудови мережі проєкту: централізована клієнт-серверна, децентралізована розподілена та гібридна архітектури.

На рисунку 1.1 представлено загальну схему топології мережевого рівня для централізованих облікових систем.

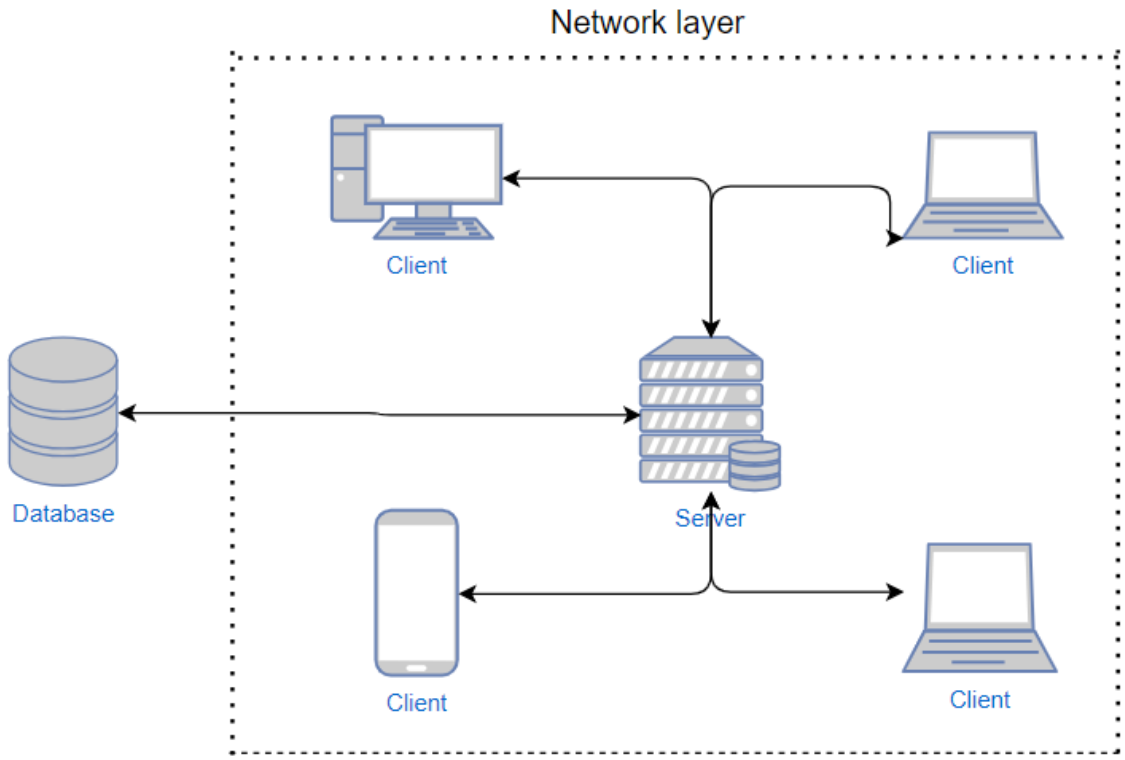

Рисунок 1.1 – Схема топології мережевого рівня централізованого реєстру

В основі централізованих систем лежить централізоване керування процесами. У деяких випадках такий підхід може забезпечити більшу ефективність функціонування системи. При цьому база даних зберігається на головному сервері у хмарі або кластері, система контролюється централізовано у вигляді певної установи та для проведення операцій користувачі надсилають запити до єдиного реєстру системи. Тому, у загальній клієнт-серверній архітектурі один або декілька клієнтів спілкуються з центральним сервером, комп'ютером, тощо. Проте, мають місце ризики, які необхідно враховувати при роботі у такій архітектурі, а саме:

– обмеження пропускної здатності з боку серверу;

– проблема відновлення втраченого набору даних;

– схильність до цензури і анонімності;

– єдина точка відмови.

На рисунку 1.2 представлено загальну схему топології мережевого рівня для децентралізованих облікової системи.

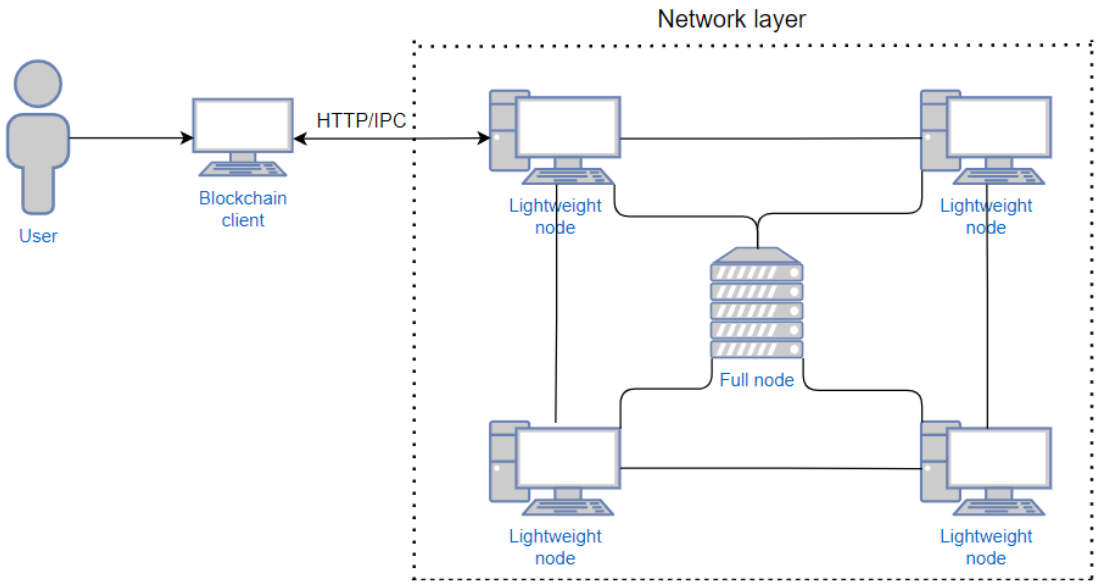

Рисунок 1.2 – Схема топології мережевого рівня розподіленого реєстру

Архітектура, що наведена на рисунку 1.2, представляє пірінгову розподілену мережу за концепцією Peer-to-Peer або Person-to-Person (P2P). Розподілена пірінгова архітектура складається з децентралізованої мережі однорангових вузлів – які можуть виступати як у ролі клієнтів, так і серверами, тобто ці обидві ролі фактично поєднуються в один компонент – вузол. Такий підхід дозволяє забезпечити збільшення рівня незалежності кожного компоненту системи, зберегти баланс між ефективністю роботи та застосовуваними засобами та цілісність системи. Мережа P2P розподіляє робоче навантаження між рівними вузлами, і всі ці однорангові вузли вносять внески та споживають ресурси в мережі без потреби централізованої установи. Проте, вузли також можуть мати певні відмінності у споживанні та обміну даними в залежності від типу вузла. У чистому вигляді архітектура P2P є повністю децентралізованою. Однак на практиці може існувати центральний сервер відстеження, розташований поверх мережі P2P, щоб допомогти вузлам знаходити один одного та керувати мережею. Такий підхід наразі використовується у мережі Napster для миттєвого обміну файлами з використанням розподіленого реєстру. Крім того, однорангові мережі можна класифікувати на дві основні категорії залежно від того, як дані в мережі пов’язані один з одним:

– неструктуровані мережі. Це тип мережі, у якій зв’язки між клієнтами, комп’ютерами, серверами, апаратними пристроями встановлюються випадковим чином. Подібні мережі легко побудувати, оскільки будь-який новий пристрій, який бажає приєднатися та обмінятися даними у мережі, може зробити це шляхом копіювання існуючих фрагментів що відповідають вихідному запиту, а потім сформувати та поширити власні. Однак запити не завжди можуть отримати доступ та права на поширення або бути доступними тільки певному колу осіб;

– структуровані мережі. На противагу неструктурованим, структуровані мережі дозволяють кожному одноранговому вузлу стежити за певним вмістом в мережі. Ці мережі призначають певне значення та відношення вмісту до вузла, за яким потім слідує спеціалізований алгоритм, який визначає, який розділ відповідає наведеному вмісту, у вигляді геш-таблиць. Таким чином, щоразу, коли хтось

звертається до вузла для пошуку контенту, мережа використовує загальний протокол для визначення розділу, відповідального за передачу даних, і направляє пошуковий запит до однорангового вузла, відповідального за це.

Підсумовуючи, можна вважати, що архітектура мережі розподіленого реєстру розроблена з урахуванням функцій у вигляді однієї програми. Таким чином, кожна програма несе відповідальність за виконання власної задачі, при цьому одночасно здійснювати ролі постачальника та замовника, які мають однакові обов'язки та можливості.

З математичної точки зору, пірінгову архітектру розподіленого реєстру можна розглядати як орієнтований граф:

$$G = (V, E), \tag{1.1}$$

де V – ребра, набір однорангових вузлів у мережі;

E – дуги, набір зв'язків між одноранговими вузлами [8].

Отже, ребра складаються з списку вузлів, кожен з яких має унікальний ідентифікаційний номер (id). Дуга, тобто напрямлене ребро, може складатись з множини кортежів зв'язків, які означають, що певний вузол має прямий шлях для надсилання даних до іншого, при цьому використовуючи id- ідентифікатор у якості адресату, при цьому кожен вузол повинен мати певний одне або декілька з'єднань. Незважаючи на те, що в базовій клієнт-серверній мережі подібні IP-адреси можуть транслюватися до найближчих фізичних місць, але у блокчейні зрідка існує подібна пряма кореляція.

Структура графа забезпечує кілька варіацій з'єднання між кожною парою однорангових вузлів і сприяє стійкості, забезпечуючи підключення, незважаючи на зміни та модифікації однорангового вузла. На рівні кожного однорангового вузла зв'язність графа відображається в термінах його суміжності з іншими компонентами. Коли однорангові вузли приєднуються або залишають мережу, сусідні вузли можуть мати неправильну інформацію щодо кінцевого стану облікової системи, тому використовується механізми обслуговування накладання

для підтримки інформації про суміжність оновленою, таким чином зберігаючи зв’язок між усіма вузлами.

Учасники децентралізованої мережі надають частину своїх ресурсів іншим учасникам мережі. Кожен одноранговий вузол забезпечує обчислювальні цикли, дискове сховище та високу пропускну здатність мережі.

Після встановлення однорангових з’єднань виникає обмін інформацією про блокчейн через зашифровані та аутентифіковані канали з’єднання TCP\IP.

В порівнянні з клієнт-серверною архітектурою, технологія розподіленого реєстру має наступні переваги:

– розподілена архітектура;

– відсутність централізованого управління. У одноранговій мережі жоден вузол немає контролю над мережею, що робить її стійкішою до цензурі, злому чи збоям;

– економічна ефективність. Децентралізовані мережі можуть бути економічно ефективнішими, ніж централізовані мережі, оскільки вони вимагають менше інфраструктури та обладнання. Такі мережі можуть покладатися на апаратне забезпечення, яке належить користувачам або клієнтам, для обробки зв’язку та зберігання даних, при цьому зменшуючи витрати, пов’язані з обслуговуванням центрального сервера;

– масштабованість. Забезпечується за рахунок потенційної обробки більшої кількості даних і трафіку, при цьому не впливаючи на загальне завантаження системи. Крім того, додатково забезпечується стійкість до стрибків трафіку, особливо під час пікового навантаження;

– конфіденційність і безпека. P2P мережі забезпечують більшу конфіденційність і безпеку, оскільки прямий зв’язок між вузлами ускладнює перехоплення або моніторинг зв’язку третім сторонам. Крім того, для побудови облікових систем даного типу часто використовують шифрування та інші методи криптографії для захисту користувачів від зловмисників;

– відмовостійкість. Система може зберігати свою працездатність навіть після відмови одного або декількох її компонентів та модулів;

– незмінність кінцевого стану бази даних;

– формальність протоколів.

1.4 Проблеми впровадження технології розподіленого реєстру у сфері фінансів

Незважаючи на те, що технології розподіленого реєстру мають значний потенціал у вирішенні багатьох економічних питань для різних галузей зі сфери фінансів, на практиці при реалізації таких технологій можуть виникати певні проблеми, які згодом можуть бути занадто ризиковими для компанії та понести великі збитки при розробці або впровадженні IT-проєктів. Також, з урахуванням того факту, що зі збільшенням рівня децентралізації зростає надійність та стійкість системи, збільшувати цей рівень не завжди просте завдання. У процесі децентралізації можуть зустрічатися деякі складності та обмеження. Одним з них є складність оновлення протоколу. Складність застосування будь-якого оновлення протоколу взаємодії у децентралізованому середовищі полягає в тому, що запропоноване оновлення має бути підтримано переважно більшою кількістю активних учасників мережі. Для цього, стороні, яка пропонує оновлення, необхідно довести необхідність його впровадження іншим вузлам системи, що саме собою є дуже складним завданням. Одним з прикладів можна навести оновлення процесу майнінгу системи з алгоритму Proof-of-Work на Proof-of-Stake, що хоч і призвело до зниження кількості споживаних ресурсів при роботі, але також зайняло декілька років на остаточне прийняття рішення стосовно переходу. Проблема відповідальності також є обмеженням до впровадження, оскільки будь-яке рішення в децентралізованій системі є результатом згоди більшості учасників, вибраних певним чином, тому не існує єдиної сторони або центру валідації, який зміг би скасувати рішення, ухвалене спільно, або навести своє. Але якщо користувач або система зазнає атаки хакерів або зловмисників, тоді швидко знайти винного скоріш

за все не вдасться, та в більшості випадків буде неможливим. Тому в такому випадку усі ризики становляться на стороні користувача та сам розрахунок ризиків буде являтися достатньо трудомістким процесом. Складний процес монетизації розробки слугує фактором, що уповільнює впровадження децентралізованих систем обліку фінансів, оскільки монетизувати та документувати централізовану систему зазвичай легше, ніж децентралізовану. Централізована система є більш керованою і може мати юридичну підтримку – існує призначений представник, відповідальний за ведення облікової діяльності у межах правового поля. В таких умовах простіше побудувати бізнес-модель та визначити бізнес-вимоги до проектування продукту. У децентралізованій системі, навпаки, досить складно забезпечити захист авторського права та проводити контроль якості. Децентралізовані системи також стикаються з необхідністю зберігання надлишкового обсягу даних та затримками при синхронізації вузлів. Зокрема, можуть існувати й інші чинники, які можуть стояти на шляху поширення ПЗ, призначеного для роботи в децентралізованому середовищі. Наприклад, мережевий трафік може піддаватися фільтрації з боку Інтернет-провайдера, що може призвести до втрати деяких даних.

На рисунку 1.3 наведено схематичне зображення ризиків, які можуть виникати при реалізації інформаційних систем у сфері фінансів з використанням блокчейн-протоколів.

