

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра комп'ютерних наук
(повна назва кафедри)

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

бакалавр

(назва освітнього ступеня)

на тему: Огляд засобів програмної реалізації для МaaS-платформи «розумних міст»

Виконав: студент IV курсу, групи СН-42

спеціальності 122 Комп'ютерні науки

(шифр і назва спеціальності)

(підпис) Яцків К.В.
(прізвище та ініціали)

Керівник _____
(підпис) Мацюк Г.Р.
(прізвище та ініціали)

Нормоконтроль _____
(підпис) Липак Г.І.
(прізвище та ініціали)

Завідувач кафедри _____
(підпис) Боднарчук І.О.
(прізвище та ініціали)

Рецензент _____
(підпис) Цуприк Г. Б.
(прізвище та ініціали)

Тернопіль
2026

Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)
Кафедра комп'ютерних наук
(повна назва кафедри)

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Боднарчук І.О.
(підпис) (прізвище та ініціали)

« 8 » червня 2026 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

на здобуття освітнього ступеня Бакалавр
(назва освітнього ступеня)

за спеціальністю 122 Комп'ютерні науки
(шифр і назва спеціальності)

Студенту Яцків Каті Василівні
(прізвище, ім'я, по батькові)

1. Тема роботи Огляд засобів програмної реалізації для МaaS-платформи «розумних міст»

Керівник роботи Мацюк Галина Ростиславівна, к.н.с.к., доцент кафедри ІМ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від « 14 » травня 2026 року № 4/7-239

2. Термін подання студентом завершеної роботи 23 червня 2026 р.

3. Вихідні дані до роботи Наукові публікації щодо «розумних міст», МaaS-платформ та програмно-алгоритмічних засобів для їх реалізації та інтеграції

4. Зміст роботи (перелік питань, які потрібно розробити)

Вступ. 1. Аналіз теоретико-технологічних засад та програмного інструментарію реалізації МaaS-платформ. 2. Програмно-технологічна реалізація та моделювання архітектури МaaS-платформи. 3. Оцінювання ефективності програмно-технологічних рішень для «розумного міста». 4. Безпека життєдіяльності, основи охорони праці. Висновки. Перелік джерел.

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)

1. Титульна сторінка. 2. Тема та мета дослідження. 3. Завдання дослідження. 4. Актуальність дослідження. 5. МaaS-платформа «розумного міста». 6. Класифікація засобів реалізації МaaS-платформи «розумного міста». 7. Діаграма варіантів використання МaaS-платформи «розумного міста». 8. Структура МaaS-платформи «розумного міста». 9. Модель даних прототипу МaaS-платформи «розумного міста». 10. Діаграма структури ланцюгового коду МaaS-платформи «розумного міста». 11. Діаграма структури коду бекенду МaaS-платформи «розумного міста». 12. Тестування бекенду МaaS-платформи «розумного міста». 13. Висновки. 14. Завершальний слайд.

АНОТАЦІЯ

Огляд засобів програмної реалізації для МaaS-платформи «розумних міст» // Кваліфікаційна робота освітнього рівня «Бакалавр» // Яцків Катя Василівна // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра комп'ютерних наук, група СН-42 // Тернопіль, 2026 // С. 71, рис. – 9, табл. – 6, кресл. – 14, додат. – 0, бібліогр. – 38.

Ключові слова: МaaS-платформа, блокчейн, ланцюговий код, модель даних, мультимодальні перевезення, навантажувальне тестування, розумне місто, стек технологій.

Кваліфікаційна робота присвячена дослідженню засобів програмної реалізації для МaaS-платформи «розумних міст». В першому розділі кваліфікаційної роботи годано теоретико-методологічні засади проєктування МaaS-платформ. Проаналізовано світовий досвід та інструментарій програмної реалізації МaaS-платформ. Висвітлено теоретико-технологічні основи застосування технології та класифікація. Проаналізовано технологічні виклики та обмеження інтеграції технології. Подано порівняльний аналіз програмних платформ розподіленого реєстру для інфраструктури «розумного міста». В другому розділі кваліфікаційної роботи сформовано функціональну архітектуру та структуру МaaS-платформи. Створено модель даних. Розглянуто ланцюговий код. Проаналізовано компонентно-модульну структуру та багаторівневу архітектуру серверної частини МaaS-платформи «розумного міста». В третьому розділі кваліфікаційної роботи описано процес розгортання та апаратно-програмне забезпечення експериментального середовища. Розглянуто функціональне тестування прототипу. Проведено дослідження продуктивності та навантажувальне тестування елементів. Розглянуто процес налаштування мережі блокчейн для прототипу МaaS-платформи. Подано порівняльний аналіз з іншими інноваційними технологіями.

ANNOTATION

Overview of Software Implementation Tools for Smart City MaaS Platforms // Qualification work of the educational level «Bachelor» // Katia Yatskiv // Ternopil Ivan Pulyu National Technical University, Computer and Information Systems and Software Engineering Faculty, Computer Sciences Department, group SN-42 // Ternopil, 2026 // P. 71, fig. – 9, tabl. – 6, chair. – 14, annexes. – 0, references – 38.

