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

Факультет інформатики та обчислювальної техніки

(повне найменування інституту, факультету)

Технічної кібернетики

(повна назва кафедри)

«До захисту допущено»
Завідувач кафедри

_____ *І. Р. Пархомей*
(підпис) (ініціали, прізвище)

“ ” 2018 р.

Дисертація
на здобуття ступеня магістра

з напряму підготовки (спеціальності) *121*
(код та назва напряму підготовки або спеціальності)

“Інженерія програмного забезпечення”

на тему *«Розробка алгоритму прискорення процесінгу транзакцій в блокчейні»*

Виконав: студент ІІ курсу, групи *ІТ-74мп Галасюк Владислав Ігорович* _____
(прізвище, ім’я, по батькові) (підпис)

Керівник *ст. вик. Сирота Олена Петрівна* _____
(*посада, прізвище, ім’я, по батькові)* (підпис)

Рецензент:

_____
(підпис)

Засвідчую, що у цьому дипломному проекті немає запозичень з праць інших авторів без відповідних посилань.
Студент _____
(підпис)

Київ – 2018 року

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

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ І.Р. Пархомей
(підпис)

«___»_____2018 р.

ЗАВДАННЯ
на магістерську дисертацію студенту
Галасюку Владиславу Ігоровичу
(прізвище, ім'я, по батькові)

1. Тема дисертації «Розробка алгоритму прискорення процесінгу транзакцій в блокчейні», _____
науковий керівник дисертації к. т. н., ст. вик. Сирота О. П.,_____
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «___»_____ 2018 р. №_____

2. Термін подання студентом дисертації _____

3. Об'єкт дослідження – архітектурні особливості розподілених мереж.

4. Предмет дослідження – порівняння рішень у області блокчейн-технологій та пошук оптимального рішення.

5. Перелік завдань, які потрібно розробити – аналіз концепції розподілених мереж обміну даними та існуючих рішень; аналіз і виявлення недоліків існуючих імплементацій блокчейну; теоретичний опис вдосконалення існуючих імплементацій блокчейну; маркетінговий аналіз стартап-проекту.

6. Орієнтовний перелік ілюстративного матеріалу – шість плакатів

7. Орієнтовний перелік публікацій – дві публікації

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання _____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Ааналіз концепції розподілених мереж обміну даними та існуючих рішень	03.07.2018 р.	
2	Аналіз і виявлення недоліків існуючих імплементацій блокчейну	20.08.2018 р.	
3	Теоретичний опис вдосконалення існуючих імплементацій блокчейну	20.09.2018 р.	
4	Маркетинговий аналіз стартап-проекту	11.10.2018 р.	
5	Висновки	10.11.2018 р.	

Студент _____ Галасюк В. І.
(підпис) (ініціали, прізвище)

Науковий керівник дисертації _____ Сирота О. П.
(підпис) (ініціали, прізвище)

АНОТАЦІЯ

Пояснювальна записка до магістерської дисертації: 65 с., 4 рис., 20 табл., 1 додаток, 12 джерел.

Об’єкт дослідження: архітектури мережевих додатків, розподілені бази даних типу “Блокчейн”.

Актуальність роботи: Питання рішення зазначених в меті роботи проблем стоїть у порядку денному в перших рядках багатьох компаній та стартапів, що займаються побудовою фінтех-рішень на блокчейні. Викладена нижче інформація є компіляцією досліджень експерементальних технологій, які мають можливість покращити досвід користувача та прискорити впровадження у реальний сектор економіки. Тому, ця тема має високий попит серед технологічних компаній, які займаються створенням та поєднанням традиційних методів сплати послуг та криптовалютних рішень.

БЛОКЧЕЙН, КРИПТОВАЛЮТА, АНАЛІТИКА, РОЗПОДІЛЕНІСТЬ, БАЗИ ДАНИХ, P2P, КЛІЄНТ-СЕРВЕР, PROOF OF STAKE, PROOF OF WORK

ABSTRACT

Explanatory note to the master's thesis: 65 p., 4 figures, 20 tables, 1 supplement, 12 sources.

Object of research: the architecture of network applications, distributed databases such as "Blokchain".

Actuality of work: Questions of the solution of the problems mentioned in the purpose of work is on the agenda in the first lines of many companies and startups involved in the construction of fintech solutions on the blockhead. The following information is a compilation of research on experimental technologies that have the potential to improve user experience and accelerate the implementation of the real economy. Therefore, this topic is in high demand among technology companies that are engaged in the creation and combination of traditional payment methods and crypto-currency solutions.

BLOCKCHAIN, CRYPTOCURRENCY, ANALITYCS, DISTRIBUTION, DATABASES, P2P, CLIENT-SERVER, PROOF OF STAKE, PROOF OF WORK

Пояснювальна записка
до магістерської диссертації

на тему: Розробка алгоритму прискорення процесінгу транзакцій в блокчейні.

Київ – 2018 року

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

P2P – peer-to-peer.

Фінтех — технічна область, решення в якої пов’язані з проблемами фінансової індустрії.

Блокчейн — такий спосіб організації зв’язанності даних, при якому кожні дані включають у себе хеш попередніх даних.

Криптовалюта — рішення в області фінтеху, яке поєднує традіційне поняття валюти як еквіваленту обміну та блокчейн як спосіб збереження результатів обміну.

ВСТУП

Мета роботи: дослідити можливі варіанти рішення проблеми сповільнення та об’єму транзакцій в розподіленій базі даних типу “Блокчейн”.

Актуальність роботи: Питання рішення зазначених в меті роботи проблем стоїть у порядку денному в перших рядках багатьох компаній та стартапів, що займаються побудовою фінтех-рішень на блокчейні. Викладена нижче інформація є компіляцією досліджень експерементальних технологій, які мають можливість покращити досвід користувача та прискорити впровадження у реальний сектор економіки. Тому, ця тема має високий попит серед технологічних компаній, які займаються створенням та поєднанням традиційних методів сплати послуг та криптовалютних рішень.

Ціль роботи: вивчення методів та варіантів рішень для блокчейну, які дозволяють прискорити транзакції та зменшити об’єм файлів, які потрібно зберігати користувачу.

1 ВВЕДЕННЯ В РОЗПОДІЛЕНІСТЬ

1.1 Мережеві рівні: поняття та класифікація

Модель взаємозв'язку Open Systems (модель OSI) - це концептуальна модель, яка характеризує та стандартизує функції зв'язку телекомунікаційної або обчислювальної системи без урахування її внутрішньої структури та технології. Його метою є сумісність різноманітних систем зв'язку зі стандартними протоколами. Модель розділяє систему зв'язку на абстрактні шари. Оригінальна версія моделі визначила сім шарів.

Шар служить шару над ним і подає шар під ним. Наприклад, шар, який забезпечує передачу безпомилкових повідомлень через мережу, забезпечує шлях, необхідний додаткам над ним, тоді як він викликає наступний нижній рівень для надсилання та отримання пакетів, які містять вміст цього шляху. Два примірника на тому ж шарі візуалізуються як пов'язані між собою горизонтальним зв'язком у цьому шарі.

Модель є продуктом проекту "Взаємозв'язок відкритих систем" в Міжнародній організації по стандартизації (ISO), який підтримується ідентифікацією ISO / IEC 7498-1.

Фізичний рівень відповідає за передачу та отримання неструктурованих вихідних даних між пристроєм та фізичним середовищем передачі. Він перетворює цифрові біти на електричні, радіо, або оптичні сигнали. Специфікації шару визначають характеристики, такі як рівні напруги, терміни зміни напруги, фізичні швидкості передачі даних, максимальні передавальні відстані та фізичні з'єднувачі. Це включає в себе макет штифтів, напруги, лінійний імпеданс, характеристики кабелю, часові сигнали та частоту для бездротових пристроїв. Управління бітовою швидкістю здійснюється на

фізичному рівні та може визначати режим передачі як симплекс, напівдуплекс та повний дуплекс. Компоненти фізичного рівня можна описати в термінах топології мережі. Всі Bluetooth, Ethernet та USB мають специфікації для фізичного рівня.

Рівень передачі даних забезпечує передачу даних від вузла до вузла - зв'язок між двома безпосередньо з'єднаними вузлами. Він виявляє та, можливо, виправляє помилки, які можуть виникати на фізичному рівні. Він визначає протокол для встановлення та припинення з'єднання між двома фізично пов'язаними пристроями. Він також визначає протокол управління потоком між ними.

IEEE 802 розділяє рівень каналу передачі даних на два підрівня:

- Рівень керування доступом до середовища (MAC) - відповідає за контроль за тим, як пристрої в мережі отримують доступ до середовища та дозволяють передавати дані.

- Рівень логічного контролю (LLC) - відповідає за ідентифікацію та інкапсуляцію протоколів мережевого рівня, а також контролює перевірку помилок та синхронізацію кадрів.

На рівні каналу передачі даних працюють MAC та LLC-шістки мереж IEEE 802, таких як 802.3 Ethernet, 802.11 Wi-Fi та 802.15.4 ZigBee.

Протокол «точка-точка» (PPP) - це протокол рівня каналів передавання даних, який може працювати на декількох різних фізичних рівнях, наприклад, синхронних та асинхронних послідовних лініях.

Стандарт ITU-T G.hn, який забезпечує високошвидкісну локальну мережу через існуючі дроти (лінії електропередачі, телефонні лінії та коаксіальні кабелі), включає в себе повний рівень каналу передачі даних, який забезпечує як корекцію помилок, так і контроль потоку.

Мережевий рівень забезпечує функціональні та процедурні засоби передачі послідовностей даних змінної довжини (так званих пакетів) з одного вузла в інший, підключеного до «різних мереж». Мережа є середовищем, до якого можуть бути підключені багато вузлів, на яких кожен вузол має адресу, і який дозволяє підключеним до нього вузлам передавати повідомлення іншим вузлам, підключеним до нього, просто надаючи вміст повідомлення та адресу адресата вузол і дозволяючи мережі знаходити спосіб доставки повідомлення на цільовий вузол, можливо, маршрутизацію через проміжні вузли. Якщо повідомлення завелике для передачі від одного вузла до іншого на рівні каналу передачі даних між цими вузлами, мережа може здійснювати доставку повідомлень, розбиваючи повідомлення на кілька фрагментів на одному вузлі, відправляючи фрагменти самостійно та повторно збираючи фрагменти на інший вузол. Це може, але не потрібно, повідомляти про помилки доставки.

Доставка повідомлень на мережевому рівні не обов'язково гарантована для надійності; протокол мережевого рівня може забезпечити достовірне надсилання повідомлень, але це не потрібно робити.

Ряд протоколів керування шаром, функція, визначена в прикладі керування ISO 7498/4, належить до мережевого рівня. До них відносяться протоколи маршрутизації, управління груповими групами, інформація про мережевому рівні та помилки, а також призначення адресної адреси мережевого рівня. Це функція корисного навантаження, яка робить їх приналежними до мережевого рівня, а не до протоколу, який їх здійснює.