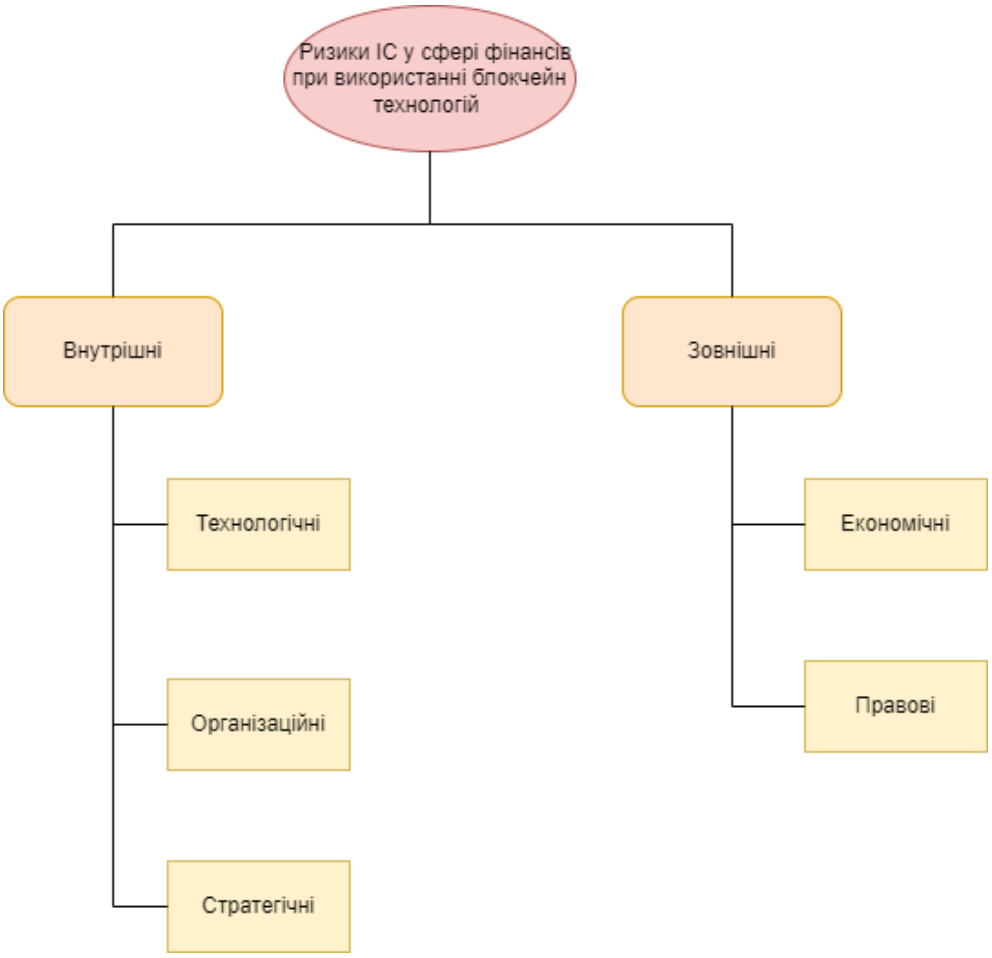

Рисунок 1.3 – Ризики впровадження ІС у сфері фінансів з використанням технології розподіленого реєстру

У контексті впровадження ІТ-проєкту ризик визначається нетиповою подією або умовою, при виникненні якій загальний стан продукту може приймати позитивні або негативні наслідки та зміни. В даному випадку ризики можна умовно поділити на внутрішні – ризики, що зумовлені діяльністю самого проєкту та зовнішні – ті ризики, які безпосередньо не пов’язані з діяльністю підприємства.

До внутрішніх відносяться ризики, що пов’язані з технологічними, організаційними та стратегічними процесами:

– технологічні ризики. Використання нових протоколів та технологій є важливим аспектом для побудови успішної стратегії розвитку проєкту з використанням технології розподіленого реєстру, але при цьому використання сучасних та ще недостатньо досліджених методологій може бути наслідком

непередбачуваних подій. До них відносяться конфіденційність та приватність даних і транзакцій у блокчейні, обмеження, що пов’язані з продуктивністю платформи розподіленого реєстру та проблеми, пов’язані з інтеграцією проєкту з іншими корпоративними або комерційними системами;

– організаційні ризики. Пов’язані з рішеннями щодо можливих оновлень та покращень системи у децентралізованому середовищі, оскільки наявність незалежних один від одного сторін та учасників ускладнює вирішення ряду питань, таких як управління, контроль якості, аудит стану бази даних, тощо [9];

– стратегічні ризики. Пов’язані з можливою втратою, зниженням рівня розвитку або недостатньою конкурентоспроможністю на ринку. Таким чином, це може впливати на низку стратегічних проблем, включаючи визначення відповідної ціннісної пропозиції, управління брендом і репутацією та управління змінами [9].

До зовнішніх ризиків відносяться ризики, які пов’язані з економічними та правовими концепціями.

– економічні ризики. Обумовлені змінами в економіці держави та світу в цілому. Пов’язані з можливими фінансовими втратами та ризиками, що пов’язані з фінансуванням валідаторів та тих, хто підтримує роботу розподіленого реєстру. Все це слід брати до уваги під час створення блокчейн-додатків, платформ та інфраструктури у сфері фінансів. Крім того, існують різноманітні труднощі з обліком та звітністю;

– правові ризики. Пов’язані з особливостями нормативно-правових актів та вимог законодавства. Не зважаючи на те, що блокчейн як технологія не регулюється, програми, які використовують цю технологію, повинні відповідати загальному регламенту щодо захисту даних і конфіденційності (GDPR). Невизначеність щодо юридичних правил, антимонопольні порушення, можливості довільного виконання смарт-контрактів, протидія махінацій та шахрайства (AML), ідентифікація клієнтів (KYC), а також захист інтелектуальної власності (IP) поєднують регуляторні ризики [9].

1.5 Постановка задачі дослідження

Роботу присвячено дослідженню можливостей використання технології розподіленого реєстру при реалізації ІТ-проектів у сфері фінансів.

Технологія розподіленого реєстру (DLT), також відома як блокчейн, є інноваційним рішенням, яке здатне змінити традиційний підхід до фінансових операцій та послуг. Використання блокчейн-протоколів у сфері фінансів відкриває широкий спектр можливостей для покращення ефективності, безпеки та прозорості фінансових операцій та транзакцій. У зв'язку з цим, метою дослідження є аналіз існуючих підходів, виявлення особливостей та розробка рекомендацій щодо підвищення безпеки та надійності для визначення факторів аналізу та впровадження розподілених технологій в інформаційних облікових системах, що пов'язані з фінансовими структурами та установами, а також вивчення економічного впливу технології розподіленого реєстру на сферу фінансів та визначення переваг та наслідків її впровадження. Зокрема, метою роботи є розширення загальних принципів і вдосконалення методологій мережевої взаємодії та пірінгових протоколів, які наразі недостатньо досліджені через замалу існуючу базу знань.

Об'єктом дослідження є процес розробки інформаційних управляючих систем з використанням технології розподіленого реєстру.

Предметом дослідження є моделі та технології, що базуються на використанні децентралізованих облікових систем та блокчейн протоколів у сфері надання фінансових послуг.

В тому числі, у межах об'єкту дослідження будуть визначені інструментальні засоби та методи криптографії для удосконалення та вирішення проблем безпеки та надійності сучасних економічних систем.

Для досягнення мети роботи необхідно вирішити наступні основні завдання:

– провести огляд технології розподіленого реєстру та розгляд основних принципів роботи блокчейнів, їх протоколів та алгоритмів;

– виявити різні можливі варіанти застосування блокчейну у фінансовій сфері, включаючи криптовалюти, цифрові активи, смарт-контракти та інші фінансові інструменти та розгляд варіантів успішного впровадження;

– зробити аналіз переваг використання технології розподіленого реєстру, такі як доступність, незалежність, захищеність та швидкість транзакцій;

– виявити потенційні недоліки та ризики впровадження технології розподіленого реєстру у фінансову сферу, такі як регуляторні аспекти, масштабованість, конфіденційність даних тощо;

– здійснити порівняння з традиційними архітектурними та обліковими підходами у фінансовій сфері та оцінити ефективність та доцільність впровадження децентралізованих технологій;

– удосконалити криптографічні методи забезпечення цілісності та автентичності даних;

– розробити та описати прототип децентралізованої інформаційної управляючої системи обліку фінансів;

– надати рекомендації та пропозиції для організацій фінансової діяльності стосовно впровадження технології розподіленого реєстру.

Результат дослідження буде представляти собою розробку та опис прототипу децентралізованої інформаційної управляючої системи обліку фінансів, що побудована на базі використання смарт-контрактів разом з захищеним сховищем даних у вигляді ланцюжків блоків (on-chain) та розподілених баз даних (off-chain).

За допомогою отриманих результатів можна визначити простір та потенціал для забезпечення застосування кардинально нових підходів до проектування автоматизованих інформаційних систем у вигляді децентралізованих інфраструктур, з усуненням ризиків надмірності та надлишковості механізмів програмного забезпечення та гарантуванням конфіденційності даних користувачів та клієнтів у відкритих та прозорих структурах.

Вирішення даних аспектів може стати основою для подальших стратегічних рішень та розробки інноваційних фінансових продуктів на базі блокчейн-технологій.

2 ОСОБЛИВОСТІ ВИКОРИСТАННЯ ТЕХНОЛОГІЇ РОЗПОДІЛЕНОГО РЕЄСТРУ ПРИ РОЗРОБЦІ ІС В СФЕРІ ФІНАНСІВ

2.1 Структура та вимоги до формування блоків та транзакцій у розподіленому реєстрі

Методологія розробки інформаційних управляючих систем у сфері фінансів з використанням технології блокчейн базується на концепції децентралізованого та розподіленого обміну фінансових даних у вигляді транзакцій. Для того, щоб забезпечити надійну передачу даних, перш за все, встановлюється мережа реєстру, у якій кожен вузол мережі має можливість перевіряти, зберігати та підтверджувати усі інформаційні потоки, що беруть участь у фінансовій діяльності. Для досягнення консенсусу щодо дійсності стану системи, використовуються ефективні алгоритми на основі обчислювальних ресурсів – Proof-of-Work або грошової частки володіння – Proof-of-Stake. Але, для того, щоб впровадити захищену, надійну, прозору та автоматизовану фінансову інфраструктуру, яка буде здатна покращити спосіб взаємодії в глобальних фінансових операціях, необхідно визначити формат та методи запису та зберігання даних, які зроблять неможливим або важким модифікацію, злам або маніпулювання системою.

Отже, процес реалізації ІТ-проєктів у сфері фінансів, базується на основі використання технології розподіленого реєстру, що представляє собою побудовану за певними правилами структуру, яка зберігає публічні записи транзакцій у безперервній послідовності блоків, у кількох базах даних, які пов’язані між собою у вигляді однозв’язного списку, у мережі, з’єднаній через однорангові вузли. При цьому, кожен наступний блок підтверджує цілісність попереднього, що гарантує, що він був створений на базі всієї попередньої історії.

Для того, щоб усі учасники облікової системи перебували у синхронізованому стані та були згодні з остаточною історією інформаційних потоків, дані щодо кожної активності, дії, операції або процесу групуються у блоки. Блок – це впорядкований набір транзакцій. Кожен блок окрім транзакцій містить у

собі криптографічно зашифроване значення попереднього блоку та додаткову службову інформацію, які необхідні для забезпечення цілісності і незмінності введених даних, а також для роботи самого ланцюжка блоків [10].

Структура блоків наведена на рисунку 2.1

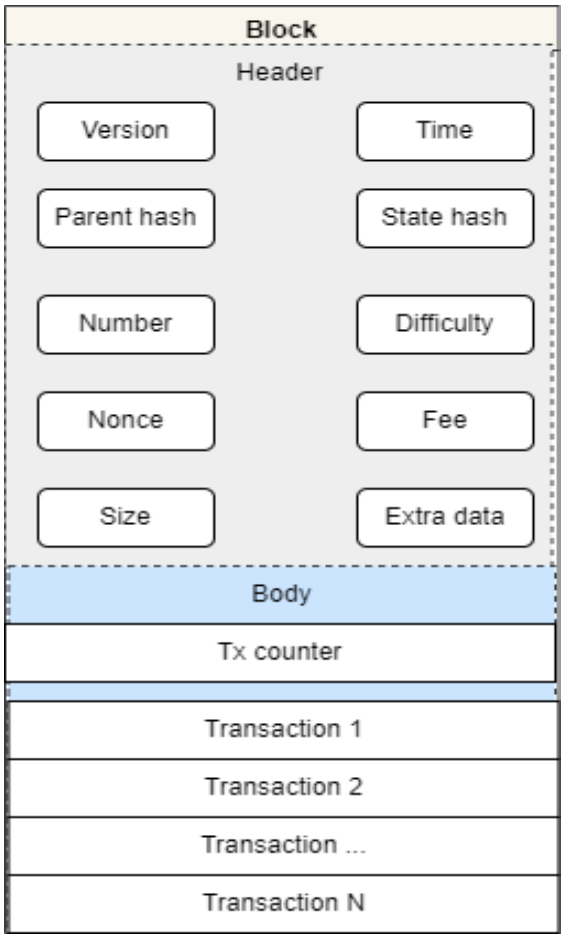

Рисунок 2.1 – Загальна структура блоку у блокчейн

Блок даних розподіляється на дві концептуальні частини: заголовок (header) та тіло (body). Заголовок складається з наступних 10 полів.

1. Version – поточна версія програмного забезпечення бази даних, число.

2. Time – час, коли блок був сформований та перевірений, дата у форматі ДД:ММ:РРРР ГГ:ХХ:СС.

3. Parent hash – унікальний ідентифікатор для попереднього блоку в блокчейні, 256-бітний геш попереднього заголовка блоку.

4. State hash – унікальний ідентифікатор для усього стану системи. Містить облікові баланси, код договору і лічильник транзакцій, 256-бітний геш.

5. Number – загальна довжина ланцюжку блоків у блокчейні, число.

6. Difficulty – перелік навантаження на систему для формування 1 блоку, кількість транзакцій на секунду, 32-бітне число.

7. Nonce – геш-значення, необхідне для доведення, що блок пройшов перевірку на доказ роботи, 32-бітне число.

8. Fee – мінімальний розмір комісії, що потрібна для проведення однієї транзакції, яка буде включена до даного блоку, 32-бітне число.

9. Size – розмір блоку у кілобайтах.

10. Extra data – додаткові дані, необхідні для уточнення стану блоку (валідатор, винагорода, адреси, тощо).

Тіло блоку надає інформацію кількість транзакцій, які включені до блоку та перелік кожної транзакції. Транзакція – це криптографічно підписані інструкції операції збереження даних у блокчейні, під час якої відбувається передача активів або іншої інформації між користувачами мережі [10].

Структуру транзакції наведено на рисунку 2.2

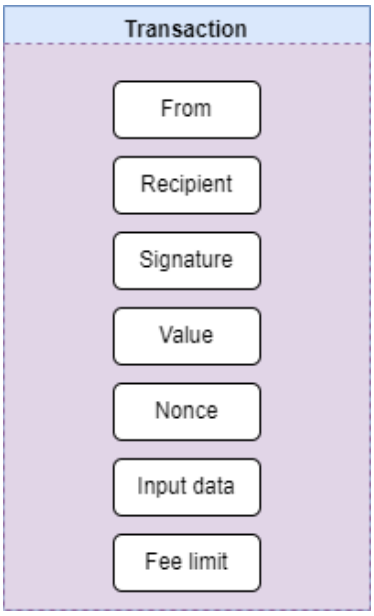

Рисунок 2.2 – Загальна структура транзакції у блокчейні

Транзакція складається з наступних компонентів.

1. From – адреса відправника, який буде підписувати транзакцію. Представляє зовнішній обліковий запис користувача, оскільки контрактні облікові записи не можуть надсилати транзакції, 256-бітне геш-значення відкритого ключа.

2. Recipient – адреса отримувача. Якщо це акаунт користувача, транзакція здійснює обмін активами з іншим користувачем. Якщо це акаунт смарт-контракту (договору), тоді транзакція виконає код або умови контракту, 256-бітне геш-значення відкритого ключа.

3. Signature – цифровий підпис відправника. Генерується коли закритий ключ відправника підписує транзакцію та підтверджує, що відправник авторизував наступну транзакцію. Складається з 3 полів: r, s, v, у якій r та s – це геш-значення підпису закритого ключа, а v – числовий ідентифікатор відновлення відкритого ключа, необхідного для перевірки достовірності даних користувача.

4. Nonce – послідовно зростаючий лічильник, який демонструє номер операції з рахунку. Використовується для того, щоб запобігти заміні або повторі повідомлень, які можуть ініціювати повторний вивід коштів з рахунку.