Keywords: MaaS platform, blockchain, chain code, data model, multimodal transportation, load testing, smart city, technology stack.

The qualification paper is devoted to the research of software implementation tools for a smart city MaaS platform. The first chapter presents the theoretical and methodological foundations of MaaS platform design. The global experience and software implementation tools for MaaS platforms are analyzed. The theoretical and technological framework of technology application and its classification are highlighted. The technological challenges and limitations of technology integration are analyzed. A comparative analysis of distributed ledger software platforms for smart city infrastructure is provided. The second chapter defines the functional architecture and structure of the MaaS platform. A data model is developed, and the chaincode is examined. The component-modular structure and multi-tier architecture of the smart city MaaS platform's backend part are analyzed. The third chapter describes the deployment process and the hardware and software equipment of the experimental environment. The functional testing of the prototype is considered. Performance research and load testing of the elements are conducted. The process of blockchain network configuration for the MaaS platform prototype is examined. A comparative analysis with other innovative technologies is presented.

ПЕРЕЛІК СКОРОЧЕНЬ І ТЕРМІНІВ

СУБД – система керування базами даних.

ШІ – штучний інтелект.

API (англ. Application Programming Interface) – інтерфейс прикладного програмування.

BSMD (англ. Blockchain-based Smart Mobility Data-market) – ринок даних інтелектуальної мобільності на базі технології блокчейн.

CRUD (англ. Create, Read, Update, Delete) – базові операції керування даними (створення, зчитування, оновлення, вилучення).

EV (англ. Electric Vehicle) – електромобіль.

IoT (англ. Internet of Things) – Інтернет речей.

JWT (англ. JSON Web Token) – маркер доступу у форматі об'єктної нотації JavaScript.

МааS (англ. Mobility as a Service) – мобільність як послуга.

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

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1. АНАЛІЗ ТЕОРЕТИКО-ТЕХНОЛОГІЧНИХ ЗАСАД ТА ПРОГРАМНОГО ІНСТРУМЕНТАРІЮ РЕАЛІЗАЦІЇ МААС-ПЛАТФОРМ.....	11
1.1 Теоретико-методологічні засади проектування МaaS-платформ.....	11
1.2 Аналіз світового досвіду та інструментарію програмної реалізації МaaS-платформ.....	13
1.3 Теоретико-технологічні основи застосування технології блокчейн та її класифікація	15
1.4 Технологічні виклики та обмеження інтеграції технології блокчейн у МaaS-платформи.....	17
1.5 Порівняльний аналіз програмних платформ розподіленого реєстру для інфраструктури «розумного міста»	19
1.6 Висновок до першого розділу	21
РОЗДІЛ 2. ПРОГРАМНО-ТЕХНОЛОГІЧНА РЕАЛІЗАЦІЯ ТА МОДЕЛЮВАННЯ АРХІТЕКТУРИ МААС-ПЛАТФОРМИ.....	22
2.1 Функціональна архітектура МaaS-платформи	22
2.2 Структура МaaS-платформи «розумного міста».....	24
2.2.1 Специфікація клієнтського рівня архітектури МaaS-платформи	25
2.2.2 Специфікація рівня збереження даних на базі технології блокчейн.....	26
2.2.3 Технологічний стек та специфікація серверної частини МaaS-платформи.....	30
2.2.4 Специфікація підсистеми фінансових операцій та інтеграції платіжного шлюзу.....	31
2.3 Модель даних прототипу МaaS-платформи «розумного міста»	32
2.4 Ланцюговий код МaaS-платформи «розумного міста».....	33

2.5 Компонентно-модульна структура та багаторівнева архітектура серверної частини МaaS-платформи «розумного міста»	35
2.6 Висновок до другого розділу	38
РОЗДІЛ 3. ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ ПРОГРАМНО-ТЕХНОЛОГІЧНИХ РІШЕНЬ ДЛЯ «РОЗУМНОГО МІСТА»	39
3.1 Розгортання та апаратно-програмне забезпечення експериментального середовища	39
3.2 Функціональне тестування прототипу МaaS-платформи «розумного міста».....	40
3.3 Дослідження продуктивності та навантажувальне тестування елементів МaaS-платформи «розумного міста»	45
3.4 Налаштування мережі блокчейн для прототипу МaaS-платформи «розумного міста».....	49
3.5 Порівняльний аналіз з іншими інноваційними технологіями	53
3.6 Перспективи подальших досліджень у сфері інтеграції МaaS систем.....	56
3.7 Висновок до третього розділу	58
РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ	59
4.1 Державна система моніторингу довкілля як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн	59
4.2 Вимоги ергономіки до організації робочого місця оператора ПК.....	62
4.3 Висновок до четвертого розділу	65
ВИСНОВКИ.....	66
ПЕРЕЛІК ДЖЕРЕЛ	68