Транспортний рівень контролює надійність даної лінії через керування потоком, сегментацію / де сегментацію та контроль помилок. Деякі протоколи є орієнтованими на стан та зв'язок. Це означає, що транспортний шар може відстежувати сегменти та повторно передавати ті, які не забезпечують доставку. Транспортний рівень також забезпечує підтвердження успішної передачі даних та надсилає наступні дані, якщо не було помилок. Транспортний

шар створює сегменти з повідомлення, отриманого від прикладного рівня. Сегментація - це процес поділу довгого повідомлення на менші повідомлення.

Простий спосіб візуалізувати транспортний шар - порівняти його з поштовим відділенням, де розглядаються питання відправлення та класифікації поштових відправлень та посилок. Поштова служба перевіряє лише зовнішній конверт пошти, щоб визначити її доставку. Вищі шари можуть мати еквівалент подвійних конвертів, таких як послуги криптографічного представлення, які можуть бути прочитані лише адресатом. Грубо кажучи, протоколи тунелювання діють на транспортному рівні, наприклад, несуть протоколи, не пов'язані з протоколом IP, такі як SNA IBM або IPX Novell через IP-мережу, або повне шифрування з IPsec. Хоча генеративна інкапсуляція маршрутизації (GRE) може здаватися протоколом мережевого рівня, якщо інкапсуляція корисного навантаження відбувається лише на кінцевій точці, GRE стає ближче до транспортного протоколу, який використовує IP-заголовки, але містить повні кадри рівня 2 або 3 пакети для доставки до кінцевої точки. L2TP здійснює кадри PPP всередині транспортних сегментів.

Хоча не розроблено під еталонною моделлю OSI і не суворо відповідає визначенню транспортного рівня OSI, Протокол керування передачею (TCP) та Протокол обробки міграції даних (UDP) Internet Protocol Suite зазвичай класифікуються як протоколи рівня 4 в межах OSI.

Сесійний рівень контролює діалоги (підключення) між комп'ютерами. Вона встановлює, управляє та припиняє зв'язки між локальною та віддаленою програмою. Він забезпечує повнодуплексну, напівдуплексну або простоk операцію та встановлює процедури перевірки, відкладення, припинення та перезапуску. За допомогою моделі OSI цей шар відповідальний за витончений закриття сеансів, що є властивістю протоколу керування передачею, а також для перевірки і відновлення сеансу, який зазвичай не використовується в пакеті

Internet Protocol Suite. Сесійний шар зазвичай реалізується явно в середовищах додатків, які використовують виклики віддаленої процедури.

Шаблон презентації встановлює контекст між об'єктами рівня застосування, в якому об'єкти прикладного рівня можуть використовувати різний синтаксис і семантику, якщо служба презентації забезпечує відображення між ними. Якщо доступне відображення, підрозділи даних протоколу презентації інкапсулюються в блоки даних протоколу сеансу і передаються стек протоколу.

Цей шар забезпечує незалежність від подання даних шляхом перекладу між додатком і мережевими форматами. Шаблон презентації перетворює дані у форму, яку приймає програма. Цей рівень формує дані, що надсилаються по мережі. Його іноді називають синтаксичним шаром. Шаблон презентації може включати функції стиснення. Рівень презентації узгоджує Синтез трансферу.

Оригінальна структура презентації використовувала Основні правила кодування Абстрактної синтаксичної позначки (ASN.1) з такими можливостями, як перетворення текстового файлу EBCDIC в файл ASCII або серіалізацію об'єктів та інших структур даних з та в XML . ASN.1 фактично робить протокол додатків інваріантним щодо синтаксису.

Прикладний рівень - це найближчий до кінцевого користувача рівень OSI, що означає, що як прикладний рівень OCI, так і користувач взаємодіють безпосередньо з програмним забезпеченням. Цей шар взаємодіє з програмами, які реалізують комунікаційний компонент. Такі прикладні програми виходять за рамки моделі OSI. Функції програми-шару зазвичай включають ідентифікацію партнерів по зв'язку, визначення доступності ресурсу та синхронізацію зв'язку. При визначенні партнерів для спілкування, рівень застосування визначає ідентичність та доступність партнерів зв'язку для програми з даними для передачі. Найважливішою відмінністю на рівні

застосування є відмінність між програмою-об'єктом та додатком. Наприклад, веб-сайт резервування може містити дві заявки: один використовує HTTP для спілкування з його користувачами, а один - для протоколу віддаленої бази даних для запису резервування. Жодна з цих протоколів не має нічого спільного з застереженнями. Ця логіка знаходиться в самому додатку. Наявний шар сам по собі не має ніяких засобів визначити наявність ресурсів у мережі.

Таблиця **1.1 – модель OSI**

Номер рівня	**Мережеві рівні**	**Одиниці виміру**
7	Прикладний	Повідомлення
6	Уявлення	Повідомлення
5	Сеансовий	Повідомлення
4	Транспортний	Повідомлення
3	Мережевий	Пакети
2	З’єднування	Кадри
1	Фізичний	Біти

[2] [1]

1.2 Архітектура «клієнт-сервер»

Модель клієнт-сервер - це розподілена структура додатків, яка розділяє завдання або навантаження між постачальниками ресурсу або сервісу, називаються серверами та запитувальниками служб, які називаються клієнтами. Часто клієнти та сервери спілкуються через комп'ютерну мережу на окремому апаратному забезпеченні, але як клієнт, так і сервер можуть перебувати в тій самій системі. Хост сервера запускає одну або декілька серверних програм, які діляться ресурсами з клієнтами. Клієнт не надає ніякого ресурсу, але запитує вміст сервера або службову функцію. Тому клієнти починають спілкування з серверами, які чекають на вхідні запити. Прикладами комп'ютерних програм, що використовують модель клієнт-сервер, є Електронна пошта, мережевий друк та Всесвітня павутина.

Модель передбачає такі основні компоненти:

- набір серверів;
- набір клієнтів;
- мережа.

Характеристика “клієнт-сервер” описує взаємозв'язок програм, що співпрацюють у додатку. Серверний компонент надає функцію або послугу одному або багатьом клієнтам, які ініціюють запити на такі послуги. Сервери класифікуються за наданими ними службами. Наприклад, веб-сервер обслуговує веб-сторінки, а файловий сервер обслуговує комп'ютерні файли. Спільний ресурс може бути будь-яким програмним забезпеченням та електронними компонентами серверного комп'ютера, від програм і даних до процесорів і пристроїв зберігання даних. Обмін ресурсами сервера становить сервіс.

Чи комп'ютер - це клієнт, сервер чи обидва комп'ютера, залежить від характеру програми, яка потребує сервісних функцій. Наприклад, один

комп'ютер може запускати веб-сервер та програмне забезпечення файлового сервера одночасно, щоб обслуговувати різні дані для клієнтів, що здійснюють різні типи запитів. Клієнтське програмне забезпечення також може спілкуватися з серверним програмним забезпеченням на одному комп'ютері. Зв'язок між серверами, наприклад, для синхронізації даних, іноді називається міжсерверним або сервером-сервером зв'язку. [4]

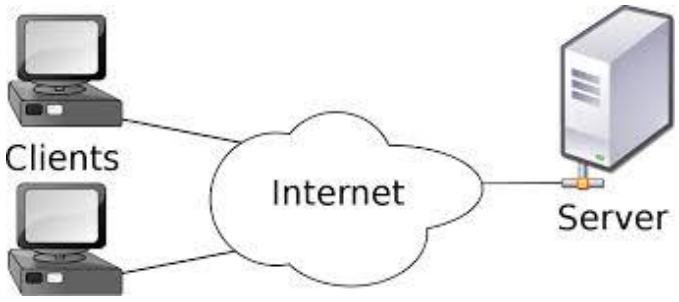


Рисунок **1.**1 – модель «клієнт-сервер»

Загалом, служба є абстракцією ресурсів комп'ютера, і клієнту не потрібно турбуватися про те, як сервер виконує під час виконання запиту та доставки відповіді. Клієнт повинен лише зрозуміти відповідь на основі відомого протоколу додатків, тобто вмісту та форматування даних для запитуваної служби.

Клієнти та сервери обмінюються повідомленнями в режимі повідомлення-запиту-відповіді. Клієнт надсилає запит, і сервер повертає відповідь. Цей обмін повідомленнями є прикладом міжпроцесного зв'язку. Щоб спілкуватися, комп'ютери повинні мати загальну мову, і вони повинні дотримуватися правил, щоб як клієнт, так і сервер знали, чого чекати. Мова та правила зв'язку визначаються в комунікаційному протоколі. Усі клієнт-серверні протоколи працюють на рівні додатків. Протокол прикладного рівня визначає основні схеми діалогу. Щоб формалізувати обмін даними ще далі, сервер може

реалізувати інтерфейс прикладного програмування (API). API - це абстрактний рівень для доступу до сервісу. Обмежуючи зв'язок із певним форматом вмісту, він полегшує розбір. Абстракуюючи доступ, він полегшує крос-платформний обмін даними.

Сервер може одержувати запити від багатьох окремих клієнтів за короткий проміжок часу. Комп'ютер може виконувати обмежену кількість завдань у будь-який момент і спирається на систему планування, щоб пріоритети вхідних запитів від клієнтів для їх розміщення. Щоб запобігти зловживанням та максимально збільшити доступність, серверне програмне забезпечення може обмежити доступність для клієнтів. Напад на відмову в обслуговуванні призначений для того, щоб використати зобов'язання сервера обробляти запити, перевантажуючи їх з надмірними ставками запиту. [3]

Загалом, служба є абстракцією ресурсів комп'ютера, і клієнту не потрібно турбуватися про те, як сервер виконує під час виконання запиту та доставки відповіді. Клієнт повинен лише зрозуміти відповідь на основі відомого протоколу додатків, тобто вмісту та форматування даних для запитуваної служби.

Клієнти та сервери обмінюються повідомленнями в режимі повідомлення-запиту-відповіді. Клієнт надсилає запит, і сервер повертає відповідь. Цей обмін повідомленнями є прикладом міжпроцесного зв'язку. Щоб спілкуватися, комп'ютери повинні мати загальну мову, і вони повинні дотримуватися правил, щоб як клієнт, так і сервер знали, чого чекати. Мова та правила зв'язку визначаються в комунікаційному протоколі. Усі клієнт-серверні протоколи працюють. Модель клієнт-сервер не вказує, що сервер-хости повинні мати більше ресурсів, ніж клієнт-хост. Скоріше за все, це дозволяє будь-якому комп'ютеру загального призначення розширити свої можливості, використовуючи спільні ресурси інших хостів. Централізовані обчислення, однак, спеціально виділяють велику кількість ресурсів на невелику кількість комп'ютерів. Чим більше обчислень вивантажується з клієнтських хостів на

центральні комп'ютери, тим простіше може бути клієнт-хост. Він сильно залежить від мережевих ресурсів (серверів та інфраструктури) для обчислення та зберігання. Бездисковий вузол завантажує навіть свою операційну систему з мережі, а комп'ютерний термінал взагалі не має операційної системи; це лише інтерфейс вводу / виводу для сервера. Навпаки, жирний клієнт, такий як персональний комп'ютер, має багато ресурсів і не залежить від сервера для виконання важливих функцій.