5. Input data – додаткове поле для включення довільних даних. Визначає корисне навантаження виконання смарт-контрактів, двійковий код у форматі ABI.

6. Fee limit – максимальна кількість комісії, яка може бути використана транзакцією. Технологія розподіленого реєстру динамічно визначає одиницю комісії, що необхідна для кожного кроку обчислення даних на основі технології mempool.

Транзакції, що змінюють стан інформаційної системи, транслюються публічно на всю мережу. Будь-який вузол може транслювати запит на виконання транзакції в мережі. Після того, як це станеться, валідатор зможе виконати транзакцію та поширити результуючі зміни до решти учасників мережі.

На рисунку 2.3 наведено схему формування ланцюжку блоків у розподіленому реєстрі.

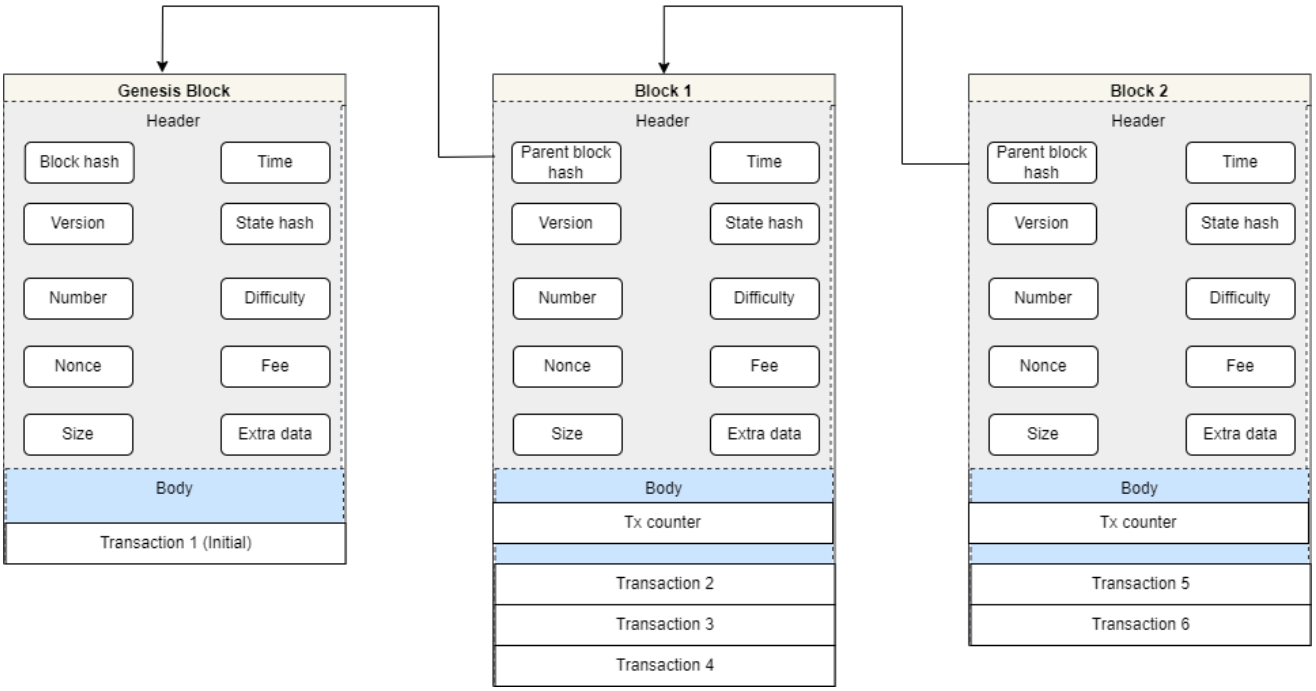

Рисунок 2.3 – Схема формування ланцюжку блоків у блокчейні

Отже, кожен блок представляє собою окрему структуру запису інформації щодо фінансових операцій в окрему базу даних, в якій наступний блок пов’язаний з попереднім за унікальним ідентифікатором, який розраховується за допомогою гешування даних усіх полів – криптографічного перетворення масиву вхідних даних в бітовий рядок фіксованої довжини, значення якого зберігається у полі “Parent block hash”.

Процес отримання геш-значення блоку складається з наступних кроків.

1) Аналізуємо значення кожного параметру заголовку блоку: версія блоку, геш-значення попереднього блоку, геш-значення кореню дерева Меркля, поточна мітка часу, значення складності видобутку блоку та розмір блоку.

2) Якщо значення представлено у шістнадцятковій системі – залишаємо, якщо у десятковій або двійковій – кодуємо у шістнадцятковий запис у форматі зворотного порядку байтів (little-endian). Такий формат є сумісним з багатьма ядрами операційних систем та ефективним з точки зору обробки даних.

3) Для кожного значення відкидаємо префікс 0x та проводимо конкатенацію в одне єдине значення.

4) Проводимо повторне гешування отриманого значення для забезпечення криптостійкості та консистентності інформації.

Продемонструємо обчислення геш-значення блоку відповідно до наступних даних у форматі JSON. У якості геш-функції використовуємо алгоритм Keccak, оскільки на даний момент він офіційно є практично стійким до потенційних вразливостей.

1. Прибираємо з геш-значення попереднього блоку (parent hash) префікс 0x: 6fde2a401de40203e1181e512ffbb59aefa8298cf177e80156c9dcbb03ef1b34.

2. Кодуємо версію блоку (version) з десяткового числа 1 у шістнадцяткове число: Hex(1) = 0001.

3. Прибираємо зі значення кореню транзакцій (state hash) префікс 0x:

a701b656f9560fb7e85675c12732303e1f6ae29e32457efd1d46bf0976e62bf9.

4. Кодуємо час формування блоку (time) з символьного рядку формату “дд-мм-рр гг:хх ” у шістнадцяткове число у форматі little-endian:

Hex(«16-08-23 22:00») = 31362D30382D32332032323A3030.

5. Кодуємо складність формування блоку (difficulty) з десяткового числа 5000 у шістнадцяткове число у форматі little-endian: Hex(5000) = 1388.

6. Кодуємо загалий розмір блоку (size) з десяткового числа 500 у шістнадцяткове число у форматі little-endian: Hex(500) = 1F4.

7. Конкатенуємо кожне значення та отримаємо одне єдине значення:

sum = parent hash + version + state hash + time + difficulty + size =

6fde2a401de40203e1181e512ffbb59aefa8298cf177e80156c9dcbb03ef1b340001a701b656f9560fb7e85675c12732303e1f6ae29e32457efd1d46bf0976e62bf931362D30382D32332032323A303013881F4.

8. Гешуємо отримане значення у 256-бітний формат

Keccak(sum) = caa797351540b09229f12f7658fe764390d71d9cf6df0531bd493306db56afd0.

Отже геш-значення блоку відповідно до значень заголовку, наведених на рисунку 2.4 є caa797…56afd0.

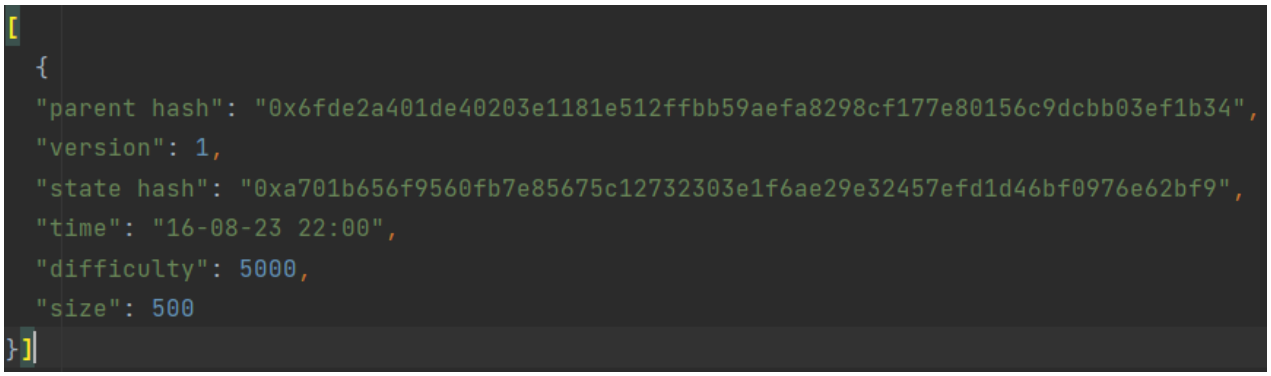

Рисунок 2.4 – Фрагмент вхідних даних для генерації унікального ідентифікатору блоку

Такий формат збереження даних для унікального ідентифікатору має ряд переваг в порівнянні з використанням цілочислового значення, а саме:

– стійкість до колізії першого роду. Визначається твердженням, що припустимо не можна підібрати повідомлення, геш-значення якого повністю збігалося б з іншим повідомленням. В такому випадку для цього знадобиться необмежена кількість ресурсів та 2^{256} ітерацій для виконання алгоритму зламу;

– стійкість до колізії другого роду. Визначається фактом відсутності ефективного зворотного алгоритму, тобто не можна відновити текст по відомому геш-значенню за розумний час. Навіть якщо вдасться отримати текстове значення з гешу, не можна бути повністю впевненим, що початкове вихідне значення було саме таким (NP-задача) [11];

– швидкодія. Кожний криптографічний алгоритм гешування має високу пропускну здатність за рахунок ефективних бінарних та унарних операцій циклічного зсуву, виключного або, кон’юнкції та диз’юнкції та використання невеликої кількості апаратних ресурсів (оперативної пам’яті). Складність таких алгоритмів становить O(n);

– лавинний ефект. Досягається тим, що при щонайменшій зміні повідомлення його геш-значення змінюється до невпізнання. Потенційний зловмисник не має можливості зробити припущення щодо метаданих вхідної

інформації, при цьому покладаючись тільки на вихідні дані геш-значення. В такому випадку вихідне повідомлення залишиться цілісним та достовірним.

Таким чином, наступний блок поєднується з попереднім, пов’язуючи кожну транзакцію даних в єдиний реєстр, що надає змогу проводити швидкісний пошук та сортування в залежності від бізнес-контексту завдання аналізу даних. Проте, варто звернути увагу, що у ланцюжку блоків повинен бути нульовий блок, який є закодованим у програмне забезпечення інформаційної системи та не має посилань на інші блоки – він слугує відправною точкою історії облікової системи та містить лише 1 транзакцію, в якій запропоновано наступні правила та регламент формування транзакцій та блоків у мережі:

– загальний інтервал формування одного блоку складає 1-10 хвилин;

– якщо при формуванні блоку його розмір перевищує 1000 кілобайтів або число транзакцій становить більше 10-ти, тоді автоматично створюється додатковий блок з метою оптимізації даних;

– час нового блоку має бути більшим, ніж медіанне значення часу попередніх 5 блоків;

– час нового блоку має бути меншим, ніж медіанне значення години всіх підключених вузлів + 1 година;

– транзакція може бути включена до блоку, якщо різниця часу створення транзакції та часу створення блоку менше або дорівнює загальному інтервалу формування блоку;

– якщо два блоки створюються одночасно, то першим до ланцюжка блокчейну потрапляє менший за розміром, якщо кількість транзакцій однакова, тоді потрапляє той, в якому залучено більше коштів або активів.

2.2 Особливості реалізації та впровадження цифрового підпису у розподілений реєстр за допомогою спрямованого шифрування

Важливу роль для побудови комунікаційних каналів зв’язку в інформаційних системах, що використовують технологію розподіленого реєстру, відіграють системи цифрового підпису. Цифровий підпис – це криптографічний механізм, який використовується для перевірки достовірності та цілісності цифрових даних [12]. Під визначенням достовірності розуміється послуга безпеки, що представляє доказ відправки транзакції певним справжнім користувачем, під визначенням цілісності – факт того, що дані транзакції (відправник, отримувач, сума надісланих коштів) не було змінено під час передачі. Цифровий підпис можна також розглядати як цифрову версію звичайних рукописних підписів, але з вищим рівнем складності та безпеки. При роботі з фінансами цифровий підпис перш за все необхідний для доказу володіння активами. Власник активів, у результаті, визначається знанням особистого ключа, який потрібний для обчислення цифрового підпису. На практиці дуже важливо, щоб схеми формування ключів та цифрового підпису працювали коректно, а механізми забезпечення безпеки були надійними. Для власника ключа важливо, щоб ніхто інший не міг вкрасти, розрахувати чи генерувати такі самі ключові дані. Зокрема, у системі фінансів це має серйозне значення особливо для процесів формування звітностей, укладання договорів, контрактів, юридичних та інших фінансових документів. За допомогою схем цифрового підпису існує можливість уникнення потенційних ризиків при складанні угод між багатьма сторонами, які не мають повної довіри один до одного. Цифровий підпис гарантує наявність інформації про участь визначеної особи, яка не зможе у майбутньому заперечувати своє відношення до деякої діяльності. За замовчуванням, цифровий підпис не розголошує особисті дані кожної особи публічно, такі дані є конфіденційними та зашифрованими, проте для систем, які мають зв’язок з особою, що здійснює фінансові операції, є можливість перевірити

цифровий підпис щоб отримати метаінформацію задля коректної аутентифікації користувача у обліковій системі.

На рисунку 2.5 наведено UML діаграму станів для процесу створення цифрового підпису.

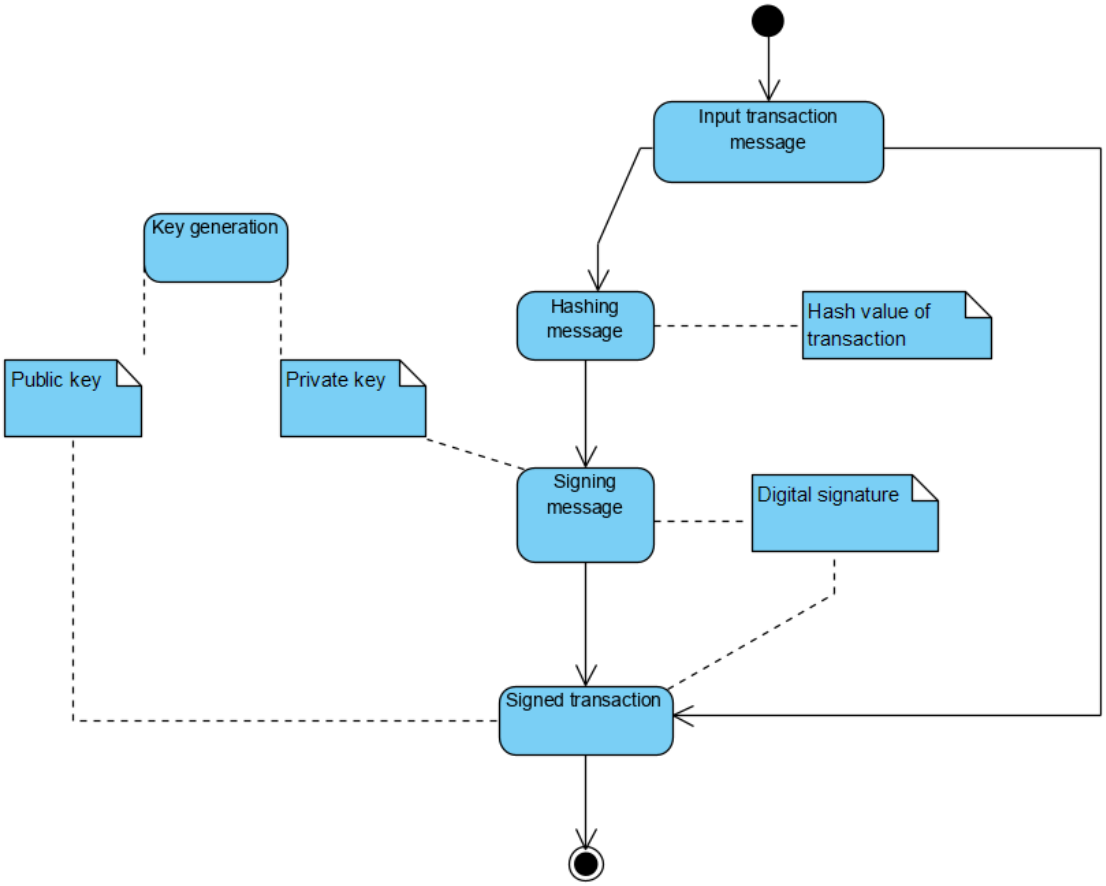

Рисунок 2.5 – Діаграма станів створення цифрового підпису

У контексті технології розподіленого реєстру, схема створення цифрових підписів найчастіше складається з таких етапів, як гешування повідомлення, генерація ключової пари та створення підпису. Першим кроком є гешування даних. Виконується за рахунок обробки інформації за допомогою алгоритму гешування для генерації безпосередньо дайджесту повідомлення. Незважаючи на те, що повідомлення можуть значно відрізнятися за своїм розміром, проте після проведення гешування всі хеші володітимуть однаковою довжиною. Також такий