ВСТУП

Актуальність теми. Стрімкі процеси урбанізації та зростання чисельності міського населення зумовлюють критичне навантаження на транспортну інфраструктуру сучасних мегаполісів, що вимагає переходу від традиційних моделей управління до концепції «розумного міста». У цьому контексті особливої ваги набуває парадигма мобільності як послуги (MaaS), яка інтегрує різноманітні види громадського, приватного та шерингового транспорту в єдину цифрову екосистему. Проте ефективне функціонування таких платформ стримується децентралізованим характером ринку перевезень, відсутністю взаємної довіри між операторами та складністю забезпечення конфіденційності користувацьких даних. Відтак, виникає гостра науково-практична потреба в дослідженні та обґрунтуванні новітніх засобів програмної реалізації, здатних забезпечити надійну координацію мультимодальних транспортних потоків.

Одним із найбільш перспективних інструментів для подолання зазначених викликів є інтеграція технології блокчейн у структуру серверної частини MaaS-платформ. Застосування розподіленого реєстру дозволяє побудувати прозоре, захищене та стійке до маніпуляцій середовище для збереження даних і виконання фінансових операцій без залучення сторонніх посередників. Розробка та впровадження смартконтрактів (ланцюгового коду) дає змогу автоматизувати процеси взаєморозрахунків, перевірки ідентифікаторів користувачів та розподілу доходів між транспортними провайдерами в режимі реального часу. Водночас вибір конкретного технологічного стеку та оптимізація системних параметрів, як от індексування СКБД CouchDB чи налаштування інтервалів формування блоків, залишаються малодослідженими і потребують детального аналітичного оцінювання.

Актуальність цього напряму дослідження підтверджується також необхідністю аналізу метрик продуктивності та навантажувального тестування прототипів подібних систем перед їх розгортанням у промисловому

середовищі. Визначення часових показників відгуку платформи залежно від щільності конкурентних запитів та обсягу накопичених у реєстрі транзакцій CRUD є базовою умовою для забезпечення масштабованості міських сервісів. Системний огляд, порівняльний аналіз наявних програмних рішень, а також експериментальне вимірювання характеристик швидкодії обробки даних відкривають можливості для створення вискоелективної архітектури інтелектуальної мобільності. Це дозволить суттєво підвищити рівень інтероперабельності міських служб та забезпечити сталий екологічно безпечний розвиток транспортної системи «розумного міста». Тому засоби програмної реалізації для MaaS-платформ «розумних міст» є актуальним напрямком сучасних досліджень в галузі комп'ютерних наук.

Мета і задачі дослідження. Метою даної кваліфікаційної роботи освітнього рівня «Бакалавр» полягає в системному аналізі, обґрунтуванні та експериментальному оцінюванні засобів програмної реалізації архітектури MaaS-платформи «розумного міста» на базі технології блокчейн для підвищення ефективності, захищеності та інтероперабельності процесів координації мультимодальних транспортних послуг. Для досягнення поставленої мети потрібно виконати ряд завдань, зокрема:

- Дослідити теоретичні засади MaaS-платформ та світовий досвід їх програмної реалізації в «розумних містах».
- Проаналізувати можливості, обмеження та технології для інтеграції в транспортні системи.
- Спроекувати структурно-функціональну архітектуру MaaS-платформи «розумного міста».
- Спроекувати модель даних MaaS-платформи «розумного міста».
- Провести тестування для оцінювання метрик продуктивності MaaS-системи «розумного міста».

Практичне значення одержаних результатів полягає у розробці працездатного прототипу та архітектурних специфікацій MaaS-платформи, а також у визначенні оптимальних конфігурацій індексування СКБД CouchDB та

параметрів блокчейн-мережі, які можуть безпосередньо використовуватися ІТ-компаніями й муніципальними службами для розгортання та масштабування високонавантажених транспортних сервісів у реальній інфраструктурі «розумних міст».

РОЗДІЛ 1. АНАЛІЗ ТЕОРЕТИКО-ТЕХНОЛОГІЧНИХ ЗАСАД ТА ПРОГРАМНОГО ІНСТРУМЕНТАРІЮ РЕАЛІЗАЦІЇ МААS-ПЛАТФОРМ

1.1 Теоретико-методологічні засади проєктування МааS-платформ

У сучасних «розумних містах» інтеграція інноваційних технологій у міську інфраструктуру є критично важливою умовою для оптимізації транспортних послуг та ефективного забезпечення потреб у пасажирських і вантажних перевезеннях [1]. Одним із ключових аспектів цієї трансформації є розбудова мультимодальних транспортних систем, які об'єднують різні види транспорту в єдиний комплекс. Така інтеграція дає змогу підвищити рівень мобільності, забезпечити відстежуваність процесів та покращити координацію. У цьому контексті технологія блокчейн виступає як перспективне рішення, що забезпечує прозору платформу для інтенсифікації взаємодії між різними постачальниками послуг та користувачами у транспортній галузі.