Оскільки мікрокомп'ютери знижували ціну та збільшували потужність з 1980-х до кінця 1990-х років, багато організацій перекладали обчислення з централізованих серверів, таких як мікропроцесори та міні-комп'ютери, до жирних клієнтів. Це забезпечило більший, індивідуалізований панування над ресурсами комп'ютера, але ускладнило управління інформаційними технологіями. Протягом 2000-х років веб-програми витримали достатньо, щоб конкурувати з прикладними програмами, розробленими для конкретної мікроархітектури. Це дозрівання, більш доступне масове зберігання та поява сервіс-орієнтованої архітектури були одним з факторів, які породжували тенденцію хмарних обчислень у 2010 році. на рівні додатків. Протокол прикладного рівня визначає основні схеми діалогу. Щоб формалізувати обмін даними ще далі, сервер може реалізувати інтерфейс прикладного програмування (API) [3]. API - це абстрактний рівень для доступу до сервісу. Обмежуючи зв'язок із певним форматом вмісту, він полегшує розбір. Абстрагуючи доступ, він полегшує крос-платформний обмін даними [4].

Сервер може одержувати запити від багатьох окремих клієнтів за короткий проміжок часу. Комп'ютер може виконувати обмежену кількість завдань у будь-який момент і спирається на систему планування, щоб пріоритети вхідних запитів від клієнтів для їх розміщення. Щоб запобігти зловживанням та максимально збільшити доступність, серверне програмне забезпечення може обмежити доступність для клієнтів. Напад на відмову в

обслуговуванні призначений для того, щоб використати зобов'язання сервера обробляти запити, перевантажуючи їх з надмірними ставками запиту.

1.3 Архітектура «P2P»

Обчислення чи взаємодія однорангової мережі (P2P) - це розподілена архітектура додатків, яка розділяє завдання або навантаження між однолітками. Колеги є однаково привілейованими, рівноправними учасниками програми. Вони, як кажуть, утворюють мережу вузлів однорангової мережі.

Вузли роблять частину своїх ресурсів, таких як потужність обробки, зберігання дисків або смугу пропускання мережі, безпосередньо доступними іншим учасникам мережі, без необхідності центральної координації з боку серверів або стабільних хостів. Вузли є постачальниками та споживачами ресурсів, на відміну від традиційної моделі клієнт-сервер, в якій розподіляється споживання та розподіл ресурсів. Новітні спільні системи P2P виходять за межі епохи вузлів, що роблять подібні речі, розподіляючи ресурси, і шукають різноманітних вузлів, які можуть принести унікальні ресурси та можливості віртуальному співтовариству, тим самим надаючи їй можливість займатися більшими завданнями, ніж такі, що можуть бути виконані індивідуальними вузлами, але це корисно для всіх вузлів.

Хоча P2P-системи раніше використовувалися в багатьох областях застосування, архітектура була популяризована системою обміну файлами Napster, яка була випущена в 1999 році. Концепція надихнула нові структури та філософії у багатьох сферах людської взаємодії. У таких соціальних контекстах рівний-рівному як мем відноситься до егалітарних соціальних мереж, що виникла в усьому суспільстві, що увімкнуто Інтернет-технологіями в цілому. Фраза «peer-to-peer» була вперше використана у 1984 році Парбауелом Йохнухуйтсманом (Parbawell Yohnuhuitsman) при розробці архітектури Advanced Peer to Peer Networking фірми IBM.

Рівноправна мережа розроблена навколо поняття рівних однорангових вузлів, які одночасно функціонують як "клієнти", так і "сервери" для інших вузлів мережі. Ця модель мережевого розташування відрізняється від моделі клієнт-сервер, де зв'язок зазвичай надходить із центрального сервера та з нього. Типовим прикладом передачі файлів, який використовує модель клієнт-сервер, є служба передачі файлів (FTP), в якій клієнтські та серверні програми відрізняються: клієнти ініціюють передачу, а сервери задовольняють цим запитам.

Однорангові мережі зазвичай реалізують деяку форму віртуальної накладної мережі на вершині топології фізичної мережі, де вузли в накладанні утворюють підмножина вузлів у фізичній мережі. Дані все ще обмінюються безпосередньо над базовою мережею TCP / IP, але на рівні додатків однорангові здатні спілкуватися один з одним безпосередньо через логічні накладні посилання (кожен з яких відповідає шляху через базову фізичну мережу). Накладання використовуються для індексації та виявлення одноліткiв, і зробити P2P-систему незалежною від топології фізичної мережі. Виходячи з того, як вузли зв'язані один з одним в накладній мережі, а також про те, як ресурси індексуються та розміщуються, ми можемо класифікувати мережі як неструктуровані або структуровані (або як гібрид між цими двома).

Неструктуровані однорангові мережі не накладають певної структури на накладну мережу за дизайном, а формуються вузлами, які випадково утворюють зв'язки один з одним. (Gnutella, Gossip і Kazaa є прикладами неструктурованих протоколів P2P).

Оскільки немає глобальної належності до них структури, неструктуровані мережі легко створювати і дозволяють локалізовані оптимізації в різних регіонах накладання. Також, оскільки роль всіх одноліткiв у мережі однакова, неструктуровані мережі дуже надійні перед високими темпами «затримки», тобто коли велика кількість одноліткiв часто приєднуються до мережі та виходять з неї.

Проте основні обмеження неструктурованих мереж також виникають з-за цього нестачі структури. Зокрема, коли одноранг хоче знайти бажаний фрагмент даних у мережі, пошуковий запит повинен бути запущений через мережу, щоб знайти максимально можливу кількість однолітків, які діляться даними. Затоплення викликає дуже великий обсяг трафіку сигналу в мережі, використовує більше процесорів / пам'яті (вимагаючи від кожного учасника обробляти всі пошукові запити) і не гарантує, що пошукові запити завжди будуть вирішені. Крім того, оскільки відсутня кореляція між рівним рівнем доступу та вмістом, яким керує це, немає гарантії, що затоплення знайде однорангу, який має потрібні дані. Найпопулярніший вміст, імовірно, буде доступний у кількох однолітків, а будь-який одночасний пошук, можливо, знайде те саме. Але якщо одноранг шукає рідкісні дані, якими поділяють лише кілька інших однолітків, то цілком малоймовірно, що пошук буде успішним.

У структурованих однорангових мережах оверлей організовано у певну топологію, а протокол забезпечує, що будь-який вузол може ефективно шукати в мережі файл / ресурс, навіть якщо цей ресурс є надзвичайно рідким.

Найпоширеніший тип структурованих мереж P2P реалізує розподілену хеш-таблицю (DHT), в якій варіант послідовного хешування використовується для присвоєння власності кожного файлу певному рівному. Це дозволяє однолітцям шукати ресурси в мережі, використовуючи хеш-таблицю: тобто пар (ключ, значення) зберігаються в DHT, і будь-який вузол-учасник може ефективно отримати значення, пов'язане з заданою клавішею.

Проте, для ефективного трафіку трафіку через мережу, вузли в структурованому накладенні повинні підтримувати списки сусідів, які задовольняють певним критеріям. Це робить їх менш надійними в мережах із високою швидкістю обертання (тобто з великою кількістю вузлів, які часто приєднуються та залишають мережу). Більш пізні оцінки рішень щодо виявлення ресурсів P2P під реальними робочими навантаженнями вказали на кілька рішень на базі DHT-рішень, таких як висока вартість ресурсів реклами /

виявлення та статичний та динамічний дисбаланс завантаження. Відомі розподільні мережі, що використовують DHT, включають розподілений трекер BitTorrent, мережу Kad, Storm botnet, YaCy та мережу розподілу вмісту Coral. Деякі відомі наукові проекти включають в себе проект Chord, Kademlia, утиліту зберігання PAST, P-Grid, самоорганізовану та нові накладні мережі та систему розподілу вмісту CoopNet. DHT-мережі також широко використовуються для досягнення ефективного пошуку ресурсів для мережевих обчислювальних систем, оскільки це допомагає керувати ресурсами та планувати використання додатків. Гібридні моделі - це комбінація моделей однорангової та клієнт-серверів. Спільною гібридною моделлю є наявність центрального сервера, який допомагає колегам знаходити один одного. Spotify був прикладом гібридної моделі. Є цілий ряд гібридних моделей, кожен з яких робить компроміси між централізованою функціональністю, що надається структурованою мережею сервера / клієнта, та рівності вузла, наданої чистими одноранговими неструктурованими мережами. В даний час гібридні моделі мають більш високу продуктивність, ніж чисті неструктуровані мережі чи чисті структуровані мережі, оскільки певні функції, такі як пошук, вимагають централізованої функціональності, але мають перевагу від децентралізованої агрегації вузлів, наданих неструктурованими мережами. Системи рівних рівнів створюють унікальні проблеми з точки зору комп'ютерної безпеки.

Як і будь-яка інша форма програмного забезпечення, програми P2P можуть містити вразливі місця. Тим не менш, що робить це особливо небезпечним для програмного забезпечення P2P, це те, що однорангові програми працюють як на серверах, так і на клієнтів, що означає, що вони можуть бути більш вразливими до віддалених експлуатацій. Крім того, оскільки кожен вузол відіграє певну роль у маршрутизації трафіку через мережу, шкідливі користувачі можуть виконувати різні "атаки маршрутизації" або атаки на відмову в обслуговуванні. Приклади загальних атак маршрутизації включають "неправильний маршрутизатор пошуку", за допомогою якого

шкідливі вузли навмисно пересилають запити неправильно або повертають хибні результати "неправильні оновлення маршрутизації", де шкідливі вузли пошкоджують таблиці маршрутизації сусідніх вузлів, відправляючи їм неправдиву інформацію та "неправильний розділ мережі маршрутизації", коли нові вузли приєднуються, вони завантажуються через шкідливий вузол, який поміщає новий вузол у розділ мережі, який заселяється іншими зловмисними вузлами. [5]

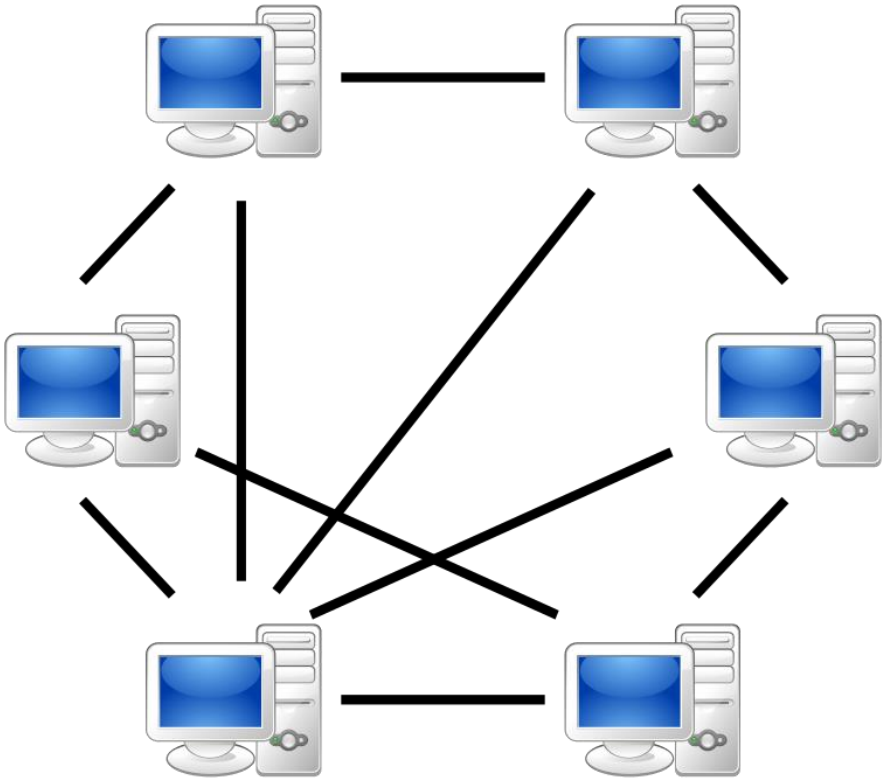


Рисунок 1.2 – модель P2P

1.3.1 Переваги «P2P»