підхід дозволить спростити процес обробки інформації. Наступним кроком, який може виконуватись паралельно з гешуванням транзакції – це генерація ключової пари. Ключова пара – це структура, що складається з двох ключів: закритого ключа (private key) та відкритого ключа (public key) [12]. Ці ключі створюються разом та є комплементарними по відношенню один до одного. Закритий ключ – ключ, відомий лише своєму власнику. Тільки збереження власником у таємниці свого закритого ключа гарантує неможливість підробки зловмисником документа та цифрового підпису від імені завіряючого. Такий ключ створюється через апаратні або програмні генератори випадкових чисел та повинен представляти собою просте число, що не має ентропії (джерела випадковості) та не може повторитись знову при розрахунку на одному і тому ж алгоритмі. Для того, щоб закритий ключ був надійним та безпечним, запропоновано наступні вимоги:

– довжина ключа повинна сягати на менше 256 біт (32 байти) та мати розмір кратний 2^n;

– значення ключа повинно задовольняти умові: $0 < d < q$, де d – значення ключа, q – просте число, що вказується в загальносистемних параметрах;

– при кодуванні ключа у двійковий код найбільша послідовність одиничних або нульових бітів повинна сягати не менше 36 бітів;

– при кодуванні ключа у двійковий код загальна кількість одиниць є приблизно рівною загальній кількості нулів у послідовності та мати похибку не більше 5 %;

– при кодуванні ключа у двійковий код та розділенні послідовності бітів на k блоків, кожної довжиною m та кількість однакових блоків становить від 1 до 1.5 * p (p – довжина ключа у байтах), значення яких розраховується за формулою:

$$X = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i^2\right) - k, \tag{2.1}$$

де X– довжина всієї тестової послідовності;

k – кількість однакових блоків в послідовності;

m – довжина блоків;

i – ітератор проходження одного блоку;

n_i – кількість зустрічання блоку Поккера в послідовності.

Відкритий ключ – ключ, який може бути опублікований у сертифікаті та використовується для перевірки справжності підписаного документа, а також для запобігання шахрайству з боку запевняючої особи у вигляді її відмови від підпису документа. Відкритий ключ виконує роль адреси (рахунку), на яку пересилаються цифрові фінансові активи. Є доступним для публічного перегляду кожному користувачу. За допомогою відкритого ключа виконується процес шифрування задля створення транзакції. Відкритий ключ розраховується на основі закритого шляхом проведення математичних перетворень, детальний опис яких буде наведено нижче.

Після проведення гешування повідомлення та генерації ключової пари виконується процес підписання повідомлення. Функція підпису приймає на вхід закритий ключ та геш-значення повідомлення та в результаті перетворення повертає цифровий підпис. Дані цифрового підпису запропоновано кодувати у форматі ASCII85, що наразі є найбільш ефективним форматом кодування двійкової інформації у текстову з показником 80 %. Алгоритм цього формату побудований так, що його реалізацію можна легко імплементувати на різних мовах програмування, особливо на тих, що мають потужні інструменти для багатопотоковості, за рахунок ізоляційних операцій, які займають невелику кількість пам’яті та обчислювальних ресурсів комп’ютеру. Незважаючи на те, що такий формат є не зовсім зручним для запису, бо містить у таблиці підстановок спеціальні символи екранування, можна стверджувати, що декодування такого значення не займає багато часу.

Варто також зазначити, що цифрові підписи безпосередньо взаємопов'язані зі змістом кожного повідомлення. Таким чином, на відміну від рукописних підписів, які є детермінованими незалежно від контексту документа, кожне повідомлення з цифровим підписом має нелінійність обчислення за рахунок різного цифрового ідентифікатору підпису [12].

На рисунку 2.6 наведено UML діаграму станів для процесу перевірки цифрового підпису.

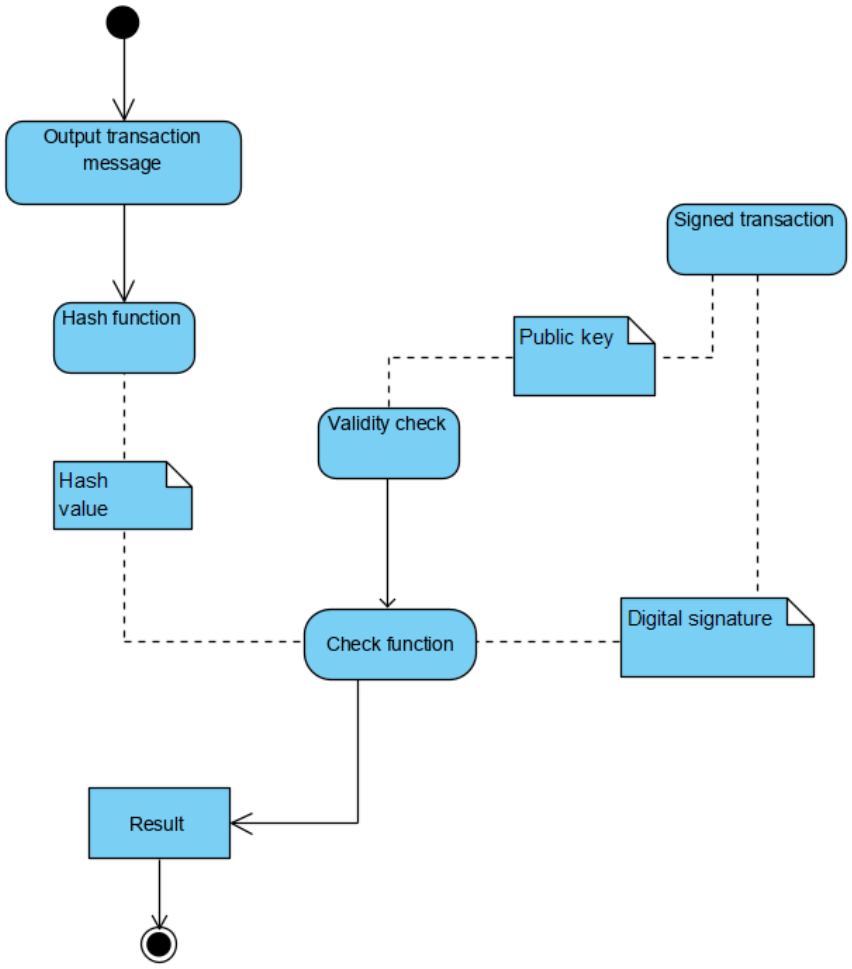

Рисунок 2.6 – Діаграма станів перевірки цифрового підпису

Отже, перевірка цифрового підпису відбувається наступним чином:

– система отримує повідомлення та цифровий підпис відповідно до транзакції відправника;

– система використовує відкритий ключ відправника для розшифрування цифрового підпису за необхідності. В результаті розраховується геш-значення повідомлення;

– алгоритм порівнює геш-значення повідомлення зі значенням, отриманим з цифрового підпису. Якщо вони збігаються, тоді система підтверджує факт того, що

повідомлення було відправлено певною особою та не було змінено в дорозі (статус True). Інакше – статус False.

Для того, щоб лише система мала можливість перевіряти цифровий підпис відправника, пропонується використовувати алгоритм спрямованого шифрування повідомлення. Спрямоване шифрування є формою криптографічного захисту, де дані шифруються таким чином, що вони можуть бути розшифровані лише одним певним отримувачем. Мета такого асиметричного перетворення полягає в тому, що інформація шифрується на відкритому ключі одержувача, або на ключі, отриманому з допомогою відкритого ключа, а розшифровується на секретному ключі одержувача або на ключі, отриманому за допомогою секретного ключа. Наразі найбільш стійким алгоритмом спрямованого шифрування вважається алгоритм цифрового підпису на еліптичних кривих, оскільки задача логарифмування на кубічній кривій є більш складною, ніж завдання дискретного логарифмування для кінцевого поля.

Алгоритм цифрового підпису, заснований на еліптичних кривих (ECDSA) є криптографічною схемою для побудови цифрових підписів з використанням рівняння кривої:

$$y^2 = x^3 + ax + b, \tag{2.2}$$

Параметри a та b в формулі (2.2) можуть бути довільними числовими значеннями, але рекомендується брати параметри кривої secp256k1 (a = 7, b = 0), на відміну від інших кривих форматів brainpool та sect, оскільки вона активно використовується у найпопулярніших блокчейнах Bitcoin та Ethereum, є добре вивченою і вважається безпечною. Також крива такого формату має відносно невеликий розмір ключів, що робить її ефективною для використання у програмах з обмеженими ресурсами.

При використанні еліптичної кривої secp256k1 для знаходження відкритого ключа шляхом скалярного множення базової точки на велике натуральне число важливою властивістю є те, що ця операція є незворотною, тобто не можна

вирахувати закритий ключ, якщо відомо дані відкритого ключа, оскільки операції ділення точки на число не існує для групи точок еліптичної кривої.

2.3 Використання смарт-контрактів при реалізації ІС у фінансовій сфері

Для того, щоб інформаційна система, яка використовує технологію розподіленого реєстру у сфері фінансів, мала можливість до автоматизованого виконання угод між сторонами та здатність до забезпечення достовірного результату для всіх користувачів ІТ-проєкту, при цьому позбуваючись участі посередників або втрати часу, рекомендується використовувати смарт-контракти. Смарт-контракт – це незмінні децентралізовані комп'ютерні програми, інтерфейси або протоколи транзакцій які розміщені у мережі та виконуються детерміновано в контексті певного блокчейну. Вони призначені для автоматичного виконання, контролю або документування подій і дій відповідно до умов договору або угоди. Код смарт-контракту складається з набору інструкцій, що визначає заздалегідь визначені умови, виконання яких призводить до змін стану облікової системи [13]. Завдяки тому, що смарт-контракти працюють розподілено та не мають централізованого сервера, вони дозволяють багатьом учасникам процесу досягти спільного результату в точний, своєчасний і захищений від втручання спосіб та позбутися вразливості до окремих точок атаки зловмисників. При застосуванні у багатосторонніх цифрових угодах, програми смарт-контрактів можуть зменшити ризик посередників, підвищити ефективність, знизити витрати та забезпечити новий рівень прозорості процесів.

Життєвий цикл процесу розробки та використання смарт-контракту у розподіленому реєстрі складається з наступних етапів:

– смарт-контракт проектується, реалізовується з використанням мови програмування та компілюється в байт-код для зменшення обсягу пам'яті;

– смарт-контракт ретельно тестується та перевіряється з використанням автоматизованого або мануального тестувань;

– смарт-контракт впроваджується у мережу за допомогою спеціальної транзакції, у якому визначається його публічна адреса у вигляді геш-значення;

– користувачі безпосередньо взаємодіють зі смарт-контрактом напряму шляхом подання транзакцій на виконання функції, що визначена в смарт-контракті або через децентралізовані додатки (DAPP). Смарт-контракт, у свою чергу, може взаємодіяти з іншими контрактами, що знаходяться в блокчейні.

За необхідності, є можливість знищити смарт-контракт з метою економії ресурсів.

Однією з головних переваг впровадження смарт-контракту при реалізації ІТ-проєктів у сфері фінансів – композиційність. Це принцип, який полягає у поєднанні різних компонентів для створення нових систем. У контексті комп'ютерних наук, смарт-контракт може розглядатися як публічний прикладний програмний інтерфейс (API), з яким кожен може взаємодіяти або інтегрувати у власні інформаційні системи для розширення функціональності. Композиційність смарт-контрактів зазвичай базується на трьох принципах:

– модульність. Це здатність окремих компонентів виконувати певне завдання. Тобто кожен смарт-контракт має певний власний варіант використання;

– автономність. Складові компоненти повинні мати можливість працювати незалежно. Кожен смарт-контракт повинен виконуватися самостійно і функціонувати, не покладаючись на інші частини системи;

– інкапсуляція. Розробники не можуть викликати або виконувати зовнішні смарт-контракти або інтегрувати їх в якості бібліотек та інтерфейсів у власні програми, якщо вони недоступні для всіх [13].

Зважаючи на зазначені положення, можна стверджувати, що розробка смарт-контракту має невеликий життєвий цикл, оскільки смарт-контракти здебільшу є публічними та мають відкритий вихідний код, що означає підтримку спільноти у вигляді великої кодової бази. Також можна зазначити, що використання смарт-контракту здатна привносити інноваційні рішення для рішення складних проблем

автоматизації. Взаємодія між компонентами системи розподіленого реєстру через смарт-контракти покращує досвід користування (UX). Користувачі можуть отримати доступ до більшої функціональності у випадку коли прикладні програми інтегрують зовнішні смарт-контракти, аніж у монолітній системі, де програми не можуть спілкуватися.

Проте, доцільно буде зазначити про обмеження, що стають на шляху, зважаючи на те, що смарт-контракти власне не можуть отримати інформацію про події «реального світу», оскільки вони не можуть отримати дані з джерел поза мережею. Це означає, що за задумом, вони не мають властивості до самостійного реагування на зовнішні події. Покладання на зовнішню інформацію може поставити під загрозу консенсус, який важливий для безпеки та децентралізації. Однак для ІТ-проєктів, що реалізовані за підтримки технології розподіленого реєстру, важливо мати можливість використовувати дані як “on-chain”, так і “off-chain”. Безкомпромісним рішенням є використання оракулів, які є програмними каналами, які приймають дані поза мережею та роблять їх доступними для смарт-контрактів. Ще одним обмеженням смарт-контрактів є максимальний розмір контракту. Смарт-контракт може мати максимум 1 Мб, інакше не можливо буде створити транзакцію (проблема закінчення газу). Але, це можна вирішити, використовуючи шаблон проектування “Diamond”.

2.4 Характеристика бізнес-процесів некомерційних фінансових організацій (фондів) та їх реалізація з використанням технології розподіленого реєстру

У рамках задачі дослідження та використання технології розподіленого реєстру при реалізації ІТ-проєктів у сфері фінансів розглядається фінансова сфера діяльності з бізнес-доменом, який являє собою фонд – некомерційну організацію, яка започаткована громадянами та юридичними особами на основі добровільних майнових внесків, що переслідує благодійні, культурні, освітні чи інші соціальні,

суспільно корисні, зокрема економічні цілі [14]. З точки зору фінансової сфери, фонд можна визначити як підприємство, яке займається управлінням фінансовими активами, наданням фінансових послуг та здійсненням фінансових операцій. Тобто, фонд може представляти собою як організацію для проведення благодійних зборів, так і компанію, що займається фінансуванням та запуском IT-стартапів.