Метою дослідження [1] є аналіз процесів розробки та впровадження МааS-систем. Зокрема, досліджується здатність блокчейн-технологій підвищувати рівень безпеки, прозорості та операційної ефективності сервісів міської мобільності з особливим акцентом на підсистемах розрахунків та продажу квитків. Ключова гіпотеза полягає в тому, що децентралізована архітектура блокчейну дасть можливість підвищити цілісність транзакцій та рівень кооперації між стейкхолдерами, формуючи ефективнішу та захищенішу інформаційно-технологічну платформу міського транспорту. Зазначені системи орієнтовані переважно на пасажирські перевезення, проте володіють потенціалом для управління логістикою та вантажопотоками. Особливу увагу приділено трансформації методів оплати та квиткових рішень завдяки забезпеченню взаємодії між користувачами та різними операторами сервісів: автобусного та залізничного сполучення, систем спільного використання велосипедів, логістичних компаній та служб замовлення таксі. Інтеграція системи такого класу в інфраструктуру «розумного міста» не лише підвищує

зручність для кінцевого користувача, а й сприяє досягненню глобальних цілей екологічної сталості та посилення безпеки даних.

Зважаючи на зростання суспільної потреби в захищених, прозорих та ефективних міських транспортних системах, у межах різноманітних ініціатив досліджуються можливості оптимізації МaaS-платформ за допомогою технології блокчейн. Повноважним прикладом є проєкт DELPHI, який фінансується Європейським Союзом у межах програми Horizon Europe та спрямований на розвиток міської мобільності й логістики за допомогою інноваційних технологій. Ця ініціатива орієнтована на створення уніфікованої транспортної системи, що інтегрує декілька видів пасажирського та вантажного транспорту. Завдяки впровадженню блокчейну у проєкті [1] передбачається підвищити рівень захисту даних, оптимізувати взаємодію між учасниками ринку та вдосконалити транспортну логістику. Науковий вектор проєкту зосереджений на створенні блокчейн-платформи для безпечного керування даними між різними видами транспорту. Центральним елементом є інтеграція платіжних і квиткових систем у структуру міської мобільності, що робить безпосередній внесок у формування захищеної, прозорої та ефективної архітектури МaaS, передбаченої концепцією [2].

Для комплексного розуміння потенціалу блокчейну в межах концепції МaaS необхідно проаналізувати як загальний стан розвитку технології розподіленого реєстру, так і поточні виклики її інтеграції в мобільні системи. Проблеми масштабованості, безпеки та міжсистемної взаємодії залишаються суттєвими бар'єрами, проте фундаментальні характеристики блокчейну – децентралізація, прозорість та незмінність даних – позиціонують його як спроможний інструмент для подолання цих перешкод.

Основна дослідження [1] присвячена проєктуванню та аналізу архітектури МaaS-системи, що функціонує на базі технології блокчейн. При цьому детально розглядаються технічні аспекти програмно-алгоритмічного рішення та його модель даних. Також висвітлено шляхи вирішення проблем масштабованості для забезпечення технічної стійкості системи. Сучасні МaaS-

системи стикаються з низкою викликів, серед яких вразливості системи безпеки, обмеження масштабованості та операційна неефективність. Запропонована блокчейн-платформа безпосередньо нівелює ці проблеми, створюючи децентралізоване, захищене та масштабоване рішення для міської мобільності [1]. Завдяки використанню приватної архітектури Hyperledger Fabric із розмежуванням прав доступу та її модульних механізмів консенсусу, платформа суттєво підвищує рівень безпеки даних і швидкість обробки транзакцій, демонструючи стабільну продуктивність за умов змінного навантаження. На початковому етапі функціонал платформи буде сфокусовано на пасажирських перевезеннях, проте подальші дослідження передбачають розширення її архітектури на сферу вантажної логістики. Це повною мірою корелює з глобальною метою проєкту [1] щодо створення уніфікованої та високоефективної транспортної інформаційно-технологічної платформи.

1.2 Аналіз світового досвіду та інструментарію програмної реалізації МaaS-платформ

У [1] досліджено концепцію МaaS та її потенціал щодо трансформації сучасної транспортної інфраструктури. Основний акцент зміщено з володіння приватними транспортними засобами на використання спектра мобільних послуг, що надаються за запитом. Автори наголошують, що така архітектура здатна інтегрувати та оптимізувати також і вантажні перевезення. Модель МaaS консолідує різні транспортні сервіси в межах уніфікованої, орієнтованої на користувача платформи. Технологія блокчейн дає змогу вдосконалити МaaS-системи завдяки забезпеченню захищеного та прозорого керування транзакціями й інформаційними потоками між різними видами транспорту. Додатково, використання смартконтрактів та токенизації спрощує процеси системної інтеграції та створює механізми стимулювання користувачів у межах МaaS-інфраструктури «розумного міста».