- Зменшення вартості. Централізовані системи, які обслуговують багато клієнтів, зазвичай потребують дуже великих коштів на підтримку. Але система P2P розподіляє, у ідеальному випадку, усе навантаження між користувачами.
- Об'єднання ресурсів. Децентралізований підхід веде до об'єднання ресурсів. Кожен вузол в системі P2P має певні ресурси, як наприклад обчислювальна потужність або пам'ять. У програмах, які потребують величезну кількість цих ресурсів, як наприклад моделювання складних систем або розподілені файлові системи, можна впровадити P2P, щоб використати ці ресурси.
- Вдосконалена масштабованість та надійність. Мережа P2P має підвищену стабільність завдяки структурі. Доки існує хоча б два вузли, мережа є активною. У клієнт-серверній архітектурі, відмова центрального вузла – відмова всієї мережі.
- Збільшена автономія. У багатьох випадках користувачі розподіленої системи не бажають залежати від будь-якого централізованого постачальника послуг. P2P дає їм таку можливість, справедливо розподіляючи ресурси між клієнтами.
- Анонімність/конфіденційність. Пов'язаним із автономією є поняття анонімності і конфіденційності. Користувачі, які потребують захисту приватної інформації, можуть знайти її у такому типі мережі. Криптографічні заходи, що забезпечують надійність передачі даних, запобігають потраплянню третьої сторони у взаємодію, використовуються у P2P. TOR (The Onion Network) є одним з найбільш досконалих прикладів такої інтеграції, бо має систему передачі даних через інші вузли мережі. Це не дає змоги у розумний час відстежити звідки контент прийшов та куди він пішов.
- Динамічність. Системи P2P припускають, що оточення надзвичайно динамічне. Тобто, ресурси, як наприклад вузли, з'являються та зникають із

системи безперервно. У випадках комунікації, як наприклад мережі для обміну повідомленнями, використовуються так званий «список контактів», щоб інформувати користувачів, коли їхні друзі стають доступними. Без цього потрібно було би, щоб користувачі «опитували» партнерів, посилаючи періодичні повідомлення. У випадку розподілених обчислень, як наприклад distributed.net і SETI@home, система повинна пристосуватись до заміни учасників. Тому вони повинні повторно видавати завдання для обчислення іншим учасникам, щоб гарантувати, що робота не втрачена, якщо попередні учасники відпадають від мережі, поки вони виконували крок обчислення. [5]

1.4 Висновки до розділу 1

Перший розділ даної дипломної роботи присвячений вивченню типів мережевої архітектури та пошуку архітектури, яка є найбільш пристосованою для поставленої задачі. Було розкрито недоліки клієнт-серверної архітектури та визначено переваги архітектури «P2P». Для рішень такого класу, який визначений у темі даної роботи, застосовується архітектура «P2P», завдяки очевидним перевагам децентралізації та незалежності. Точніше ця тема буде розкрита у розділі 2.

2 БЛОКЧЕЙН ТА КРИПТОВАЛЮТА

2.1 Асиметричні алгоритми шифрування

2.1.1 Визначення асиметричних криптосистем

Криптографія з відкритим ключем або асиметрична криптографія - це будь-яка криптографічна система, яка використовує пари ключів: відкриті ключі, які можуть бути широко розповсюджені, та приватні ключі, відомі лише власнику. Це виконує дві функції: аутентифікація, де публічний ключ перевіряє, що власник парного приватного ключа надіслав повідомлення, і шифрування, де тільки парний приватний ключовий власник може розшифрувати повідомлення, зашифроване за допомогою відкритого ключа.

У системі шифрування відкритих ключів будь-яка людина може шифрувати повідомлення за допомогою відкритого ключа отримувача. Це зашифроване повідомлення може бути розшифровано лише з приватним ключем одержувача. Щоб бути практичним, створення публічного та приватного ключа-пари має бути обчислено економічно. Сила криптографічної системи відкритого ключа залежить від обчислювальних зусиль (робочого фактору в криптографії), необхідних для пошуку приватного ключа з його парного відкритого ключа. Ефективна безпека вимагає лише приватного ключа приватного характеру; відкритий ключ може бути відкрито розподілений без шкоди для безпеки.

Криптографічні системи з відкритим ключем зазвичай спираються на криптографічні алгоритми на основі математичних задач, які в даний час не допускають ефективного рішення, зокрема ті, що притаманні певній цілій факторизації, дискретному логарифмі та відносинам з еліптичними кривими. Алгоритми відкритого ключа, на відміну від симетричних алгоритмів ключів,

не вимагають безпечного каналу для первинного обміну однією або декількома секретними ключами між сторонами.

Через обчислювальну складність асиметричного шифрування вона зазвичай використовується лише для невеликих блоків даних, як правило, для передачі симетричного ключа шифрування (наприклад, ключ сеансу). Цей симетричний ключ потім використовується для шифрування решти потенційно довгої послідовності повідомлень. Симетричне шифрування / дешифрування базується на більш простих алгоритмах і набагато швидше.

У системі підписів відкритого ключа людина може поєднувати повідомлення з приватним ключем, щоб створити короткий цифровий підпис у повідомленні. Кожен, хто має відповідний відкритий ключ, може поєднувати повідомлення, передбачуваний цифровий підпис на ньому та відомий відкритий ключ, щоб перевірити, чи підпис був дійсним, тобто зробив власник відповідного приватного ключа. Зміна повідомлення, навіть заміна однієї літери, призведе до помилки підтвердження. У безпечній системі підписів це неможливо для обчислення для тих, хто не знає приватного ключа, щоб вивести його з відкритого ключа або будь-якої кількості підписів, або щоб знайти підпису на будь-якому повідомленні, для якого підпис ще не був помічений . Таким чином, автентичність повідомлення може бути продемонстрована підписом, якщо власник приватної ключа зберігає секретний пароль.

Алгоритми відкритого ключа є основними інгредієнтами безпеки в криптосистемах, додатках і протоколах. Вони підкріплюють різні стандарти Інтернету, такі як Transport Layer Security (TLS), S / MIME, PGP та GPG. Деякі алгоритми публічного ключа забезпечують розповсюдження та секретність ключових слів (наприклад, обмін ключем Діффі-Хеллмана), деякі забезпечують цифрові підписи (наприклад, алгоритм цифрового підпису), а деякі - як (наприклад, RSA).

Криптографія відкритого ключа знаходить застосування, зокрема, в дисципліні інформаційної безпеки, інформаційної безпеки. Інформаційна безпека (IS) стосується всіх аспектів захисту електронних інформаційних активів від загроз безпеки. Криптографія відкритого ключа використовується як спосіб забезпечення конфіденційності, автентичності та невідчутності електронних повідомлень та зберігання даних. [9][10]

2.2 Блокчейн

2.2.1 Визначення блокчейну

Блокчейн, або ланцюг блоків, являє собою зростаючий список записів, які називаються блоками, які пов'язані за допомогою криптографії. Кожен блок містить криптографічний хеш попереднього блоку, мітку часу та даних транзакцій (загалом він представлений як хеш кореневої кореневої деревини).

За конструкцією блочний блок стійкий до модифікації даних. Це - "відкритий, розподілений обліковий запис, який дозволяє оперативно фіксувати транзакції між двома сторонами ефективно та постійно". Для використання в якості розподіленої книги, блокчейн, як правило, керується мережею рівних рівнів, яка спільно дотримується протоколу міжвузлового зв'язку та перевірки нових блоків. Після реєстрації дані в будь-якому конкретному блоці не можуть бути змінені зворотньо без зміни всіх наступних блоків, що вимагає консенсусу більшості мереж. Незважаючи на те, що записи блокчейнів не є незмінними, блокчейни можуть бути визнані безпечними за конструкцією та зразком розподіленої обчислювальної системи з високою візантіальною відмовою. Отже, децентралізований консенсус був затребуваний блокчейном.

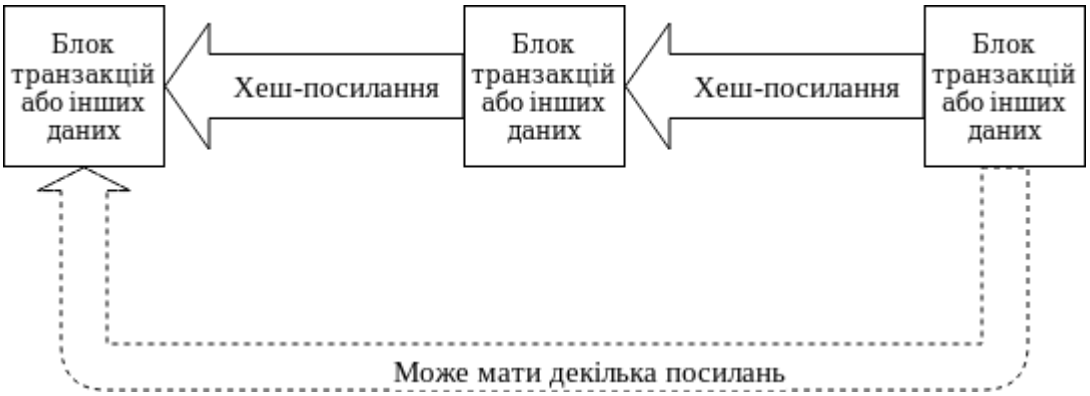


Рисунок 2.1 – схема блокчейну

Блокчейн був винайдений Сатоші Накамото в 2008 році для того, щоб служити загальною книгою транзакцій криптовалютного біткойну. Винахід блочного діапазону для біткоіну зробив це першою цифровою валютою для вирішення проблеми подвійного витрачання без потреби надійного авторитету або центрального сервера. Дизайн біткойна надихнув інші програми, а блокчейни, які читаються широкою публікою, широко використовуються криптовалютами. Блокчейн вважається типом платіжної рейки. Приватні блокчейни були запропоновані для комерційного використання. Джерела, такі як Computerworld, називають маркетинг подібних блокчейнів без належної моделі безпеки "змііне молоко".