Діапазон бізнес-операцій, що включено до такої бізнес-галузі, може включати:

– управління активами. Фонд може займатися управлінням інвестиційними портфелями клієнтів, включаючи купівлю, продаж та управління акціями, облігаціями, пайовими фондами та іншими фінансовими інструментами;

– інвестиційні послуги. Фонд може надавати різноманітні інвестиційні послуги, такі як консультування з інвестування, портфельний менеджмент, аналіз ринків, розрахунок ризику та управління інвестиційними портфелями;

– позикові послуги: Фонд може здійснювати позикову діяльність, надаючи кредити підприємствам та приватним особам, встановлюючи відсоткові ставки, страхування, тощо;

– фінансовий аналіз та аудит: Фонд може надавати послуги з фінансового консультування та розробки фінансових стратегій для клієнтів, допомагаючи їм управляти ризиками, планувати бюджет витратами та доходами.

Зокрема, при функціонуванні фонду можуть знаходитися зовнішні бізнес-сутності, такі як банки, інвестори, повноважені особи, з якими фонд може мати безпосереднє відношення, що визначається за регламентованими правилами та вимогами.

На рисунку 2.7 наведено IDEF0 діаграму бізнес-процесів фонду при використанні технології розподіленого реєстру.

Для реалізації бізнес-процесів та розв’язання задач відповідно до постановки завдання на реалізацію IT-проєкту у сфері фінансів можна використати наступні методи:

– смарт-контракти для автоматизації складання договорів для формування, управління та надання інформації про статус та поточний стан системи;

– розподілений реєстр для зберігання інформації про бізнес-становище у вигляді ланцюжку блоків;

– криптографія для забезпечення конфіденційності даних та захисту від несанкціонованого доступу;

– мережева взаємодія з визначенням характеристики та типу вузлів та алгоритмів консенсусного та виконавчого рівнів для збереження стану облікової системи на апаратному пристрої кожного співробітника компанії для усунення недоліку єдиної точки відмови.

– алгоритм досягнення консенсусу Proof-Of-Authority.

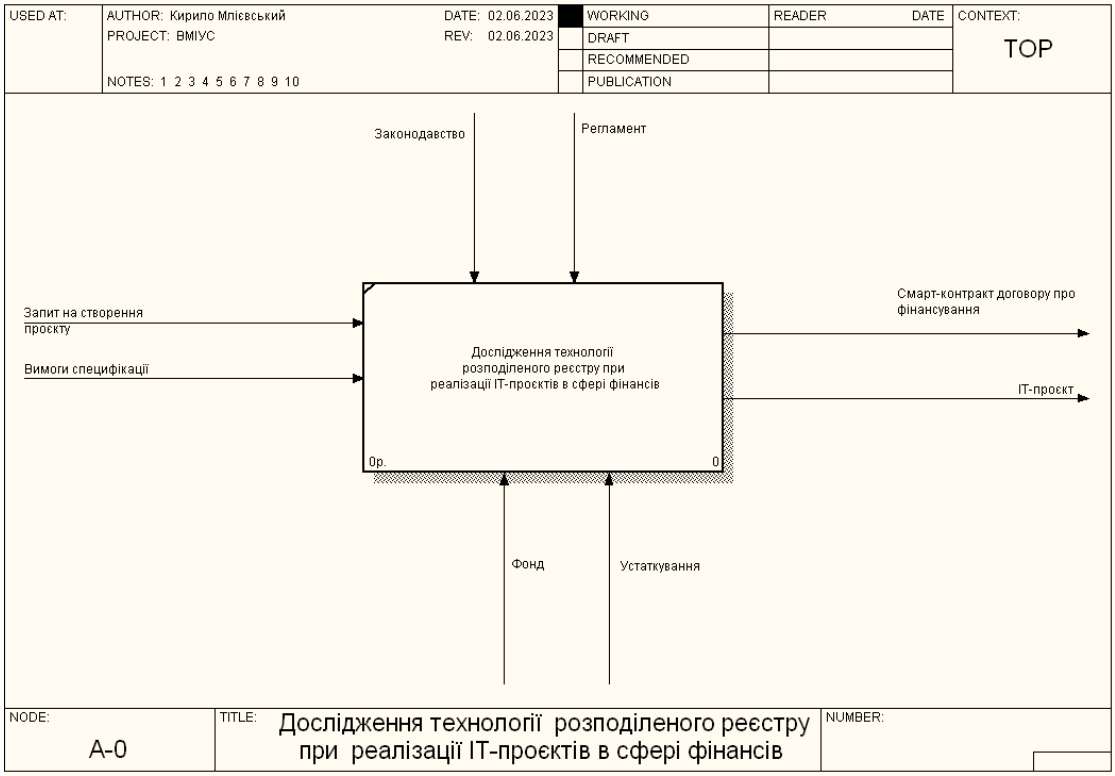

Рисунок 2.7 – IDEF0 діаграма бізнес-процесів фонду

Для проєктування та реалізації смарт-контрактів пропонується використовувати Solidity. Solidity — це об’єктно-орієнтована мова високого рівня для побудови програмного забезпечення у розподіленому реєстрі. Має статичну

типізацію, підтримує успадкування, бібліотеки та складні типи, визначені користувачем, серед інших функцій [15].

Для впровадження цифрових підписів пропонується використовувати криптографічний алгоритм ECDSA.

Для визначення унікальності набору даних з метою цілісності даних пропонується використовувати математичний алгоритм гешування.

Для організації системи управління та моніторингу бази даних пропонується використовувати блокчейн.

Запропонована методологія буде здатна підтримуватися системою та її розробка зможе покращити або змінити підхід до таких елементів зі сфери економіки як послуги, модель фінансів та доходів та ін.

Отже, головний бізнес-контекст, що підтримується фондовою організацією у сфері фінансів містить інноваційні пропозиції до впровадження, які засновані на принципах децентралізації, що повинно представляти собою розділення функцій та повноважень, доступну комунікацію між сторонами за допомогою смарт-контрактів, масштабованість даних за допомогою технології блокчейн, ефективність та резистентність до вторгнень та вразливості з боку шахраїв за допомогою криптографії, прозорість та достовірність. За рахунок виключення посередників разом присутня тенденція до скорочення загальних витрат на проведення діяльності бізнесу, а також можливість щодо викорінення корупції, яка існувала в посередницьких структурах залучених в рамках діяльності.

3 ЕКСПЕРЕМЕНТАЛЬНА ПЕРЕВІРКА МОЖЛИВОСТЕЙ ВИКОРИСТАННЯ ТЕХНОЛОГІЇ РОЗПОДІЛЕНОГО РЕЄСТРУ НА ПРИКЛАДІ РОЗРОБКИ ІС БЛАГОДІЙНОГО ФОНДУ

3.1 Аналіз діяльності благодійного фонду для оцінки можливостей впровадження технології розподіленого реєстру

У якості предметної області для дослідження та застосування технології розподіленого реєстру у сфері фінансів пропонується використати інформаційну систему публічного благодійного фонду.

В загальному випадку, можна дати наступне визначення організації такого типу. Фонд – комерційна або некомерційна установа, яка може бути заснована громадянами або юридичними особами, основою фінансування якої є добровільні або попередньо обговорені та визначені внески задля соціальних, благодійних, дослідних, культурних чи освітніх цілей. За різновидами діяльності фонди можуть поділятися на інвестиційні або благодійні [14, 16].

Інвестиційний фонд представляє собою фінансовий акумулятор, який об'єднує, концентрує фінансові потенціали групи осіб де вартість інвестування цільових об'єктів розподіляється поміж всіх них. Залежно від спеціалізації, такі фонди можуть інвестувати в акції, цінні папери, криптоактиви тощо. Їхня основна мета – професійне управління коштами клієнтів та їх подальше інвестування з метою одержання прибутку. Такі фонди також можуть поділятися за формою доступу – відкриті або закриті; за структурою активів – диверсифіковані або недиверсифіковані; за терміном впровадження – термінованими або безстроковими [16].

Благодійний фонд – це організація, яка створена для збору коштів, предметів або засобів на благодійні цілі, наприклад пожертвування або гранти [16]. Головне завдання такого фонду полягає у підтримці інших організацій, громад або окремих осіб, які потребують допомоги у фінансуванні, підтримці чи обізнаності. Благодійний фонд зазвичай звільняється від податків і отримує фінансування з

різних джерел, в залежності від типу та формату діяльності. Тому, можна виділити наступні різновиди благодійних фондів, а саме:

– публічні;

– приватні;

– громадські або суспільні;

– корпоративні;

– незалежні;

– грантові.

В ході подальшого дослідження розглядатиметься структура та функціонування публічного благодійного фонду. Він уособлює організацію, яка в основному надає гранти та отримує підтримку від громадської спільноти, тобто отримує фінансування з кількох джерел, як-от окремі громадяни, приватні фонди та державні установи. Основними напрямками, за якими працює фонд, є: гуманітарна допомога; ліки; допомога дітям; евакуація людей; допомога ЗСУ тощо.

На рисунку 3.1 представлено схему організаційної структури публічного благодійного фонду.

На схемі, що зображена на рис. 3.1 можна побачити, що у фонду міститься елементарна організаційна структура, яка представляє собою трирівневий розподіл. Створена організаційна структура містить 4 підрозділи. Вищим органом управління благодійного фонду є рада – колегіальний орган, який відповідає за обрання виконавчого та управляючого органів, затвердження напрямків та програм благодійної діяльності та прийняття ключових рішень. Президент фонду відповідає за загальне керівництво фонду, натомість виконавчий директор відповідальний за щоденне керівництво фондом, реалізацію стратегії та прийняття управлінських рішень. Адміністративний відділ відповідає за найм або пошук проєктів, що потребують фінансування. Виконавчий відділ – команда, яка відповідає за кожен окремий проєкт та його доцільне супроводження та містить власний штаб співробітників, таких як координатори, волонтери, аудитори тощо. Фінансовий відділ відповідає за облік коштів та майна фонду, а також за підготовку звітності.

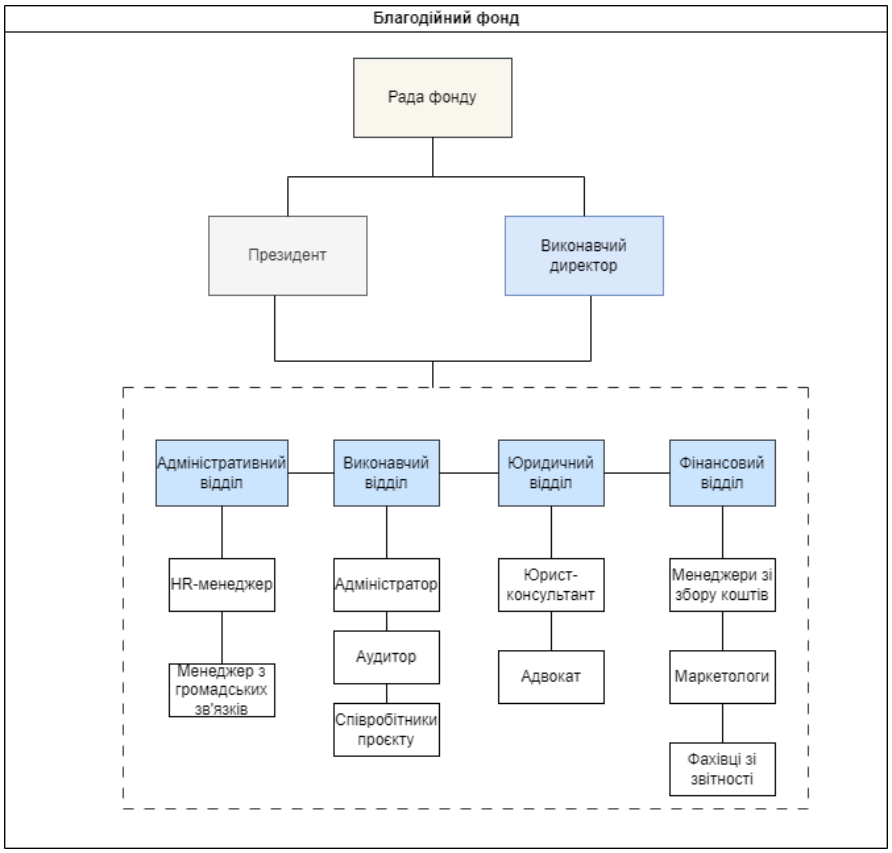

Рисунок 3.1 – Організаційна структура благодійного фонду

Виходячи з отриманої інформації, можна припустити, що існує певний ряд недоліків в організації та управлінні фондом. Серед них – недостатня ефективність використання ресурсів, непрозорість фінансового управління та низька ступінь довіри з боку осіб, що здійснюють внески, відсутність стандартів та регулювання, конфлікт інтересів тощо. Це все може призвести до неконтрольованого або недоцільного витрачання коштів, браку прозорості інформаційної системи, дорогих витрат на проведення адміністрування та аудиту.

Для того, щоб вирішити вищенаведені проблеми, пропонується впровадити в ІС благодійного фонду технологію розподіленого реєстру, яка буде побудована на основі децентралізованих програм – смарт-контрактів, які зможуть покращити ведення звітності та документації, зменшити або поліпшити діяльність адміністративного та фінансового відділів та запровадити певні стандарти регулювання та валідації за допомогою алгоритмів. Також, наявність блокчейн

протоколів та механізмів консенсусу зможе надати перспективу до поліпшення організаційної структури у вигляді перерозподілу в однорівневу горизонтальну модель, в якому рада та керуюча гілка осіб матиме вплив на рівні з іншим персоналом та співробітниками.

3.2 Опис інформаційної технології розподіленого реєстру при розробці ІС благодійного фонду

Розглянемо графічну нотацію (IDEF0 діаграму), що призначена для формалізації та опису бізнес-процесів діяльності благодійного фонду при впровадженні технологій розподіленого реєстру та блокчейн протоколів, які пропонуються в ході дослідження. В ній розглядаються логічні відношення між процесами та даними фонду, без урахування часової послідовності. Основними елементами даної інформаційної моделі є діаграми, що містять у собі блоки, що представляють головні функції фонду, дуги, що символізують набір обмежень, умов, правил та показують взаємозв’язки між блоками [17].

На рисунку 3.2 наведено діаграму IDEF0 першого рівня декомпозиції функціональної структури використання технології розподіленого реєстру при реалізації ІС для фондової організації.