Особливості застосування технології блокчейн для програмної реалізації МaaS-платформ розглянуто у праці [3], де особливу увагу приділено оцінці продуктивності систем за умов варіювання навантаження. Автори аналізують дві поширені та технологічно зрілі блокчейн-платформи: мережу Ethereum, що функціонує як публічний (відкритий) реєстр, та інфраструктуру «Hyperledger Fabric», яка є представником приватних систем із розмежуванням прав доступу. Платформа Ethereum характеризується децентралізованим консенсусом та широкою підтримкою з боку розробників, проте її публічна архітектура зумовлює проблеми з масштабованістю та низьку швидкість обробки транзакцій, що обмежує її застосування у високонавантажених середовищах, як-от системи міської мобільності. Натомість «Hyperledger Fabric» пропонує закриту архітектуру мережі з можливістю гнучкого налаштування механізмів консенсусу. Це робить її найбільш адаптованою до вимог суворого контролю, конфіденційності та високої продуктивності, що є критично важливим для МaaS-систем. Попри здатність обох платформ підтримувати функціонал МaaS, для практичної реалізації у [1] було обрано «Hyperledger Fabric» завдяки її модульності, масштабованості та відповідності корпоративним стандартам безпеки.

У дослідженні [4] запропоновано альтернативний підхід, що базується на використанні блокчейн-інфраструктури «Hyperledger Indy» [5] для створення багаторівневої архітектури ринку даних інтелектуальної мобільності (BSMD). Зазначений підхід вирішує критичні питання конфіденційності, безпеки, адміністрування та масштабованості в середовищі міського транспорту. Автори вказують на вразливості класичних методів збереження даних, зокрема, централізованих серверів, які схильні до кібератак та несанкціонованого розповсюдження інформації. Архітектура BSMD забезпечує захищений і прозорий обмін зашифрованими даними між учасниками ринку через блокчейн, де всі транзакції регламентуються взаємоузгодженими правилами.

1.3 Теоретико-технологічні основи застосування технології блокчейн та її класифікація

Технологія блокчейн, концептуально запропонована у 2008 році [6] та практично реалізована на початку 2009 року, функціонує як розподілений реєстр, що фіксує всі підтвержені транзакції у вигляді хронологічного ланцюжка блоків, який безперервно зростає. Серед фундаментальних характеристик блокчейну слід виділити децентралізацію, можливість аудиту, стійкість до несанкціонованих змін (персистентність) та анонімність. Функціонування архітектури блокчейну забезпечується інтеграцією передових рішень, як от криптографічне хешування, асиметричне шифрування для формування цифрових підписів та розподілені механізми консенсусу. Синергетичне використання цих технологій надає можливість реалізовувати децентралізовані транзакційні процеси, що забезпечує зниження накладних витрат та підвищення загальної ефективності системи [1].

Інформація в блокчейні структурується у вигляді зв'язаної послідовності блоків, де кожен наступний елемент містить криптографічне посилання на попередній, що гарантує захищеність даних математичними методами. Вузли мережі взаємодіють на основі топології P2P, завдяки чому нівелюється ризик виникнення єдиної точки відмови. Узгодженість даних у мережі забезпечується протоколами консенсусу, які гарантують ідентичність послідовності блоків та транзакцій на всіх вузлах, підтримуючи цілісність і надійність розподіленої бази даних [1]. При цьому кожна транзакція в межах блокчейну є дискретним записом у структурі загального розподіленого реєстру [7].

Якісним стрибком у розвитку технології розподіленого реєстру, етап Блокчейн 2.0, стала поява смартконтрактів як інтегральної складової блокчейн-платформ. Смартконтракти є програмними кодами, що виконуються всередині блокчейн-мережі. Вони дають можливість формалізувати тригери, умови та бізнес-логіку, забезпечуючи автоматизоване виконання складних програмованих транзакцій [1]. Варто зазначити, що концепт смартконтракту під

цією назвою був уперше сформульований фахівцем у галузі комп'ютерних наук та криптографії Ніком Сабо задовго до створення безпосередньо технології блокчейн [8].

Зважаючи на те, що Bitcoin є найбільш відомим прикладом практичного втілення технології розподіленого реєстру, його часто помилково ототожнюють із самим поняттям блокчейну. Проте сфера застосування блокчейну є значно ширшою за ринок криптовалют. Головна перевага цієї технології полягає в можливості здійснення транзакцій без залучення довірених посередників, що дає змогу ефективно впроваджувати її у фінансові сервіси, зокрема в системи електронних платежів, управління цифровими активами та грошових переказів [1]. Поява смартконтрактів суттєво розширила перелік галузей, які отримують технологічні переваги від децентралізації. Серед нових сфер застосування блокчейну поза межами цифрових фінансів варто виділити [9]:

- «розумну» логістику;
- «розумне» управління ланцюгами постачань;
- інтернет речей (IoT) [10];
- «розумну» охорону здоров'я;
- системи автентифікації особи;
- обробку авторських винагород;
- протидію відмиванню коштів;
- «розумні» виборчі системи;
- «розумний» аграрний сектор;

За архітектурними особливостями організації доступу блокчейн-системи поділяють на два основні типи: приватні – із розмежуванням прав доступу та публічні – відкриті.