2.2.2 Блок транзакцій

Блоки зберігають партії дійсних транзакцій, які хешовано та закодовані в дереві Merkle. Кожен блок включає в себе криптографічний хеш попереднього блоку в блок-схемі, зв'язавши ці два. Зв'язані блоки утворюють ланцюжок. Цей ітераційний процес підтверджує цілісність попереднього блоку, повністю повертаючись до початкового блоку генезу.

Іноді окремі блоки можуть створюватися одночасно, створюючи тимчасову вилку. На додаток до захищеної хеш-історії, будь-який блокчейн має певний алгоритм для підрахунку різних версій історії, так що один з більш високим значенням може бути вибраний над іншими. Блоки, не відібрані для включення в ланцюг, називаються блоками сиріт. Колеги, які підтримують базу даних, час від часу мають різні версії історії. Вони зберігають лише відома їм база даних з найвищою оцінкою. Кожного разу, коли одноранг отримує версію з більшою оцінкою (зазвичай стара версія з єдиним новим блоком), вони розширюють або перезаписують власну базу даних і повторно передають вдосконалення своїм одноліткам. Ніколи не існує абсолютної гарантії, що будь-який конкретний запис залишатиметься в найкращій версії історії назавжди. Blockchains зазвичай створюються, щоб додати оцінку нових блоків на старі блоки, і їм надаються стимули для розширення з новими блоками, а не перезапис старих блоків. Тому імовірність того, що вхід стає замінено, експоненціально зменшується, оскільки на нього покладено більше блоків, які в кінцевому підсумку стають дуже низькими. Наприклад, в блок-схемі, що використовує систему доказів роботи, ланцюжок з найбільш сукупним доказом роботи завжди вважається дієвим у мережі. Існує ряд методів, які можуть бути використані для демонстрації достатнього рівня обчислень. В межах блочного діапазону обчислення проводиться надмірно, аніж у традиційному сегрегованому та паралельному режимі. [7]

2.3 Криптовалюта

2.3.1 Визначення криптовалюти

Криптовалюта (від англ. *Cryptocurrency*) — це цифровий ресурс, призначений для роботи в якості засобу обміну, який використовує сильну криптографію для забезпечення фінансових транзакцій, контролю за створенням додаткових одиниць та перевірки передачі активів. Криптовалюта є свого роду альтернативною валютою та цифровою валютою (з якої віртуальна валюта є підмножиною). Криптовалюти використовують децентралізований контроль, на відміну від централізованої цифрової валюти та центральних банківських систем.

Децентралізований контроль за кожною криптовалютою здійснюється через технологію розподіленої облікової книги, як правило, блок-схеми, яка служить базою даних публічних фінансових операцій.

Біткоїн, вперше випущений як програмне забезпечення з відкритим вихідним кодом у 2009 році, як правило, вважається першою децентралізованою криптовалютою. З моменту випуску біткоїну було створено більше 4000 альткоїн (альтернативні варіанти біткоїн або інших криптовалют). [8]

2.3.2 Proof-of-Work

Доказ виконання роботи (англ. *Proof-of-work*, POW) — принцип економічного заходу для запобігання атакам на службу та інших сервісних зловживанням, таким як спам у мережі, вимагаючи певної роботи від запитуючого сервісу, зазвичай це означає, що комп'ютер обробляє час. Ця концепція була винайдена Синті Двор і Моні Наором, представленими в статті журналу 1993 року. Термін "Доказ роботи" або "PoW" вперше був сформульований і формалізований у 1999 році, написаному Маркусом Якобсоном та Арі Джюленсом. Раннім прикладом системи доказів роботи, що

використовується для визначення вартості валюти, є грошові кошти на Соломонових островах.

Ключовою особливістю цих схем є їхня асиметрія: робота повинна бути помірно важкою (але реальною) на стороні запитувача, але її легко перевірити на постачальника послуг. Ця ідея також відома як функція витрат центрального процесора, клієнтська головоломка, обчислювальна головоломка або функція цінових процесорів. Він відрізняється від CAPTCHA, який призначений для людини для швидкого вирішення, а не для комп'ютера. Пропозиції про простір (PoSpace) застосовують один і той же принцип, доводячи певну кількість пам'яті або дискового простору замість часу процесора. [11]

2.3.3 Proof-of-Stake

Proof-of-stake (PoS) (від англ. *proof of stake*, дослівно: «підтвердження частки») — метод захисту в криптовалютах, заснований на необхідності доказу зберігання певної кількості коштів на рахунку. При використанні цього методу алгоритм криптовалюти з більшою ймовірністю вибере для підтвердження чергового блоку в ланцюжку обліковий запис з великою кількістю коштів на рахунку. Метод використовують як альтернативу методу Proof-of-work (PoW) (доказ виконання роботи), в якому більшу ймовірність підтвердження блоку має обліковий запис з великими обчислювальними потужностями. Метод був запропонований в 2011.

Спільно обидва методи – PoW і PoS – використовуються, наприклад, в криптовалютах EmerCoin, NovaCoin, YaCoin. У криптовалютах PeerCoin і Reddcoin метод PoW використовується для початкового розподілу монет, а PoS — для підтвердження блоків. У криптоплатформі Nxt і BlackCoin метод PoS використовується на всіх етапах.

Аргументи, що вказують на спроможність методу:

- для проведення атаки потрібно багато коштів. Атакуючому буде просто дорого виконати атаку;
- якщо у атакуючого знайдеться багато коштів, він сам постраждає від атаки, оскільки це порушить стійкість криптовалюти.

Аргументи, що викликають побоювання:

- PoS дає додаткову мотивацію до накопичення коштів в одних руках, що може негативно позначитися на децентралізації мережі;
- якщо утвориться невелика група, яка сконцентрує у себе досить великі кошти, вона зможе нав'язувати свої умови функціонування криптовалюти, з яким будуть незгодні більшість міноритаріїв, які не контролюють процесинг. [12]

2.4 Висновки до розділу 2

У другому розділі даного дипломного проекту розглянуто, у яких технологіях блокчейн має найбільший попит, та як досягається надійність та відносна рівність обміну даними. Також були розглянуті недоліки кожної з систем захисту, які створюють неможливість використання криптовалюти у секторах реальної економіки.

3 МОДЕЛЬ РІШЕННЯ ПРОБЛЕМ (PROOF-OF-DATA)

3.1 Введення у проблеми, які вирішує Proof-of-Data (PoD)

У даній роботі було виявлено декілька недоліків традиційних підходів до побудови продуктів з допомогою збереження інформації у блокчейні:

- Нестабільність продукту
- Ненадійність обміну
- Велика затримка при обробці транзакцій
- Великий обєм даних, що зберігаються
- Велика залежність від обчислювальних потужностей
- Велика залежність від об’єма коштів на рахунку

Кожний з цих недоліків не дає змогу впроваждувати технологію блокчейна та криптовалют у повсякденне життя.

Єдине рішення, якє можливо впровадити у реальний сектор економіки, полягає у зміні парадигми, яка керує побудовою алгоритмів роботи такого продукту.

Поперше, необхідно визначити таке поняття, як “корисність” вузла розподіленної мережі.

В системах PoW такий параметр вираховується через кількість виконаної роботи за одиницю часу. Такий підхід призводить до різкого зростання складності добування криптовалюти та, у наступному часі, до повного зніцінення завдяки надвеликому часу прохідження транзакцій, складному використанню та проблемам, зв’язаним з роботою третьої сторони, як провайдера послуги доступу до системи PoW.

У свою чергу PoS не виправляє вищеописані помилки, а створює проблеми, повязані з цнтралізацією коштів. Такий підхід є захищеним від технічних атак, тому що зловмиснику треба мати великі суми коштів на рахунку, але не є надійним з точки зору рівності можливостей разподіленої

банківської системи. Ті користувачі, що мають багато грошей, мають перевагу в швидкості роботи та шансі отримання нових монет. Це не відповідає призначенню криптовалютних систем.

Тому, можна сказати, що PoS, крім проблем технічного характеру, несе проблеми ідеологічного характеру.

3.2 Опис принципів роботи PoD

Основа принципа Proof-of-Data полягає у оцінюванні користувача через те, наскільки активно він приймає участь у роботі мережі.

У параметри оцінки входять:

1. Розмір збережених даних блокчейну
2. Робоча активність вузла (час присутності у мережі, кількість активних підключень)
3. Рівень захищеності транзакції

Формула оцінки є такою: 0,6*S + 0,3*A + 0,1*Sec = X, де: S — об’єм даних блокчейну, A — сума часу присутності та активних підключень, Sec — рівень захисту транзакції, X — коефіцієнт можливості появи нової монети.

Таким чином, створюється урівноважена система, яка спонукає користувачив до:

1. Підтримки мережі за рахунок своїх ресурсів
2. Побудови надійних з’єднань з іншими вузлами мережі
3. Роботи з захищеними транзакціями (вони також мають підвищену нагороду за обробку)

Але, використання такого підходу дозволяє вбудовувати можливості мережі у продукти, задіяні у роботі реального сектору економіки, тому що:

1. Кожний користувач може, але не обов’язково, зберігати велику частину блокчейну. Це актуально для користувачів мобільних пристроїв та носимої електроніки, яка обладнана чипом NFC (Near-Field-Connection).
2. Також, ця система не потребує високої залежності від третьої сторони, бо дає можливість здійснювати транзакції з мінімальним набором блоків у 10 Мб. Кожний мобільний прістрій може дозволити собі працювати з такою технологією.

3. У мережі є принцип рівності користувачів. Система не виділяє більш забезпечених або більш могутніх користувачів. Награда видається за допомогу іншим вузлам в пошуку даних в блокчейні.

3.3 Принцип “Vox Populi”

Цей принцип описує процес прийняття рішень у мережі. По-перше, глобальні зміни стоять вище локальних. Тобто, спочатку оцінюється локальний стан блокчейну, та, якщо він не співпадає з глобальним, локальні зміни видаляються та завантажуються зміни глобальні. Завдяки такому принципу повністю викреслюється можливість атаки через формування захищених цифровим підписом локальних змін, які потім розповсюджуються через вузли мережі. Також він закриває можливість створити кластер з вузлів, які будуть допомогати швидко опрацьовувати транзакції, не виконуючи таки чином принцип рівності в мережі.

3.4 Принцип “Сам собі не вірю”

Цей принцип використовується для захисту від ситуації, коли один вузол виходить на перше місце по швидкості обробці транзакцій та створює кластер навколо себе. Тому, для захисту, цей вузол не може в повній мірі полягатися на свої можливості. Для нього важливим є надійність першого принципу захисту. У мережі такий вузол допоможе отримати нагороду іншим вузлам, але не отримає її сам, доки не стане агентом обробки транзакції.

Загалом, такий підхід має назву “пряма цифрова демократія”.

3.5 Висновки до розділу 3

Третій розділ даної роботи присвячено опису моделі поведінки системи, яка полягається на принцип “Proof-of-Data”. Така система буде значно зручнішою для повсякденного використання та впровадження в сектор реальної економіки. Тому, автор цієї роботи вважає можливим широкомасштабне впровадження, завдяки цієї технології, розподілених мереж та блокчейн-зберігачів даних у повсякденне життя людини.