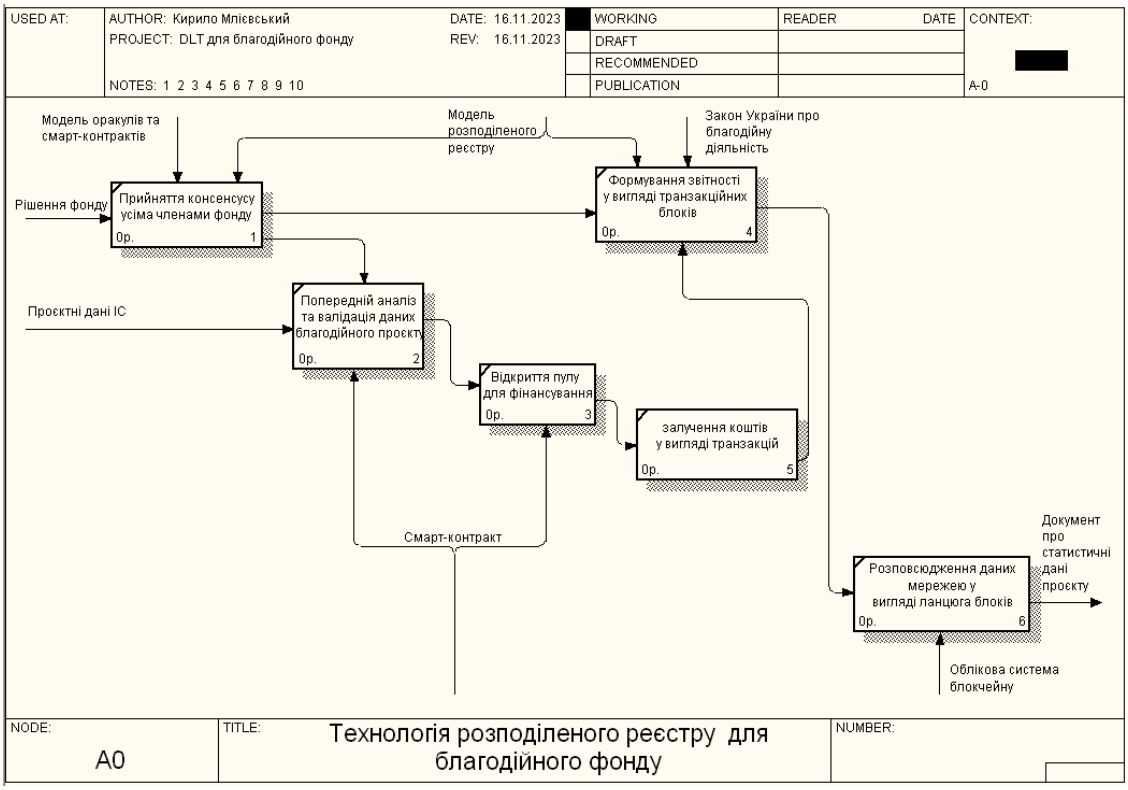

Рисунок 3.2 – Діаграма першого рівня декомпозиції використання технології розподіленого реєстру при розробці ІС благодійного фонду

Відповідно до рисунку 3.2, можна зазначити що вхідною інформацією для процесу ведення проєкту благодійного фонду є рішення фонду щодо організації та планування діяльності відповідні проєктні дані ІС, що необхідні для визначення місії та цілі фонду. У якості механізмів, що надають змогу до впровадження технології розподіленого реєстру є смарт-контракти та облікова система блокчейну, які певним чином замінюють використання традиційних систем супровіду, таких як база даних або сервер. Також варто зазначити, що за рахунок використання моделі оракулів та розподіленого реєстру зникає безпосередня необхідність в послугах осіб, що займаються менеджерською або адміністративною діяльністю. В тому числі, на систему накладаються певні регуляторні обмеження, а саме Закон України «Про благодійну діяльність та благодійні організації». Отже, початковим процесом є попередній аналіз та валідація даних для організації благодійного проєкту. Він реалізується за рахунок

прийняття спільного рішення усіма співробітниками фонду стосовно структури, вимог та потреб проєкту відповідно до визначеної галузі та стратегії. Після цього відбувається відкриття пулу у вигляді смарт-контракту для створення програми для збору коштів відповідно до стверджених даних проєкту. Після відкриття пулу вже відбувається залучення коштів бажаючих в якості джерел фінансування у вигляді транзакцій, задля забезпечення фіксації акту передачі даних особами або іншими організаціями. В результаті отримання коштів формується звітність про гроші, що були переказані, у вигляді блоків транзакцій, в яких також за бажанням зможе бути наведена додаткова інформація стосовно дати переказу, осіб, що брали участь, або цільове призначення. Після того, як блок сформований, він прив'язується до інших блоків, формуючи ланцюжок даних у вигляді зв'язкового списку та надсилається усім учасникам мережі, що необхідно для збереження повної історії діяльності фонду та захист від несанкціонованих атак та зламів системи.

3.3 Впровадження технології розподіленого реєстру при розробці ІС благодійного фонду

Впровадження технології розподіленого реєстру (DLT) у сферу фінансів при розробці інформаційних систем для благодійного фонду має величезний потенціал завдяки використанню вузлів та смарт-контрактів. Вузли замінюють та забезпечують адміністративне функціонування та будують надійну децентралізовану інфраструктуру фонду. Їх основною перевагою є спілкування та обмін один з одним для трансляції в мережу та отримання необхідного консенсусу для відображення та перевірки нових блоків даних відповідно до правил організації. Вузли мають вирішальне значення для цілісності та безпеки системи, вони перевіряють транзакції на переказ коштів та запобігають маніпуляціям і шахрайству, зберігають копії реєстру. Кожен вузол унікальний та має власне призначення.

На рисунку 3.3 наведено DFD-діаграму використання та взаємодії вузлів для забезпечення використання технології розподіленого реєстру у діяльності благодійного фонду.

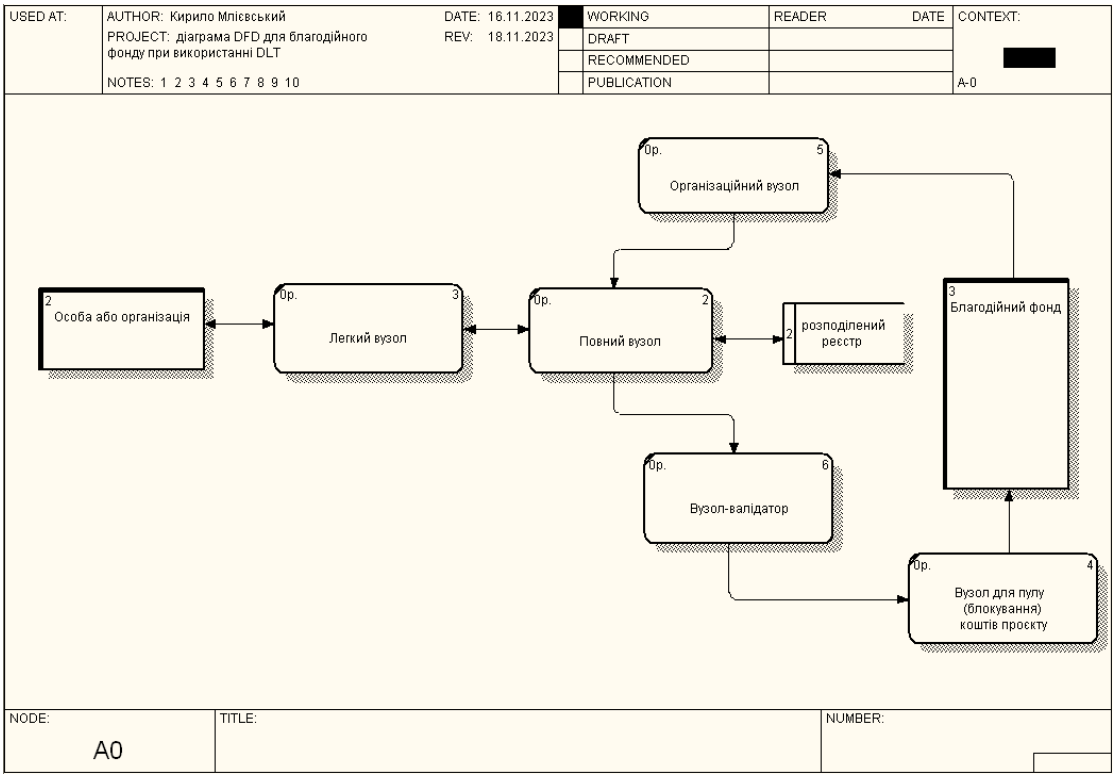

Рисунок 3.3 – Діаграма взаємодії вузлів для використання технології розподіленого реєстру при розробці ІС благодійного фонду

Діаграма потоків даних (DFD) – це стандарт та методологія моделювання потоків інформації (даних) між роботами та об'єктами, що надає інформацію про вихідні і вхідні дані кожного об'єкта і самого процесу (роботи). Вона являє собою ієрархію функціональних підсистем та сутностей, пов'язаних між собою потоками даних. В даному випадку, в якості функціональних процесів виступають вузли, оскільки вони беруть участь в обміні даними одночасно між фондом та особами або організаціями, за допомогою яких власне відбувається операційна діяльність фонду. Головна мета такого представлення – виділити функціональні відношення між програмними процесами системи [17]. На діаграмі також представлені наступні компоненти.

1. Зовнішні сутності (джерела або отримувачі даних). Такими об’єктами є фонд та особи та організації, які безпосередньо взаємодіють з системою та між собою. Фонд відповідає за формування вимог та правил щодо дії та реалізації проєкту.

2. Накопичувачі дані. Відповідно до діаграми, у якості механізму, у якому зберігаються дані, є розподілена реєстр (книга або блокчейн), в якому дані формуються в блоки, утворюючи ланцюг блоків у хронологічній послідовності усіх операцій та взаємодій з системою, зокрема з можливістю до розділення ланцюгів даних в залежності від формату та типу проєкту.

3. Системи та підсистеми. У вигляді систем виступають вузли, які відповідають за автоматизацію та дотримання основних принципів децентралізації.

4. Потоки даних та процесів (стрілки).

Функціональні блоки системи діяльності благодійного фонду з використанням технології розподіленого реєстру можна декомпозувати на підсистеми, що будуть пов’язані з головними процесами. Також можливо визначити додаткові накопичувачі даних, які призначені для зображення та збереження тимчасової або часто модифікованої інформації, яку туди можливо у будь-який момент помістити або витягти незалежно від реалізації.

Для того, щоб впровадити технологію розподіленого реєстру використовуються наступні типи вузлів:

– легкий вузол. Уособлює в собі облегшену варіацію повного вузла для спрощеної перевірки платежів, використовується для початкового з’єднання з особою, що бажає здійснити грошовий переказ. Є швидкими та ефективними, проте мають обмежену пам’ять та обчислювальну потужність. Цей тип вузла призначений для швидкої та простої обробки транзакцій та оснащений лише основними даними, необхідними для роботи та не завантажують повну версію розподіленого реєстру, а тільки заголовки даних;

– повний вузол. Зберігає повну копію реєстру та є найважливішим типом вузла в мережі інформаційної системи фонду. Будучи частиною однорангової мережі, повний вузол надсилає або збирає певні дані особи або організації від

легкого вузла за запитом та може незалежно від інших вузлів перевіряти блоки даних транзакцій. Має можливість до видалення частини історії записів, задля економії пам’яті ІС та у випадку, якщо інформація перестає бути актуальною. Саме він забезпечує стабільність системи;

– організаційний вузол. Це вузол, щоб був схвалений фондом та особами, що ним керують. Використовує механізм підтвердження повноважень, що представляє собою затвердження з застосуванням інструкцій до інших вузлів, що беруть участь в автоматизації діяльності організації та якими керують співробітники фонду;

– вузол для пулу (блокування). Використовуючи надіслані кошти як заставу, консенсусна модель визначає повноваження аутентифікації учасникам, які виконали заздалегідь обумовлені показники, такі як внесення певної кількості грошей у проєкт. Вузол стейкінгу може складатися з транзакцій щодо переказів одного або декількох осіб в єдиний пул, що об’єднує всі кошти, щоб забезпечити механізм блокування та мати краще управління ресурсів з метою вилучення переказів на рахунки інших нецільових осіб, оскільки пул представляє собою смарт-контракт;

– вузол-валідатор. Перевіряє транзакції на потенційні помилки або вразливості та за необхідністю може модифікувати дані транзакцій або додавати додаткову метаінформацію, поки транзакція не збереглася у мережі.

Технологія смарт-контрактів дозволяє виконувати усі необхідні фінансові операції, а також автоматизувати та захищати їх через безпечний, надійний та безперервний механізм. Смарт-контракти, що базуються на технології блокчейну, представляють собою програми, які автоматично виконують умови контракту при виконанні певних умов, без потреби в посередниках або інших осіб. У благодійному фонду вони можуть бути використані для автоматизації ряду операцій, таких як переказ коштів, виконання умов проєкту, відслідковування та управління активами тощо.

Для визначення структури даних системи на рисунку 3.4 наведено UML-діаграму класів використання технології розподіленого реєстру у діяльності благодійного фонду.

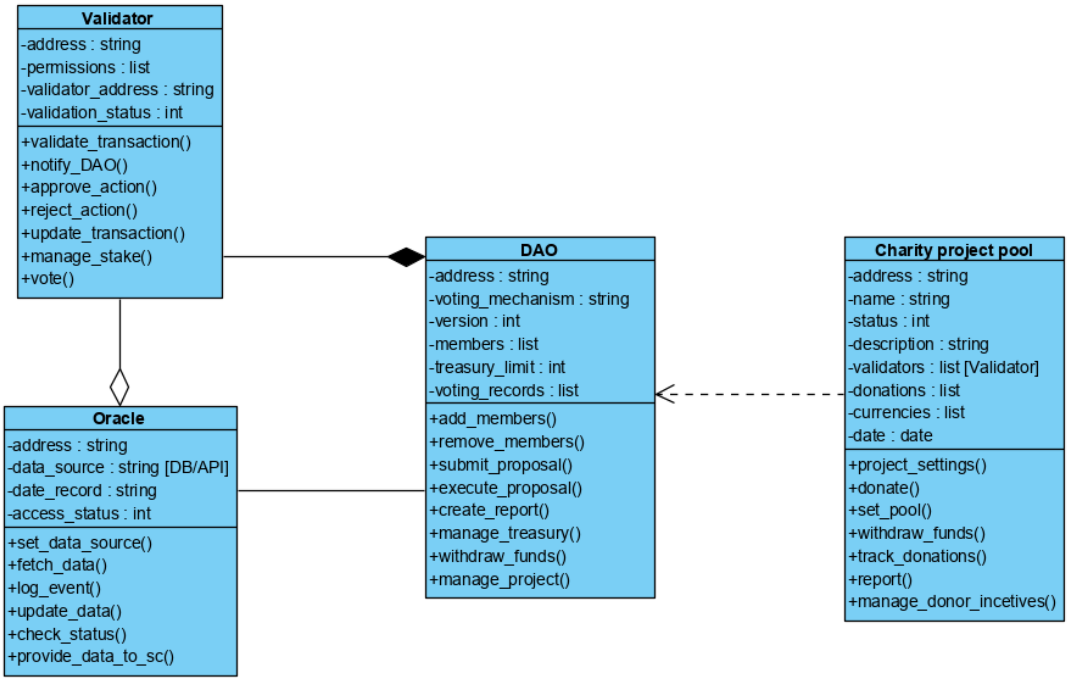

Рисунок 3.4 – Діаграма класів використання технології розподіленого реєстру при розробці ІС благодійного фонду

На рисунку 3.4 представлено діаграму, яка демонструє загальну структуру смарт-контрактів системи, що по суті представляють собою їєрархію класів у розподіленому реєстрі та їх кооперацій, атрибутів, методів, інтерфейсів і взаємозв'язків [18]. Ця діаграма класів показує основні компоненти системи, що беруть участь та мають вплив на технічну діяльність благодійного фонду. Вони мають зв’язки між собою та атрибути, що містять та зберігають інформацію, визначену у обраній предметній області. Головним об’єктом є децентралізована автономна організація фонду (ДАО) та містить методи для налаштування середовища компанії та прийняття остаточних рішень щодо змін, управління проєктами тощо.

На UML діаграмі класів було визначено наступні типи структур даних смарт-контрактів:

– ДАО смарт-контракт. Децентралізована автономна організація фонду (ДАО) керується за допомогою унікального смарт-контракту. У ДАО немає традиційної ієрархії, вона працює прозоро, згідно з прописаними в програмному коді правилами та рішеннями системи управління установи. Дозволяє колективно приймати рішення щодо змін у роботі системи благодійного фонду в цілому або конкретного проєкту, що ним фінансується. За середовищем виконання є єдиним централізованим з-поміж усіх;

– смарт-контракт оракул. Це децентралізовані програми, які отримують, перевіряють і передають внутрішню інформацію іншим смарт-контрактам. Оракул контролюється єдиною організацією – фондом, який відповідає за агрегування інформації поза мережею та оновлення даних контракту оракула за запитом. Є ефективним, оскільки покладаються на одне єдине джерело істини та працюють у випадках, коли пропрієтарні набори даних публікуються безпосередньо власником із визнаним підписом. Набори даних можуть представляти собою дані про проєкт або рішення-консенсус співробітників фонду;

– смарт-контракт валідатор. Є аналогом оракулу але за винятком того, що він отримує зовнішню інформацію, що стосується транзакцій та перевіряє їх на наявність помилок або вразливостей. Забезпечує неможливість виникнення різного роду атак, таких як атака Фінні (атака подвійної витрати), екліпс атакою, XSS атакою тощо;

– смарт-контракт для пулу проєкту. Це контракти, що виконують роль автоматизації та керують основною бізнес-логікою та операціями проєкту благодійного фонду, тобто збіром донатів та пожертвувань від окремих осіб та підприємств в єдиний пул, що представляє собою особистий рахунок проєкту. Він також забезпечує зовнішній зв’язок між особами, які користуються системою і технологією блокчейн та її розподіленим реєстром.