У приватних блокчейн-системах право на участь у мережі мають лише автентифіковані користувачі, чії ідентифікатори та рівні доступу адмініструються централізованим органом або консорціумом. Такий підхід забезпечує вищий рівень конфіденційності та продуктивності, проте вимагає довіри до суб'єкта, який керує правами доступу. Типовими прикладами

програмних платформ приватного типу є «Hyperledger Fabric» та «R3 Corda». Натомість публічні блокчейни є загальнодоступними, що надає будь-якому вузлу можливість долучитися до верифікації транзакцій. Ця відкрита модель, яскраво проілюстрована мережами Bitcoin та Ethereum, гарантує максимальний рівень децентралізації, проте може характеризуватися нижчою пропускнуою здатністю та обмеженим контролем над видимістю даних, що робить її доцільною для публічних сценаріїв, які не потребують взаємної довіри сторін.

Крім того, технологія блокчейн володіє високим потенціалом для трансформації інших прикладних сфер, зокрема процесів дистрибуції цифрового мистецтва, логістичних ланцюгів, дистанційного надання послуг у сфері «розумного» туризму, а також створення інноваційних периферійних платформ, що зміщують обчислювальні процеси безпосередньо до джерел даних та розподілених систем верифікації повноважень [1].

1.4 Технологічні виклики та обмеження інтеграції технології блокчейн у MaaS-платформи

У процесі системної інтеграції технології блокчейн у структуру громадського транспорту виникає необхідність вирішення низки критичних викликів, серед яких: обмеження масштабованості, забезпечення міжсистемної взаємодії між різнорідними платформами, адаптація до динамічних нормативно-правових вимог, висока початкова вартість впровадження, а також потреба в навчанні користувачів і забезпеченні суспільного сприйняття нової технології. Зокрема, обробка значних обсягів транзакцій, що є характерною ознакою систем масового пасажирського транспорту, залишається суттєвою проблемою для більшості наявних інформаційно-технологічних архітектурних рішень на базі блокчейну [1]. Постійне зростання пасажиропотоку вимагає такої швидкості опрацювання транзакційних запитів, яка наразі перевищує технічні можливості переважної більшості розгорнутих блокчейн-мереж.

Водночас різні транспортні відомства та оператори ринку можуть використовувати гетерогенні (різномірні) програмні платформи, які є взаємно несумісними, причому ці рішення не обов'язково базуються на розподілених реєстрах. Відсутність єдиної стандартизації таких систем створює штучні бар'єри для безперешкодного обміну інформацією та спільного надання сервісів у межах МaaS-інфраструктури [11].

Регуляторна база, що регламентує застосування технології розподіленого реєстру, на сьогодні перебуває на етапі становлення. Через це оператори громадського транспорту можуть стикатися з правовими ризиками у сфері забезпечення конфіденційності персональних даних, інформаційної безпеки та відповідності чинному законодавству [1]. Необхідність адаптації до цих динамічних нормативних правил здатна суттєво уповільнити процеси практичного впровадження інноваційних технологічних рішень.

Попри те, що блокчейн володіє високим потенціалом для зниження операційних витрат у довгостроковій перспективі, капітальні інвестиції, необхідні на початкових етапах для розробки програмного забезпечення, модернізації наявної інфраструктури та перекваліфікації персоналу, є досить високими. Цей фінансовий чинник може стати стримувальним фактором для інтеграції блокчейн-рішень дрібними транспортними підприємствами та локальними операторами [12].

Зрештою, успішне функціонування квиткових та транспортних систем на базі блокчейну можливе лише за умови готовності кінцевих користувачів до освоєння новітніх технологій. У зв'язку з цим виникає об'єктивна потреба в організації обов'язкових програм технічної підтримки та навчання, що призводить до додаткового підвищення загальної вартості реалізації проєкту [1].

1.5 Порівняльний аналіз програмних платформ розподіленого реєстру для інфраструктури «розумного міста»

Серед найбільш відомих технологій блокчейн виділяють «Hyperledger Fabric», «Ethereum», «R3 Corda» та «Bitcoin». Кожне з цих програмних рішень володіє специфічними характеристиками, визначеними для застосування в промислових секторах, зокрема, логістиці, банківській справі та транспортних системах «розумного міста».

Платформа «Bitcoin», як перший представник публічних блокчейнів без обмеження доступу, забезпечує децентралізовані цифрові транзакції між користувачами, проте гарантує лише обмежену конфіденційність даних у критичних аспектів взаємодії. Механізм консенсусу «proof-of-work», покладений в основу «Bitcoin», здатний обробляти лише близько семи транзакцій за секунду та потребує значних енергетичних витрат, що унеможлиблює його застосування у високонавантажених системах [13]. Крім того, обмежені можливості базової мови скриптів «Bitcoin» перешкоджають розробці складних смартконтрактів, звужуючи потенційні сценарії його індустріального використання [1].