4 РОЗРОБКА СТАРТАПУ

4.1 Опис ідеї проекту

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Прискорення транзакцій блокчейну без втрати на рівні безпеки	Страхування, медицина, фінансова галузь, управління документами	Зручність використання, надійність, швидкість

Таблиця 5.2. Опис ідеї стартап-проекту

№ п/п	Техніко-економічні характеристики ідеї	Продукція конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	BTC	ETH	BSH			
1	Зручність	+	-	-	+			+
2	Швидкість	+	-	+	+			+
3	Надійність	+	+	+	+			+
4	Універсальність	+	-	+	-			+

4.2 Технологічний аудит ідеї проекту

Таблиця 4.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Цілком однорангова мережа	P2P, TCP, UDP	Наявні	Доступні
2	Дворангова мережа	P2P, TCP, UDP, рахування рейтингу системи	Наявні	Доступні
3	Централізована мережа	Використання понадпотужних серверних систем	Наявні	Недоступні
Обрана технологія реалізації ідеї проекту:1				

Висновок: технологічна реалізація продукту – можлива, вибрана технологія №1

4.3 Аналіз ринкових можливостей запуску стартап-проетку

Таблиця 4.4. Попередня характеристика потенційного ринку

№ п/п	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	10
2	Загальний обсяг продаж, грн./ум.од	$ 59 974 055 372
3	Динаміка ринку	Ринок зменшується
4	Наявність обмежень для входу	Зниження репутації процесу ICO
5	Специфічні вимоги до стандартизації та сертифікації	Відкритість коду продукта
6	Середня норма рентабельності в галузі або по ринку, %	30%

Висновок: враховуючи кількість головних гравців по ринку, динаміку ринку, що знижується, невелику кількість конкурентів та велику капіталізацію

ринку можна зробити висновок, що на даний момент, ринок для входження стартап-продукту є привабливим.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Швидкість транзакцій	Малий та середній бізнес та поодинокі споживачі	Бізнес зацікавлений у надійному блокчейні, споживачі – у швидкому та надійному блокчейні	Споживачі потребують надійний та швидкий продукт

Таблиця 4.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренти	Наявність конкурентів котрі надають схожі рішення	Зменшення ціни на поставлену послугу; Розробка унікальних характеристик товару; Надання ліцензій на обслуговування
2	Кошти на розробку та підтримку продукту	Закінчення грошей та недостатнє фінансування	Залучення додаткових інвесторів, мотивація роботи на перспективу; Ітеративна розробка продукту задля покрокового виведення продукту на ринок та отримання відповіді користувачів
3	Вихід аналогу	Вихід аналогу даного товару може призвести до знецінення та безідейності даного товару	Вихід товару на ринок в коротші строки з не повною, але достатньою, функціональністю для зацікавлення усіх цільових аудиторій; Проведення рекламної компанії

Таблиця 4.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Новий продукт	Вихід на ринок, Зменшення монополії, Надання нових рішень у сфері	Розробка нової функціональності; Вихід нової продукції на ринок; Надання різноманітних типів ліцензій в залежності від потреб користувача \ замовника.
2	Вихід аналогу	Надати продукт з певними характеристиками та можливостями що відсутні у компаній конкурентів	Аналіз ринку та користувачів задля задоволення їх потреб та надання функціональності у найкоротші строки за ціну, котра є дешевшою ніж у продуктів-замінників.
3	Зворотній зв’язок від користувачів	Можливість отримання необхідної інформації для вдосконалення продукту	Наявність вхідних даних та реакція на них з боку команди розробників задля задоволення потреб та бажань кінцевих користувачів системи кешування даних.
4	Грошова винагорода за рекламу	При достатньому попиту на систему кешування даних можлива комерціалізація продукту на основі реклами задля отримання грошової винагороди для подальшого розвитку продукту та оплати заробітної плати працівникам	Точкова комерціалізація продукту; Введення реклами; Ведення додаткових коштів у проект задля його подальшого розвитку.

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: монополістична	Товар від кожної компанії на ринку, являється недосконалим замінником товару, реалізованого іншими фірмами; На ринку є умови для входу та виходу; Ціна корелює між суперниками;	Розробка продукту з характеристиками, які покривають сфери вживання що не покривають інші товари-замінники; Кореляція цін у відповідності до товарів замінників; Різни типи ліцензій.
2	Рівень конкурентної боротьби: світовий	Всі продукти замінники розроблялись інтернаціональними командами з різних куточків світу, продукти не належать до певної держави, а належать команді розробників	Вихід на ринок збуту продукту з клієнто-необхідною функціональністю; Налагодження маркетингу на основних Інтернет ресурсах задля охоплення великої кількості потенційних користувачів; Надання бета-версій продукту.
3	Галузева ознака: внутрішньогалузева	Даний тип продукту може використовуватися тільки у сфері розробки IT додатків \ продуктів	Надання зручного, інтуїтивно зрозумілого інтерфейсу; Підтримка всім відомих методів взаємодії з середовищем розробки; Наявність документації та он-лайн підтримки.
	Конкуренція за видами товарів: товарно-видова	Дана конкуренція – конкуренція між товарами одного виду.	Впровадження функціональності яка відсутня у товарів-замінників; Спрощення інтерфейсів; Надання підтримки.
	Характер	Цінові переваги – точкова	Надання платних ліцензій

	конкурентних переваг: цінова та не цінова	комерціалізація; Не цінова – надання функціональності, що відсутня у товарах-замінниках.	лише на критично важливу функціональність для клієнта з певним строком підтримки, що зазначена у відповідній ліцензії; Впровадження унікальної функціональності.
	За інтенсивністю: марочна	Наявність унікального знаку що відрізняє даний продукт від продуктів-замінників	Впровадження власної назви та власного знаку.

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	NEO, NEM	Немає	Немає	Малий та середній бізнес	Немає
Висновки	Дуже високий поріг входу. Не зможуть задовольнити усіх клієнтів				

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею [11].

Для виходу на ринок продукт повинен мати функціонал, що має більшу швидкодію та надійність виконуваних дій.

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

	Фактор конкурентоспроможності	Обґрунтування
	Частка ринку	Стартап-проекти, які хочуть досягнути безпеки у системах обміну валютними еквівалентами
	Ціна	Невелика, у порівнянні з такими продуктами як WWInfo
	Асортимент	Не грає ролі
	Репутація виробника	Грає велику роль

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін системи PoD

	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим рішенням						
			3	2	1		1	2	3
	Частка ринку	5	-	-	-	0	-	-	-
	Ціна	10	-	-	-	0	-	-	-
	Асортимент	20	-	-	-	0	-	-	-
	Репутація виробника	0	-	-	-	0	-	-	-

Таблиця 4.12. SWOT аналіз стартап-проекту

Сильні сторони (S): – Надійність – Швидкість	Слабкі сторони (W): – Час реалізації – Контроль
Можливості (O): – Впровадження у секторі реального бізнесу	Загрози (T): – Несприйняття зі сторони крипто-ентузіастів

Таблиця 5.13. Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Безкоштовне надання певного функціоналу у користування	Головний ресурс – люди, даний ресурс - наявний	4-5 місяців

	споживачам на обмежений термін		
2	Реклама	Залучення власних коштів для реклами товару	1-2 місяці
3	Написання статей та опис товару на відомих ресурсах	Головний ресурс – час, даний ресурс - наявний	2-3 тижні
4	Презентація товару на хакатонах й інших ІТ заходах	Ресурс – час та гроші для участі, наявні	1-3 місяці

4.4 Розроблення ринкової стратегії проекту

Таблиця 4.14. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Людина, що зацікавлена впровадженням електронних коштів в своєму бізнесі	Споживачі бажають знайти таке рішення	Попит складе 70% цільової аудиторії	Конкуренція велика у початковому сегменті	Вхід є складним
Які цільові групи обрано: групу 1 (весь ринок)					

Відповідно до проведеного аналізу можна зробити висновок, що підходящою цільовою групою для розповсюдження даного програмного продукту є працівники ІТ сфери, ІТ компанії в цілому та будь-які підприємства котрі використовують програмні продукти з елементами фінансових відношень. Відповідно до стратегії охоплення ринку збуту товару обрано стратегію масового маркетингу, оскільки для підприємств, ІТ працівників та ІТ компаній у цілому надається стандартизований продукт з можливістю розширення функціональності за домовленістю (відповідно до ліцензії).

Таблиця 4.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Надання функціональності що відсутня у товарів-замінників, підтримка клієнтів	Проведення реклами, освітлення унікальної функціональності через інтернет ресурси та інші канали, контакт напряму з споживачами; формування лояльності і прихильності споживачів	Зниження ступеню замінності товару; Прихильність клієнтів; Відмітні властивості товару; Відмітні характеристики товару;	Стратегія диференціації

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
оскільки є товари-заміннники, але дані товари замінники не мають деякого необхідного функціоналу	Так, ціль компанії знайти нових споживачів та, частково, забрати існуючих у конкурентів задля задоволення потреб останніх	Компанія частково копіює характеристики товару конкурента, основна ціль компанії розробка нового унікального функціоналу, з підтримкою основного функціоналу конкурентів	Стратегія заняття конкурентної ніші

Таблиця 4.17. Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Аудиторія має бути спроможною інтегровувати рішення у свої бізнес-процеси	Поступове нарощування нових функцій	Швидкість впровадження нових функцій	Швидкість, надійність

Відповідно до проведеного аналізу можна зробити висновок, що стартап-компанія вибирає як базову стратегію розвитку – стратегію поступового розвитку, як базову стратегію конкурентної поведінки – стратегію заняття конкурентної ніші.

4.5 Розроблення маркетингової програми стартап-проекту

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Швидкість, надійність	Швидкість, надійність	Швидкість, надійність

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	Технологія блокчейну на принципі PoD		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Швидкість	Монотонна	Немає
	Надійність	Монотонна	Немає
3.Товар із підкріпленням	До продажу: наявна повна документація, акції на придбання декількох ліцензій, знижки для певних сегментів на покупку товару		
	Після продажу: додаткова підтримка спеціалістів налаштування, підтримка з боку розробника		
За рахунок чого потенційний товар буде захищено від копіювання: захист інтелектуальної			

власності, патент

В/Нв – відчутні/невідчутні; М/Нм – монотонні/немонотонні; Пр/Нпр – параметричні/непараметичні; Вр/Тх/Тл/Е/Ор – вартісні/ технічні/ технологічні/ ергономічні/ органолептичні; О/К/С – обов'язкові/ кількісні/ сюрпризні

Таблиця 4.20. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
Високий	Середній	Високий	-30% від ціни замінників

Таблиця 4.21. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Немає	Портал еміссії	Дорівнює об’єму умовних одиниць, що поступають на рахунок компанії	Портал збуту (сайт)

Таблиця 4.22. Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Азарт	Веб-портали	Шанс на отримання надприбутків	Шанс на отримання надприбутків	Звернення має поєднувати усі групи цільових клієнтів
2	Наполегливість		Постійна робота приносить великий дохід	Постійна робота приносить великий дохід	

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

4.6 Висновки по розділу

В четвертому розділі описано стратегії та підходи з розроблення стартап-проекту, визначено наявність попиту, динаміку та рентабельність роботи ринку, як висновок було вказано що існує можливість ринкової комерціалізації проекту. Розглянувши потенційні групи клієнтів, бар’єри входження, стан конкуренції та конкурентоспроможність проекту було встановлено що проект є

перспективним. Розглянуто та вибрано альтернативу впровадження стартап-проекту та доведено доцільність подальшої імплементації проекту.