3.4 Опис реалізації технології розподіленого реєстру при розробці ІС благодійного фонду

Основні структурні елементи системи розподіленого реєстру з використанням криптографічних алгоритмів, алгоритмів цифрового підпису та смарт-контрактів складається з сервісів – модулів, робота яких забезпечує діяльність благодійного фонду можна представити наступним чином.

1. Сервіс формування блоків та ланцюжку блоків фонду.

```
328  class Blockchain(Block):
329      coinDatabase = dict()
330      blockHistory = []
331      txDatabase = []
332      faucetCoins = 0
333      blocks = []
334
335      def __init__(self):
336          self.__class__.blocks.append(self)
337
338      @staticmethod
339      def initBlockchain():
340          genesis_block = Block()
341          genesis_block.createBlock(setOfTransaction=[genesis_transaction], prevHash=None)
342          return genesis_block
343
344      def createBlock(self, setOfTransaction, prevHash):
345          block = Block()
346          block.prevHash = prevHash
347          block.setOfTransaction.extend(setOfTransaction)
348          block.blockID = hex(abs(hash(str(self.block_counter) + str(len(setOfTransaction)) + str(prevHash))))[:33]
349          Block.block_counter += 1
350          return block
351
352      def validateBlock(self, block):
353          if len(block.prevHash) == 0:
354              raise Exception('Block prevhash do not exist or match')
355          elif block.setOfTransaction in self.txDatabase:
356              raise Exception('Block transactions already in txDatabase')
357          elif block.setOfTransaction.setOfOperations.amount > max(self.coinDatabase.values()):
358              raise Exception('Operations amount more than account balance!')
```

Рисунок 3.5 – Фрагмент коду для сервісу формування блоків

На представленому фрагменті блоку відображено три головні функції, що забезпечують роботу блокчейну. Функція initBlockchain забезпечує початкову ініціалізацію та запуск системи з першого блоку – genesis, який містить тільки одну транзакцію, яка задає умови та правила, що стосуються діяльності фонду. Наступна

функція – createBlock, що призначена для створення блоків з даними про транзакції, тобто перекази коштів. Функція validateBlock виконує роль перевірки блоків даних на консистентність та нормалізацію, яка була визначена у підрозділі 2.1.

2. Сервіс для цифрового підпису користувачів.

Цифрові підписи відіграють важливу роль у забезпеченні цілісності, безпеки та прозорості системи благодійного фонду. Використовуючи цифрові підписи, благодійні фонди можуть захистити свою операційну діяльність, підвищити безпеку компанії за рахунок моніторингу цілісності транзакцій, зміцнити довіру між особами та фондом, оптимізувати адміністративні задачі та відповідати нормативним вимогам та обмеженням законодавства.

На рис. 3.6 наведено фрагмент коду для створення цифрового підпису за рахунок застосування алгоритму, який базується на використанні еліптичних кривих.

```
class ECDSA(ElipticCurve.ElipticCurve):
    curve: str = None
    private_key: int = None
    public_key: ElipticCurve.EcPoint = ElipticCurve.EcPoint()
    base_point: ElipticCurve.EcPoint = ElipticCurve.EcPoint()
    signature: list

    def __init__(self):
        super().__init__()
        private_key = ElipticCurve.ec.generate_private_key(ElipticCurve.ec.SECP256K1(), ElipticCurve.default_backend())
        public_key = private_key.public_key()
        self.curve = public_key.public_numbers().curve.name
        self.base_point.x = public_key.public_numbers().x
        self.base_point.y = public_key.public_numbers().y
        self.private_key = 0
        self.public_key.x = 0
        self.public_key.y = 0

    def key_generation(self):
        private_key = ECDH.generate_private_key()
        self.private_key = private_key.private_numbers().private_value
        self.public_key.x = ECDSA.transformIntToHex(ECDH.compute_public_key(private_key).x)
        self.public_key.y = ECDSA.transformIntToHex(ECDH.compute_public_key(private_key).y)
        return self.private_key, self.public_key

    def sign_message(self, message):
        h = int(Keccak256(message), 16)
        k = random.randint(1, ECDH.P - 1)
        ec = ElipticCurve.ElipticCurve()
        r = ec.scalar_EC_point(k, self.base_point)
        s = (pow(k, -1) * (h + r.x * self.private_key)) % self.p
        self.signature = [r.x, s]
        return self.signature
```

Рисунок 3.6 – Фрагмент коду генерації цифрового підпису

Ключовими функціями є: key_generation, яка генерує публічний та приватний ключі користувача; sign_message, що виконує математичні обчислення для підпису повідомлення з використанням приватного ключа та verify_signature, яка перевіряє підпис за допомогою публічного ключа. В даному випадку важливим є те, що перевірити підпис може будь-яка особа, яка є співробітником фонду, а підписати повідомлення – тільки користувач, що формує транзакцію для переказу коштів на рахунок благодійного фонду. Головними компонентами для генерації цифрового підпису є крива – secp256k1 та алгоритм для гешування – Keccak256, який вважається дійсним стандартом гешування інформаційних систем.

```python
class ElipticCurve:
    EcPoint: EcPoint
    curve: ec.SECP256K1
    p: int
    formula: bool

    def __init__(self):
        self.EcPoint = EcPoint()
        self.curve = ec.SECP256K1()
        self.a = 0
        self.b = 7
        self.p = pow(2, 256) - pow(2, 32) - 977

    def generate_EC_point(self) -> EcPoint:
        private_key = ec.generate_private_key(self.curve)
        public_key = private_key.public_key()
        self.EcPoint.x = public_key.public_numbers().x
        self.EcPoint.y = public_key.public_numbers().y

        return (self.EcPoint.x, self.EcPoint.y)

    def is_on_curve(self):
        if (self.EcPoint.y ** 2) % self.p == (self.EcPoint.x ** 3 + self.a * self.EcPoint.x + self.b) % self.p:
            return True
        else:
            return False

    def add_EC_points(self, a: EcPoint, b: EcPoint) -> EcPoint:
        if a != b:
            m = (b.y - a.y) / (b.x - a.x)
            self.EcPoint.x = m ** 2 - a.x - b.x
            self.EcPoint.y = m * (a.x - self.EcPoint.x) - a.y
```

Рисунок 3.7 – Фрагмент коду формування еліптичної кривої

3. Сервіс смарт-контрактів для збору коштів, аутентифікації валідаторів, створення регламенту роботи фонду.

Сервіс для збору коштів та пожертвувань насамперед забезпечує виконання головної бізнес-логіки інформаційної системи благодійного фонду. Саме тому було прийнято рішення використовувати програмне забезпечення на основі смарт-контрактів, оскільки в першу чергу додатки, що написані на базі смарт-контрактів можуть легко інтегруватися у мережу розподіленого реєстру. Після того, як додаток туди потрапляє, усі дані стосовно коду конвертуються у байт-код машинного рівня, тим самим забезпечує полегшення та прискорення процесу роботи програми, що є особливо важливим у разі високого завантаження.

```
contract CharityProjectPool {
    address public charityEmployee;
    uint256 public donationPool;
    mapping(address => uint256) public donations;

    struct Donation {
        address donor;
        uint256 amount;
    }

    event DonationMade(address indexed donor, uint256 amount);

    constructor() {    362040 gas 337400 gas
        charityEmployee = msg.sender;
    }

    modifier onlycharityEmployee() {
        require(msg.sender == charityEmployee, "Only employee of foundation can perform this action");
        _;
    }

    function setPool(uint256 _amount) external onlycharityEmployee {    24687 gas
        donationPool = _amount;
    }

    function donate() external payable {    infinite gas
        require(msg.value > 0, "Donation amount should be greater than zero");

        donationPool += msg.value;
        donations[msg.sender] += msg.value;

        emit DonationMade(msg.sender, msg.value);
    }

    function trackDonations(address _donor) external view returns (uint256) {    2885 gas
```

Рисунок 3.8 – Фрагмент коду, що реалізує смарт-контракт для збору коштів для благодійного проєкту

На рисунку 3.8 представлені функції налаштування пулу коштів, які необхідно зібрати, функція внесення коштів з перевіркою значення на невід'ємність та функцію для відстеження донатів за певною особою. Для функції налаштування загального значення пулу коштів використано модифікатор onlycharityEmployee, що означає, що тільки співробітник фонду може змінювати це значення.

4. Сервіс для створення вузлів.

Під сервісом для створення вузлів розуміється певний модуль, який виконує роль посередника між транзакціями користувачів та обліковою системою розподіленого реєстру благодійного фонду. Вузол, в свою чергу, представляється у вигляді будь-яких пристроїв, такого як телефони, планшети, ноутбуки, ПК, які може виконувати алгоритмічні обчислення для перевірки даних та відправки їх по мережі блокчейну. Завдяки цьому, усувається пряма необхідність у використанні потужного та обчислювального центру – серверів, що вже зменшує витрати на облаштування апаратного забезпечення. Пропонується, що у кожного співробітника буде власний вузол-валідатор та за потреби він може встановити цей вузол на своєму пристрої для забезпечення швидкого переказу.

5. Сервіс для прийняття консенсусу за допомогою оракулів.

Для реалізації механізму прийняття консенсусу визначено програму-оракул, що за допомогою API викликів буде збирати інформацію кожного співробітника фонду щодо рішень, які стосуються питань діяльності благодійного проєкту або фонду в цілому. Такий підхід дозволить, у децентралізованому форматі, досягати спільної угоди щодо подальших змін установи.

Варто відмітити, що схема для голосування є наступною: якщо x% голосів підтримує якусь пропозицію, а 0% не підтримує (утримались від голосування), то вважається, що беззаперечно всі підтримали пропозицію (x=100%). В тому числі, якщо всі співробітники проголосували за або проти та ніхто не утримався, тоді застосовується правило простої більшості: 50%+1 голос за прийняття рішення.

Отже, для того, щоб усі вищенаведені та описані сервіси мали змогу працювати та обмінюватися даними, пропонується використовувати мікросервісну

архітектуру інформаційної системи благодійного фонду, що представляє собою підхід до створення додатків у вигляді набору незалежно розвернутих сервісів, які є децентралізованими та розробляються незалежно від іншого [19]. Насамперед, дане рішення випливає через використання різних за програмною логікою сервісів, що написані на декількох мовах програмування, таких як Python, Solidity, Golang тощо. Зокрема, мікросервіси надають змогу до масштабування системи при умовах інтеграцій зі сторонніми сервісами та утилітами, що важливо особливо під час заходів із збору коштів або кампаній, де робоче навантаження може відрізнятися в різних аспектах системи. Оскільки у такому випадку кожен сервіс діє модульно, тобто незалежно від інших, ризик до виникнення помилок у всій інформаційній системі зменшується, що зможе забезпечити захищеність від повного збою та роботу інших сервісів за необхідністю. Також, такий архітектурний підхід ізолює важливі дані за рахунок принципу поліморфізму окремих процедур та аргументів за приватними, публічними та захищеними характеристиками.

3.5 Оцінка трудомісткості та тривалості розробки ІС благодійного фонду з використанням технології розподіленого реєстру у порівнянні з клієнт-серверною технологією

На рисунку 3.9 наведено графік прогнозу розміру розподіленого реєстру благодійного фонду.

Рисунок 3.9 – Графік прогнозу розміру розподіленого реєстру

Відповідно до рисунку 3.9, можна зазначити, що прогнозується експоненційне зростання розміру блоків у розподіленому реєстрі фонду, що також збільшує загальний обсяг використовуваної пам’яті. Незважаючи на те, що дані облікової системи зберігаються на кожному пристрої, це все одно стає певним недоліком щодо використання блокчейн технологій для інформаційної системи, що пропонується.

На рисунку 3.10 наведено графік прогнозу середньої завантаженості розподіленого реєстру за рік.

Рисунок 3.10 – Графік прогнозу завантаженості розподіленого реєстру

Дані, що представлені на рисунку 3.10 є прогнозованими відповідно до наразі існуючих реалізованих розподілених реєстрів. Також, причиною залежності у вигляді кривої є змінна кількість користувачів інформаційної системи, кількість активних проєктів тощо.

Для проведення розрахунків оцінок якості, трудомісткості та тривалості розробки розподіленого реєстру при реалізації інформаційної системи благодійного фонду використаємо метод функціональних точок.

Метод функціональних точок – стандартний метод вимірювання розміру програмного продукту з точки зору користувачів системи. Цей метод призначений для оцінки на основі логічної моделі обсягу програмного продукту або проекту та кількістю функціоналу [20].

Для визначення сумарної кількості не вирівняних функціональних точок UFP для логічних файлів ILF і файлів зовнішнього інтерфейсу EIF є оцінка їх складності. Оцінка складності розраховується з використанням двох показників:

– RecordElementType (RET) – сутність, яка описувала який-небудь клас програмного забезпечення або ж таблицю бази даних ІС. Оцінюється за шкалою від 1 до 6;

– DataElementType (DET_F) – унікальний атрибут, який використовується для опису класів або таблиць. Оцінюється за шкалою від 1 до 50.

Відповідно до розроблених моделей та схем, орієнтовані значення складності проєктування класів RET складає 5 (середній показник) та значення унікального атрибуту, який використовується для проєктування класів DET складає 21 (високий показник).

Для визначення складності оброблюваних даних внутрішніх логічних файлів f_{ILF} і файлів зовнішнього інтерфейсу f_{EIF} використаємо наступні формули:

$$UFP_i \Leftrightarrow f_{ILF}(DET_F, RET), \quad (3.1)$$

$$UFP_j \Leftrightarrow f_{EIF}(DET_F, RET) \quad (3.2)$$

Якщо раніше було визначено, що значення RET = 5, DET = 21, то складність оброблюваних даних f_{ILF} і f_{EIF} згідно з (3.1) – (3.2) буде дорівнювати середньому показнику.

Оскільки складність оброблюваний даних для внутрішніх логічних файлів Average, то кількість нескоректованих функціональних точок для логічних файлів UFP_i відповідно дорівнює $UFP_i \Leftrightarrow f_{ILF}(21,5) = 10$.

Оскільки складність оброблюваний даних для файлів зовнішніх інтерфейсів Average, то кількість нескоректованих функціональних точок для файлів інтерфейсів UFP_j відповідно дорівнює $UFP_j \Leftrightarrow f_{ILF}(21,5) = 7$.

Для визначення складності ведення транзакцій основними показниками структурної складності використовуються:

– DataElementType (DET_T) – унікальний елемент, який використовується для реалізації транзакції. Оцінюється за шкалою від 1 до 16;

– FileTypeReferenced (FTR) – кількість інформаційних об'єктів які модифікуються або зчитуються в ході реалізації транзакції.

Відповідно до розробки розподіленого реєстру для діяльності благодійного фонду значення FTR = 5, DET_T = 20, отже складність обробки даних транзакцій є високою.

Для визначення значень UFP_k,UFP_l і UFP_mEI, EO і EQ відповідно використаємо наступні функції визначення складності оброблюваних даних:

$$UFP_k \Leftrightarrow f_{EI}(DET_F, FTR), \quad (3.3)$$

$$UFP_l \Leftrightarrow f_{EO}(DET_F, FTR), \quad (3.4)$$

$$UFP_m \Leftrightarrow f_{EQ}(DET_F, FTR) \quad (3.5)$$

Оскільки складність оброблюваний даних f_{EI} та f_{EO} і f_{EQ} високою, то значення нескоректованих функціональних точок UFP для зовнішніх входів і зовнішніх запитів EI і EQ дорівнює $UFP_k \Leftrightarrow f_{EI}(20,5) = 6$, значення нескоректованих функціональних точок UFP для зовнішніх виходів дорівнює $UFP_l \Leftrightarrow f_{EO}(15,3) = 6$, $UFP_m \Leftrightarrow f_{EQ}(15,3) = 7$.