Натомість «Hyperledger Fabric» характеризується закритою архітектурою мережі з регламентованим доступом та модульною структурою, завдяки чому забезпечує вищі показники масштабованості та конфіденційності. Наявність таких інструментів, як ізольовані канали та колекції приватних даних, дає можливість здійснювати конфіденційні транзакції виключно між авторизованими суб'єктами, надійно захищаючи стратегічну інформацію. Модульний механізм консенсусу, що підключається за принципом плагіна, додатково підвищує пропускну здатність та спрощує адаптацію платформи під конкретні вимоги організацій [14].

Платформа «Ethereum» функціонує як публічний блокчейн, де транзакційні дані є відкритими за своєю природою. Це є суттєвим недоліком для галузей, які вимагають суворого дотримання конфіденційності [15]. Хоча

архітектурне рішення «Enterprise Ethereum» намагається нівелювати проблеми приватності в корпоративних мережах, воно суттєво ускладнює загальну структуру системи і поступається за гнучкістю інструментам контролю доступу «Hyperledger Fabric» [16]. Крім того, жорстка прив'язка «Ethereum» до мови програмування «Solidity» для розробки смартконтрактів може створити додатковий технологічний бар'єр для розробників, які звикли до класичних мов програмування [17].

Платформа «R3 Corda» спроектована переважно для фінансового сектору, де пріоритетом є відповідність нормативним вимогам та приватність, що досягається шляхом обмеження видимості транзакцій лише колом безпосередніх учасників угоди [18]. Хоча такий підхід посилює безпеку, адаптація «Corda» для масштабніших прикладних завдань може вимагати значної модифікації її архітектури [1]. Її специфічний для кожної окремої транзакції механізм консенсусу гарантує локальну ефективність, проте може викликати проблеми з масштабованістю у великих за розміром та структурою МaaS-платформах.

Зважаючи на наведені відмінності, «Hyperledger Fabric» є найбільш оптимізованим інструментом для побудови транспортних систем «розумного міста» завдяки гнучким налаштуванням приватності, високій масштабованості та можливостям інтеграції. Порівняно з «Ethereum» та «R3 Corda», її ключовими архітектурними перевагами є:

- Тонконалаштовувані інструменти конфіденційності – використання ізольованих каналів та технології приватної вибірки даних гарантує, що доступ до чутливої інформації отримають лише авторизовані учасники процесу, що є обов'язковою вимогою для секторів із високим рівнем відповідальності.

- Висока продуктивність та масштабованість: архітектурні особливості Fabric дають змогу підтримувати значну пропускну здатність транзакцій, забезпечуючи стабільне функціонування інтенсивних великомасштабних інфраструктурних об'єктів.

– Модульний та гнучкий дизайн: розробники мають можливість обирати оптимальні механізми консенсусу та базові мови для написання смартконтрактів, що надає можливість адаптувати кожне розгортання мережі під конкретні практичні завдання.

– Інтеграція в програмно-алгоритмічне середовище «розумного міста»: сумісність із поширеними класичними мовами програмування та стандартизованими інтерфейсами програмування застосунків (API) спрощує процес впровадження, мінімізуючи часові та фінансові витрати на етапі розгортання платформи.

Зазначені характеристики позиціонують «Hyperledger Fabric» як раціональний технологічний стек для ініціатив у межах концепції «розумного міста», де критеріями ефективності виступають операційна швидкість, захист даних та міжсистемна взаємодія.

1.6 Висновок до першого розділу

В першому розділі кваліфікаційної роботи подано теоретико-методологічні засади проектування MaaS-платформ. Проаналізовано світовий досвід та інструментарій програмної реалізації MaaS-платформ. Висвітлено теоретико-технологічні основи застосування технології блокчейн та її класифікація. Проаналізовано технологічні виклики та обмеження інтеграції технології блокчейн у MaaS-платформи. Подано порівняльний аналіз програмних платформ розподіленого реєстру для інфраструктури «розумного міста».

РОЗДІЛ 2. ПРОГРАМНО-ТЕХНОЛОГІЧНА РЕАЛІЗАЦІЯ ТА МОДЕЛЮВАННЯ АРХІТЕКТУРИ МААС-ПЛАТФОРМИ

2.1 Функціональна архітектура Маас-платформи

Основною метою Маас-платформи [1] є консолідація розрахункових операцій та платежів, що здійснюються на користь муніципальних і приватних транспортних компаній міста, у межах єдиного програмного інтерфейсу. Така інтеграція також забезпечує можливість наскрізного відстежування параметрів поїздок та формування аналітичних статистичних даних, необхідних як для оптимізації діяльності транспортних операторів, так і для інформування пасажирів. На рисунку 2.1 подано діаграму прецедентів даного програмного рішення у структурі «розумного міста».