ВИСНОВКИ

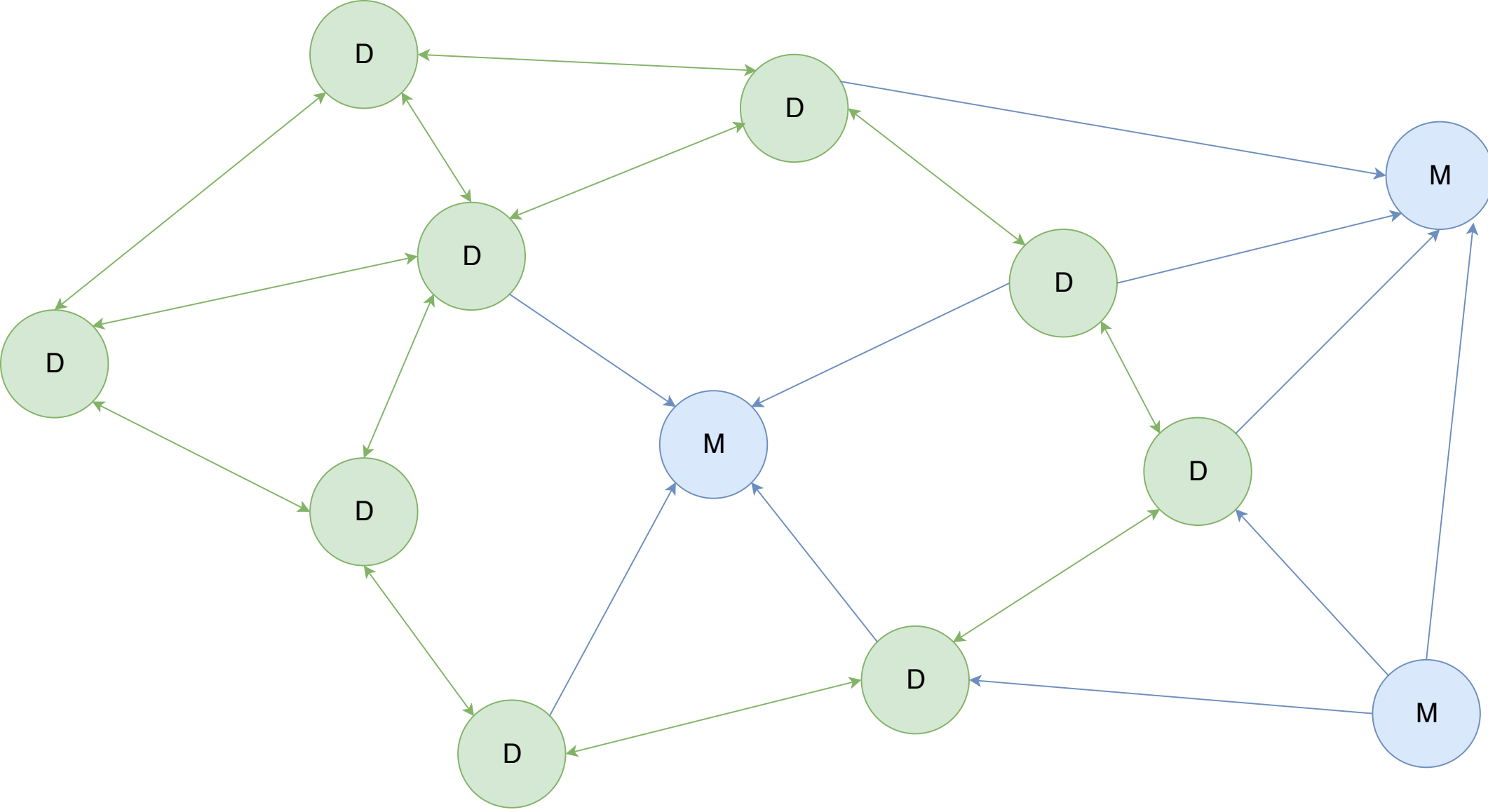
У процесі досліджень для цієї наукової роботи було виконано моделювання декількох продуктів, що використовують блокчейн як технологію, досліджено їх недоліки та переваги. Завдяки дослідженням було побудовано модель такого продукту, який би задовольнив потреби користувачів та замовників у секторі реальної економіки, що було доведено у розділі 4 цієї роботи. Тому, вважаю за необхідне сказати, що така модель, впроваджена на рівні функціонуючого продукту, може призвести до повної перебудови продукту. Тому, усі сервіси, що хочуть використовувати її, потрібні бути спроектовані з урахуванням такої моделі.

ПЕРЕЛІК ПОСИЛАНЬ

1. Мережева архітектура — Вікіпедія [електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Мережева_архітектура (дата звертання: 11.09.2018).
2. Federal Standard 1037C: Glossary of Telecommunications Terms [електронний ресурс] – Режим доступу: https://www.its.bldrdoc.gov/fs-1037/fs-1037c.htm (дата звертання: 11.09.2018).
3. Клієнт-серверна архітектура — Вікіпедія [електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Клієнт-серверна_архітектура (дата звертання: 11.09.2018).
4. The Rise of Web Service Ecosystems [electronic resource] / A.P. Barros, M. Dumas / IT Professional / Volume: 8, Issue: 5 / 30 October 2006 – Access mode: http://ieeexplore.ieee.org/document/1717340/ (access date: 11.09.2018).
5. Peer-to-peer — Вікіпедія [електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Peer-to-peer (дата звертання: 11.09.2018).
6. A Definition of Peer-to-Peer Networking for the Classification of Peer-to-Peer Architectures and Applications / Rüdiger Schollmeier / IEE – 2002.
7. Блокчейн — Вікіпедія [електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Блокчейн (дата звертання: 12.11.2018).
8. Криптовалюта — Вікіпедія [електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Криптовалюта (дата звертання: 12.11.2018).
9. Асиметричні алгоритми шифрування — Вікіпедія [електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Асиметричні_алгоритми_шифрування (дата звертання: 12.11.2018).
10. Handbook of Applied Cryptography (5th edition) / Alfred J. Menezes, Paul C. van Oorschot, Scott A. Vanstone / CRC Press – 2001.

11. Proof-of-Work — Вікіпедія [електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Доказ_виконаної_роботи (дата звертання: 12.11.2018).
12. Proof-of-Stake — Вікіпедія [електронний ресурс] – Режим доступу: https://uk.wikipedia.org/wiki/Proof-of-stake (дата звертання: 12.11.2018).

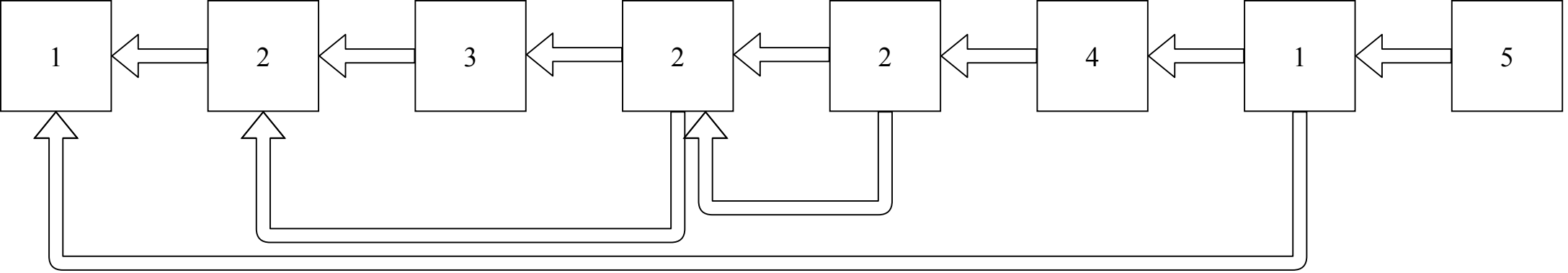
ГРАФІЧНІ МАТЕРІАЛИ



Демонстраційний плакат
до магістерської дисертації на тему: "Розробка алгоритму прискорення процесінгу транзакцій в блокчейні"

Виконав: _____

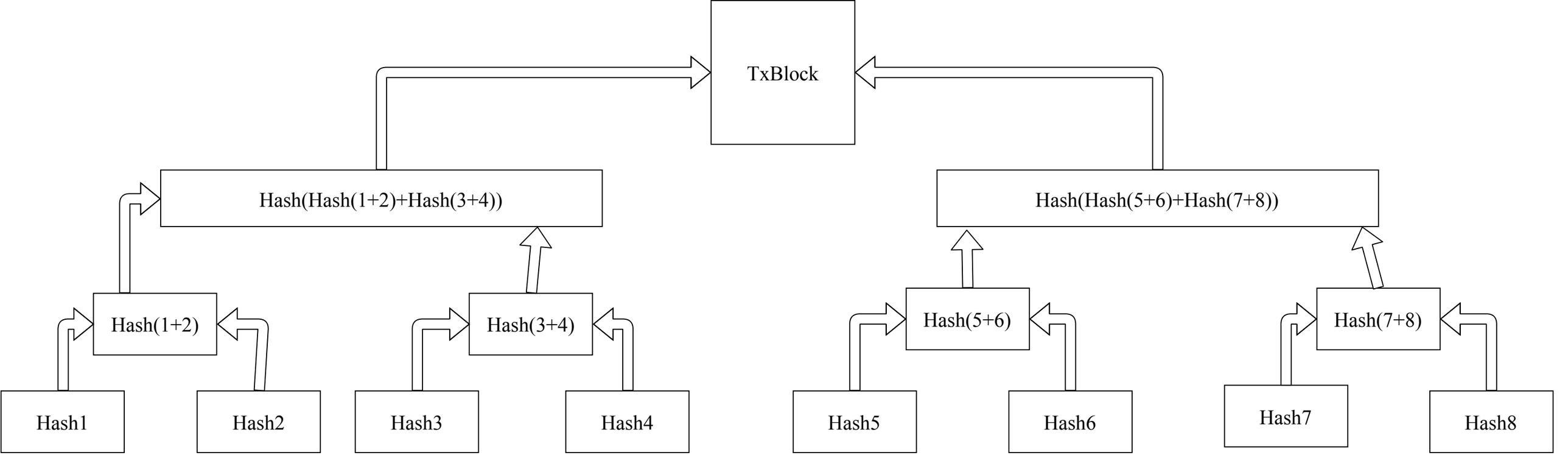
Перевірив: _____



Демонстраційний плакат
до магістерської дисертації на тему: "Розробка алгоритму прискорення процесінгу транзакцій в блокчейні"