Відповідно до отриманих даних, кількість нескоректованих функціональних балів для системи, що пропонується:

$UFP = \sum_{i=1}^{n_{ILF}} UFP_i + \sum_{j=1}^{n_{EIF}} UFP_j + \sum_{k=1}^{n_{EI}} UFP_k + \sum_{l=1}^{n_{EO}} UFP_l + \sum_{m=1}^{n_{EQ}} UFP_m$ = 10 + 7 + 6 + 6 + 7 = 36

Для врахування впливу загальносистемних характеристик і особливостей проектування в методі функціональних точок, використаємо фактор вирівнювання VAF, кожен параметр якого оцінюється за шкалою від 0 до 5, де 0 – параметр не важливий та 5 – параметр є основною характеристикою системи. Для цього спочатку визначимо значення системних характеристик у таблиці 3.1

Таблиця 3.1 – Перелік системних характеристик

№ п/п	Найменування характеристики	Значення характеристики
1.	Обмін даними	4
2.	Розподілена обробка даних	4
3.	Продуктивність	3
4.	Обмеження по апаратних ресурсів	1
5.	Транзакційна навантаження	4
6.	Інтенсивність взаємодії з користувачем	1
7.	Ергономіка	2
8.	Складність обробки	2
9.	Повторне використання	5
10.	Гнучкість	4

Після отримання значень системних характеристик, знайдемо фактор вирівнювання VAF відповідно до наступної формули:

$$\mathrm{VAF} = (\mathrm{TDI} * 0{,}01) + 0{,}65, \tag{3.6}$$

де $\mathrm{TDI} = \sum_{a=1}^{10} \mathrm{DI_a}$, а $\mathrm{DI_a}$ – дорівнюють значенням системних характеристик з таблиці 3.1.

Згідно з формулою (3.6) та даними з таблиці 3.1 отримаємо:

$\mathrm{TDI} = \sum_{a=1}^{10} \mathrm{DI_a} = 30,$

$\mathrm{VAF} = (\mathrm{TDI} * 0{,}01) + 0{,}65$ = (30 * 0,01) + 0,65 = 0,95.

Оскільки отримане значення є меншим за 1, то можна стверджувати, що загальносистемні параметри щодо розробки та впровадження розподіленого реєстру є припустимими до реалізації.

Розрахуємо кількість скоригованих функціональних точок за наступною формулою:

$$\mathrm{DFP} = \mathrm{UFP} * \mathrm{VAF}, \tag{3.7}$$

Таким чином, отримуємо у відповідності з (3.7): $\mathrm{DFP} = 36 * 0{,}96 = 34{,}2$.

Далі розрахуємо кількість рядків коду на одну функціональну точку:

$$\mathrm{k} = \frac{\mathrm{Low} + 4 * \mathrm{Average} + \mathrm{High}}{6}, \tag{3.8}$$

Тобто з (3.8) отримаємо: $\mathrm{k} = \frac{40+4*50+75}{6} = 52{,}5 \approx 53$.

Після цього розрахуємо кількість рядків коду LOC, яке треба написати для реалізації оцінюваної функціональної задачі:

$$\mathrm{LOC} = \mathrm{DFP} * \mathrm{k}, \tag{3.9}$$

Тоді, з (3.9) отримуємо LOC = 34,2 * 53 = 1795,5 = 179,5 KLOC.

У якості режиму розробки програмного забезпечення визначено сблокований режим – характеризується такими ознаками як не дуже велика команда розробників, наявність деяких інновацій, помірні обмеження та кінцеві строки проекту, нестабільне середовище розробки. Для оцінювання обсягу робіт з розробки програмного забезпечення (трудовитрат) модель COCOMO визначає наступну формулу для сблокованого режиму:

$$\mathrm{E} = 2{,}4 * (\mathrm{KLOC})^{1{,}12}, \tag{3.10}$$

де E – трудовитрати на розробку програмного забезпечення в межах ІС (людино-місяців);

KLOC – кількість тисяч строк коду, що треба написати в процесі розробки програмного забезпечення.

Оскільки було розраховано, що значення KLOC дорівнює 115.680, тоді згідно з (3.10) значення показника E дорівнює:

$$E = 2{,}4 * (KLOC)^{1{,}12} = 2{,}4 * 179{,}5^{1{,}12} = 2{,}4 * 334{,}62 = 803.$$

Для оцінювання тривалості робіт з розробки програмного забезпечення (витрат часу) модель COCOMO визначає наступну формулу для сблокованого режиму:

$$TDEV = 2{,}5 * (E)^{0{,}35}, \quad (3.11)$$

де TDEV – витрати часу на розробку програмного забезпечення в межах ІС (місяці);

E – трудовитрати на розробку програмного забезпечення в межах ІС (людино-місяці).

Оскільки було розраховано, що значення E дорівнює 718, тоді значення показника TDEV згідно з (3.11) дорівнює: $TDEV = 2{,}5 * (E)^{0{,}35} = 2{,}5 * (803)^{0{,}35} = 2{,}5 * 10{,}39 = 25{,}97.$

Знаючи оцінки обсягу робіт та тривалості розробки програмного забезпечення, можна за допомогою моделі COCOMO вирахувати потребу в персоналі та продуктивності праці розробників. Для цього використовуються наступні формули:

$$SS = E/\ TDEV, \quad (3.12)$$

$$P = KLOC/E \quad (3.13)$$

де SS – кількість виконавців ІТ-проекту з розробки програмного забезпечення (людей);

P – кількість тисяч строк коду, які один виконавець ІС повинен писати за місяць;

E – трудовитрати на розробку програмного забезпечення в межах ІС (людино-місяці);

TDEV – витрати часу на розробку програмного забезпечення в межах ІС (місяці).

У відповідності з (3.12) та (3.13) отримаємо:

$$SS = E/\,TDEV = \ 803/25{,}97 \ = \ 30{,}92\ ,$$

$$P = KLOC/\,E = \ 179{,}5/803 \ = \ 0{,}22.$$

Відповідно до проведених розрахунків, для розробки інформаційної системи благодійного фонду з використанням технології розподіленого реєстру витрати на проведення повного життєвого циклу створення системи складають: два роки, 179 тис. рядків коду, 30 співробітників.

За допомогою спрощеного методу функціональних точок визначимо складність існуючих інформаційних систем благодійних фондів із використанням клієнт-серверної архітектури:

$$FP = (C_1 + C_2 + C_3\ )^{2{,}35}, \qquad (3.14)$$

де С1 (масштаб проекту) – програма під замовлення, дорівнює 9;

С2 (користувачі об’єкту проектування) – комерційний проект, дорівнює 9;

С3 (тип об’єкту проектування) – клієнт-серверне ПЗ, дорівнює 8.

Відповідно, згідно з (3.14) отримуємо: $FP = (C_1 + C_2 + C_3\ )^{2{,}35} = \ (9 + 9 + 8)^{2{,}35} = \ 22^{2{,}35} = 2114{,}40.$

Для визначення показника KLOC, виходячи з показника FP (1 KLOC = =1000 LOC), використовується методика відкату. За цією методикою показник KLOC визначається за формулою:

$$KLOC = k * FP, \qquad (3.15)$$

Значення коефіцієнту k обирається з таблиці, виходячи з результату вибору мови програмування або середовища розробки програмного забезпечення.

У якості мови програмування для клієнт-серверного додатку оберемо PHP, які характеризуються в середньому 67 строками коду на функціональний блок (тобто k = 67).

Результат розрахунку значення показника KLOC згідно з (3.15) наступний: $\mathrm{KLOC} = \mathrm{k} * \mathrm{FP} = 67 * 2{,}1144 = 141{,}66$.

Знайдемо трудомісткість розробки функціональної задачі за формулою: $\mathrm{E} = 2{,}4 * (141{,}66)^{1{,}12} = 616{,}03$.

Тривалість розробки функціональної задачі знайдемо за формулою: $\mathrm{TDEV} = 2{,}5 * (616{,}03)^{0{,}35} = 23{,}67$.

Потреби персоналу можна оцінюємо за формулою: $\mathrm{SS} = 616{,}03/23{,}67 = 26{,}02 \approx 26$.

Отже, відповідно до проведених розрахунків, для розробки інформаційної системи благодійного фонду з використанням клієнт-серверної архітектури витрати на проведення повного життєвого циклу створення системи складають: 23 місяці, 141 тис. рядків коду, 26 співробітників.

Отже, відповідно до значень, отриманих для розробки інформаційної системи благодійного фонду з використанням технології розподіленого реєстру, можна визначити, що оцінка обсягу та тривалості робіт при розробці інформаційної системи з використанням клієнт-серверної архітектури трохи нижча, ніж при використанні розподіленого реєстру – менше потенційна складність розробки та реалізації деяких функцій, потреба в персоналі та дещо великий час від проектування до впровадження.

Незважаючи на це, використання технології розподіленого реєстру для реалізації ІС в сфері фінансів має низку переваг:

– вищий рівень захищеності даних від атак за рахунок сервісу цифрового підпису та алгоритмів шифрування/гешування;

– вищий рівень прозорості транзакцій та звітності за рахунок використання публічної децентралізованої мережі;

– вищий фактор довіри серед користувачів за рахунок відсутності централізованого органу або установи;

– можливість швидкого переказу коштів з будь-якого куточка світу;

– відсутність єдиної точки відмови (серверу) та потенційної втрати даних за рахунок використання вузлів та загального реєстру.

Підсумовуючи, в результаті проведеного аналізу та дослідження використання технології розподіленого реєстру в сфері фінансів на прикладі інформаційної системи благодійного фонду, можна сказати, що визначені пропозиції можуть мати інноваційний характер, насамперед за рахунок відсутності наразі існуючих систем подібного типу та використання актуальних та сучасних протоколів та програмного забезпечення у вигляді смарт-контрактів для ведення фінансової адміністративної та операційної діяльностей, тим самим оптимізуючи використання людських та апаратних ресурсів. Проте, дане рішення не є еталонним та має власні обмеження, які можуть впливати на складність інформаційної системи і її обсяг та потребувати розвитку в майбутньому [21].

ВИСНОВКИ

У ході виконання кваліфікаційної роботи було розглянуто усі етапи та аспекти, що пов’язані з дослідженням та використанням технології розподіленого реєстру при реалізації ІС у сфері фінансів. Було проведено аналіз предметної області та виконано постановку задачі дослідження відповідно до існуючих блокчейн-протоколів та їх порівняльного аналізу з роботою клієнт-серверної архітектури. Також було визначено особливості використання технології розподіленого реєстру при розробці ІС в сфері фінансів із зазначенням схем та моделей роботи блоків, транзакцій. В тому числі було здійснено експериментальну перевірку можливостей використання технології розподіленого реєстру на прикладі розробки ІС благодійного фонду.

Технологія розподіленого реєстру є універсальним способом зберігання і обробки інформації, що безумовно сприяє формуванню нових та унікальних рішень у сфері фінансів. Наразі виявлено, що ця технологія і ті механізми, які вона має та пропонує, суто інноваційні та можуть призвести до глобальних змін у світовій економіці. Розподілений реєстр має змогу надати широкий вплив на фінансові установи і такі сфери, як платежі, банкінг, фонди тощо.

Дослідивши варіанти впровадження технології розподіленого реєстру, можна дійти до висновку, що на сьогоднішній день ця технологія хоч і має ряд переваг, проте також має певні обмеження та складнощі, які перешкоджають її ефективній інтеграції у сферу фінансів. Складність розробки та впровадження, великий розмір реєстру – лише деякі з проблем, які фінтех-компанії повинні подолати.

Були детально розглянуті можливості застосування технології у сфері фондової діяльності. Було виявлено, що на даному етапі технологія здатна показати гарні результати щодо підвищення продуктивності роботи системи, прозорості транзакцій. Рекомендації, що були розроблені, можуть використовуватись у майбутньому для аналізу, обліку та зберіганню даних у фінансовій сфері.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Методичні вказівки щодо розробки та оформлення кваліфікаційної роботи (для студентів усіх форм навчання другого (магістерського) рівня програми «Інформаційні управляючі системи та технології») / Упоряд.:Петров К.Е., Левикін В.М., Чалий С.Ф., Євланов М.В., Саєнко В.І., Міхнов Д.К., Міхнова А.В., Чала О.В. - Харків: ХНУРЕ, 2021. – 30с.

2. ДСТУ 3008:2015. Інформація та документація. Звіти у сфері науки і техніки. Структура і правила оформлювання. – Чинний від 22.06.2015. – Київ: ДП «УкрНДНЦ», 2016. – 31 с.

3. ДСТУ 8302:2015. Інформація та документація. Бібліографічні посилання. Загальні положення та правила складання. – Чинний від 04.03.2016. – Київ: ДП «УкрНДНЦ», 2016. – 20 с.

4. Consensus Mechanisms.

URL: https://ethereum.org/en/developers/docs/consensus-mechanisms/ (date of access: 21.11.2023).

5. Вияс Н., Бейджи А. Блокчейн і розподілений реєстр. Концепції, стратегії та практичне застосування. -М: Techtarget, 2022. – 288 с.

6. Вальдуріес П., Тамер М. Принципи організації розподілених баз даних. - М: Print2print, 2020. – 678 с.

7. Blockchain & Role of P2P Network.

URL: https://www.blockchain-council.org/blockchain/blockchain-role-of-p2p-network (дата звернення: 27.11.2023).

8. Равал С. Децентралізовані програми. Технологія Blockchain у дії. -М: O`Reilly, 2017. – 192 с.

9. Гужва В.М. Інформаційні системи і технології на підприємствах: Навч.посібник. / В.М. Гужва - К.: КНЕУ, 2001. – 400 с.

10. Кравченко П., Скрябін Б., Дубініна О., Курбатов О. Блокчейн і децентралізовані системи. Частина 1. -М: DistributedLab, 2020. – 527 с.

11. Лутз М. Розробка веб програм та систем на мові програмування Python.: O`Reilly. Марк Лутз., 2021. – 420 с.

12. Бертачіні М. Криптографічні алгоритми. -М: Packt publishing, 2022. – 358 с.

13. Башир І. Блокчейн. Архітектура, криптовалюти, інструменти розробки, смарт-контракти. -М .: Print2print, 2019. – 512 с.

14. Коваленко Ю. Економіка фінансового сектору. Навчальний посібник, -М: Центр учбової літератури, 2020. – 320 с.

15. Антонопулос А. М., Вуд Г. Освоєння Ethereum, 2018. – 200 с.

16. Варнер Б., Паулін Н. З нуля: цифровий збір коштів для некомерційних організацій, 2020. – 206 с.

17. Черемних С.В. Моделювання та аналіз систем. IDEF-технології. - М .: Фінанси і статистика, 2006. - 192 с.

18. Фаулер М. UML. Основи / М. Фаулер., 2004. – 192 с. – (3 - те видання).

19. Свон М. Блокчейн. Схема нової економіки. -М: Олімп-Бізнес, 2018. – 240 с.

20. Кон М. Agile оцінка та планування проектів. -М: Адель, 2018. – 418 с.

21. Млієвський К. Ю., Петров К. Е. Організація мережевої взаємодії при використанні технології розподіленого реєстру в децентралізованих облікових системах // Abstracts of II International Scientific and Practical Conference «Creation of new ideas of learning in modern conditions» (25.09.2023 – 27.09.2023). – Bordeaux, France, 2023. – P. 274–276. URL: https://eu-conf.com/events/creation-of-new-ideas-of-learning-in-modern-conditions/