Виконав: _____

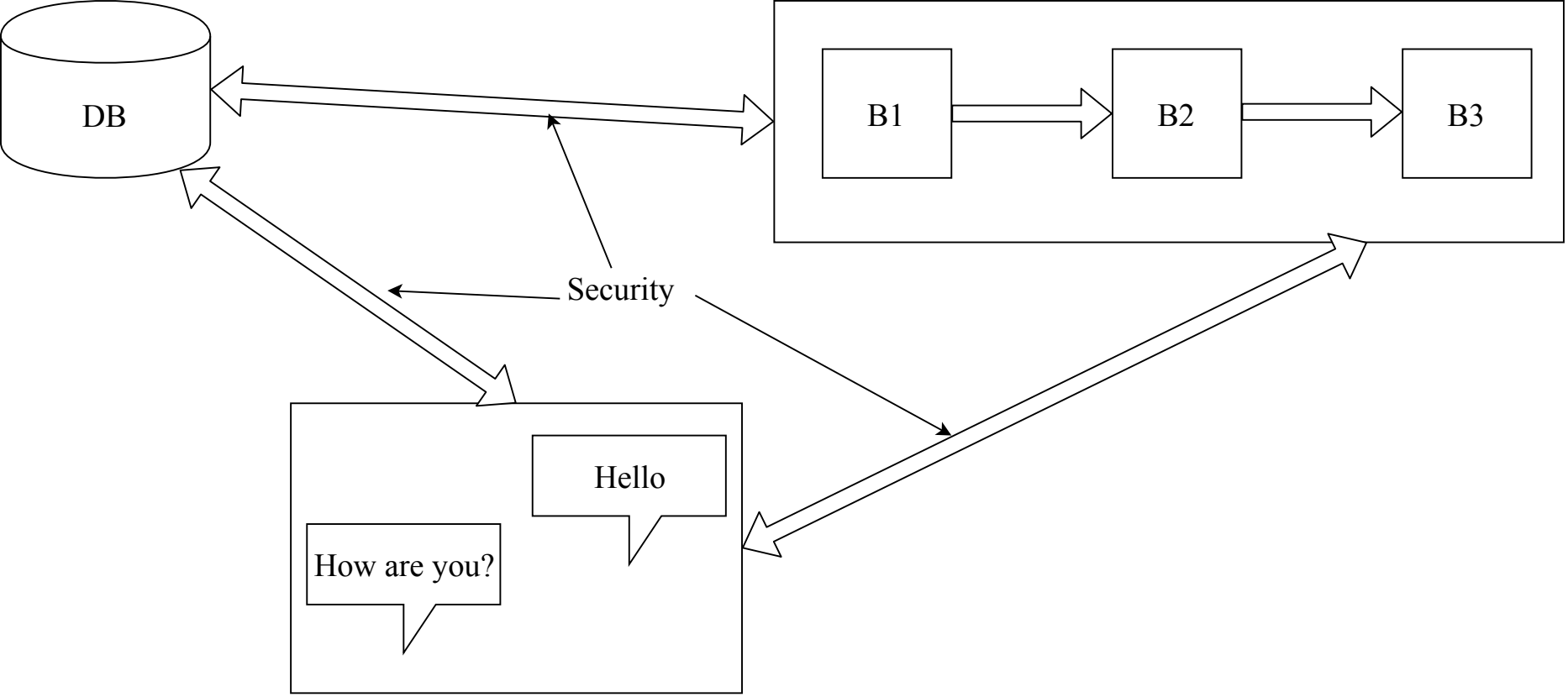
Перевірив: _____



Демонстраційний плакат
до магістерської дисертації на тему: "Розробка алгоритму прискорення процесінгу транзакцій в блокчейні"

Виконав: _____

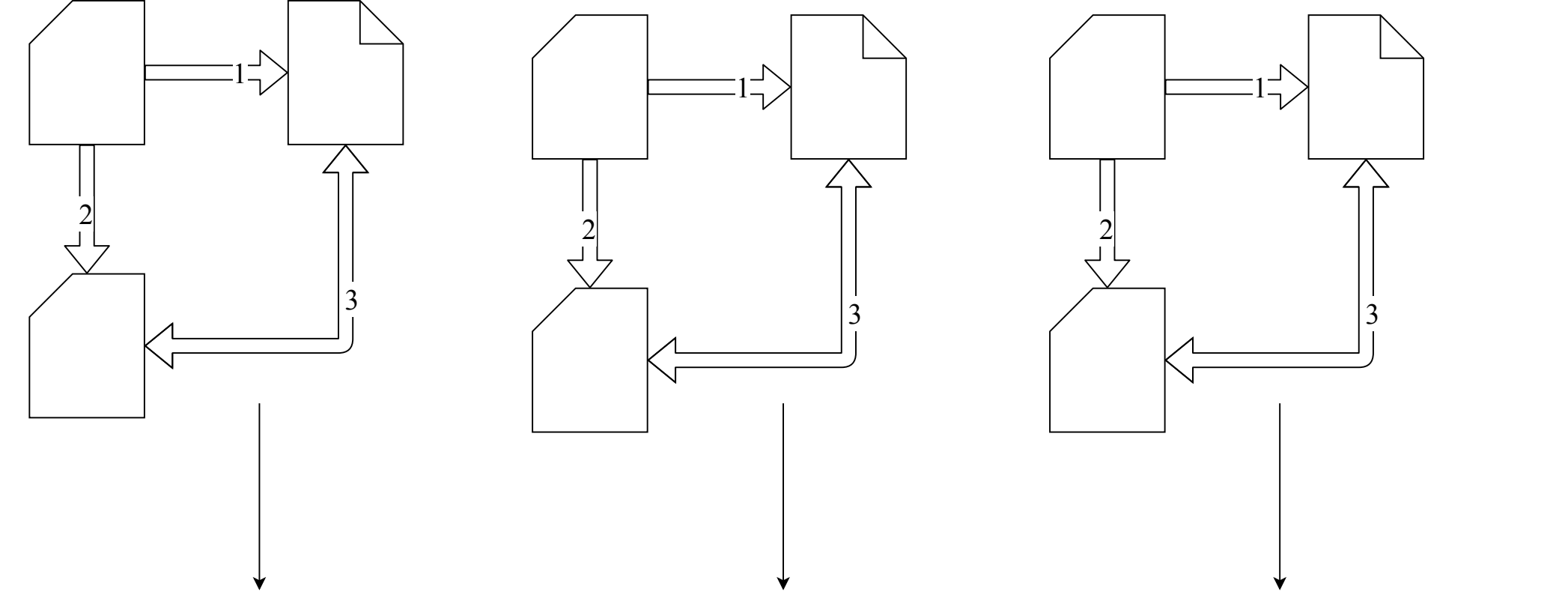
Перевірив: _____



Демонстраційний плакат
до магістерської дисертації на тему: "Розробка алгоритму прискорення процесінгу транзакцій в блокчейні"

Виконав: _____

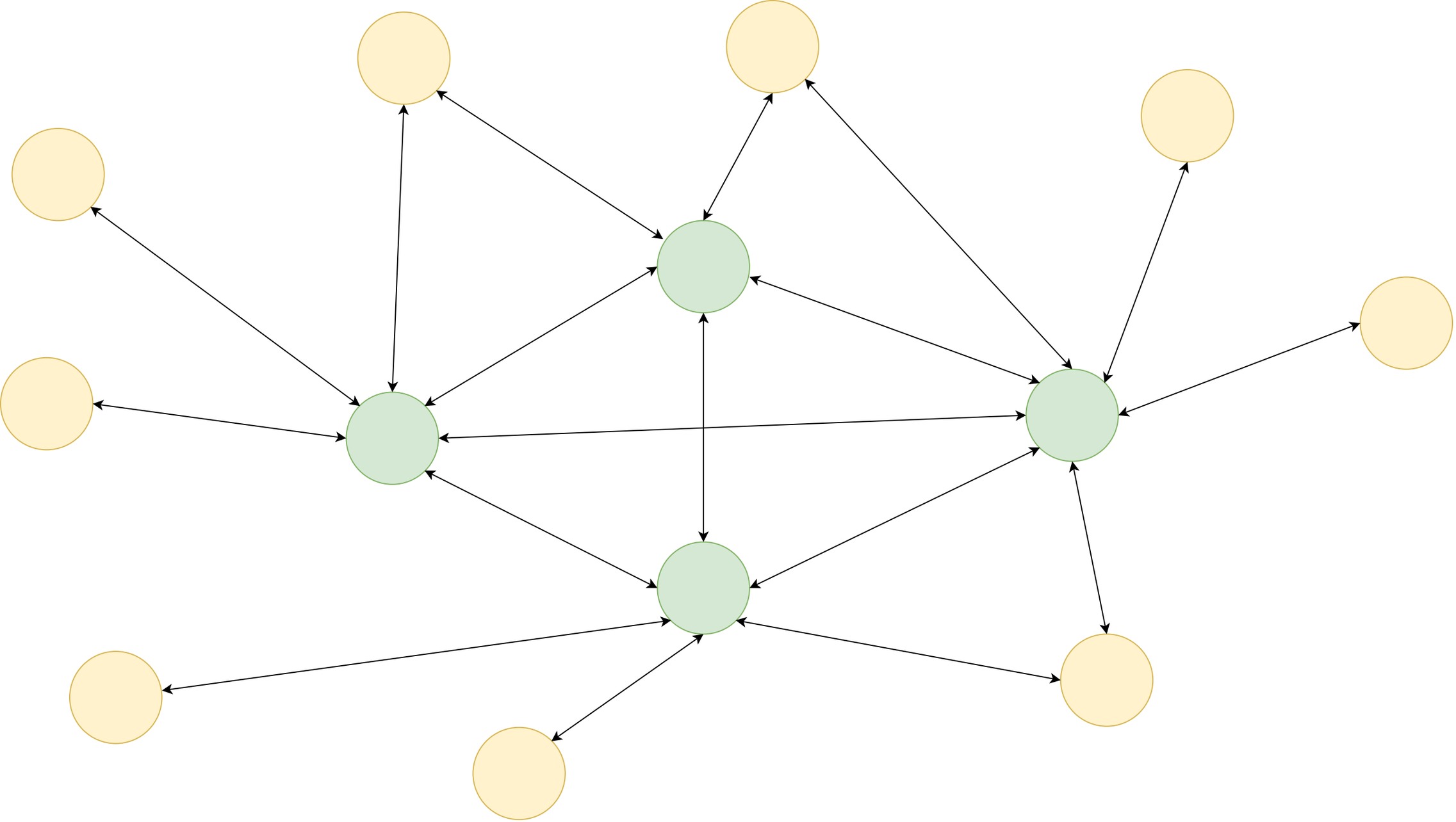
Перевірив: _____



Демонстраційний плакат
до магістерської дисертації на тему: "Розробка алгоритму прискорення процесінгу транзакцій в блокчейні"

Виконав: _____

Перевірив: _____



Демонстраційний плакат
до магістерської дисертації на тему: "Розробка алгоритму прискорення процесінгу транзакцій в блокчейні"

Виконав: _____

Перевірив: _____