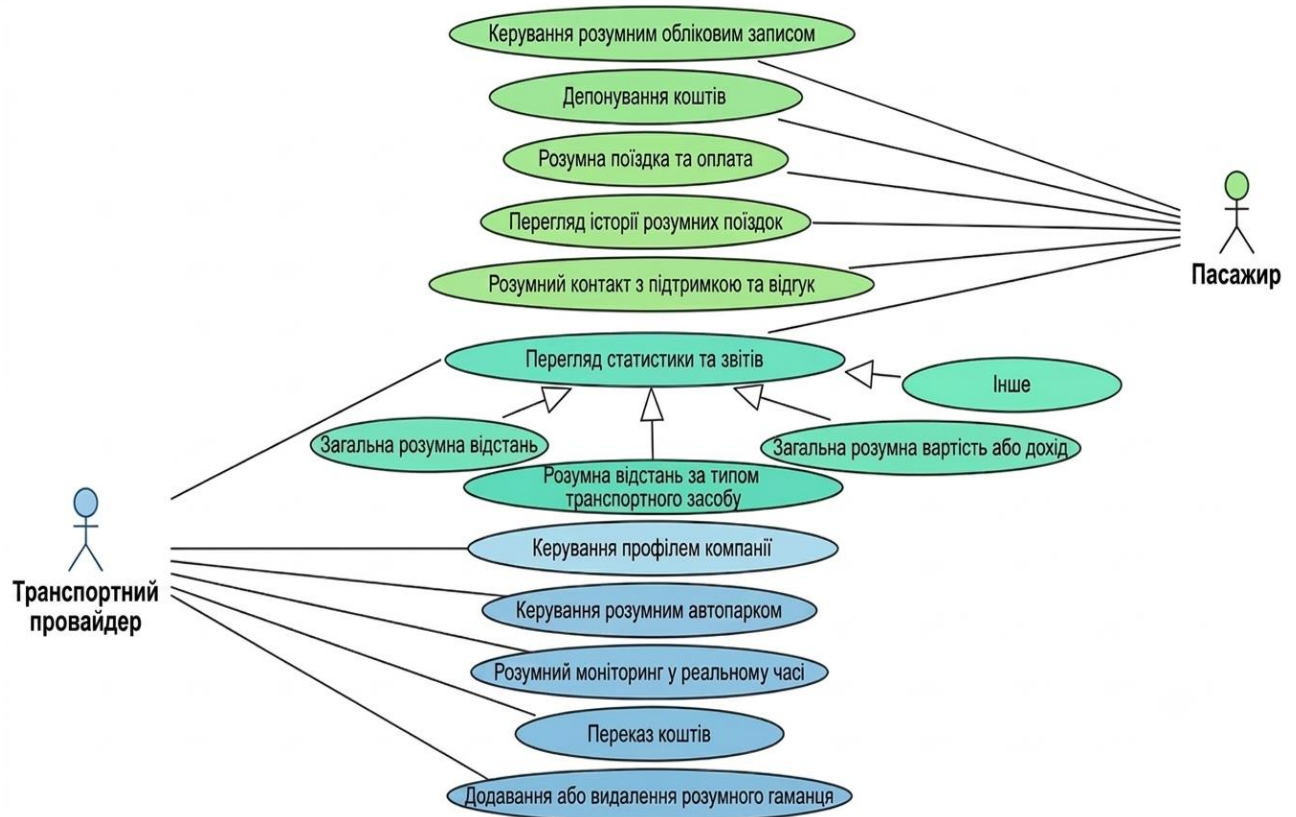


Рисунок 2.1 – Діаграма варіантів використання інформаційно-технологічної
Маас-платформи

До основних акторів МaaS-платформи належать постачальники транспортних послуг – транспортні компанії, а також суб'єкти, які використовують транспортну мережу для переміщення – фізичні особи або вантажі, що можуть транспортуватися громадським транспортом. Оскільки ці два класи користувачів мають різні рівні доступу до функціональних можливостей МaaS-системи «розумного міста», їхня взаємодія з платформою реалізується через диференційовані інтерфейси або відповідні клієнтські застосунки.

Програмний застосунок користувача – пасажир, містить захищений віртуальний гаманець, баланс якого можна поповнювати – депонувати кошти, через зовнішні платіжні шлюзи. Цей гаманець забезпечує проведення внутрішніх розрахункових операцій у межах платформи, повністю нівелюючи потребу у використанні готівкових коштів або придбанні декількох різних квитків. Безпосередньо під час посадки у транспортний засіб пасажир здійснює оплату шляхом переказу необхідної суми на віртуальний гаманець постачальника транспортних послуг. Зовнішні платіжні оператори залучаються виключно для обробки процесів депонування та виведення безготівкових коштів із МaaS-системи «розумного міста» [1].

Через клієнтський інтерфейс пасажир має можливість у режимі реального часу отримувати інформацію про поточні поїздки, зокрема про рух за маршрутом та розрахунковий час прибуття. Також у застосунку зберігається хронологічна історія минулих поїздок для подальшого аналізу. Окрім цього, інтерфейс користувача дає можливість пасажиром оцінювати якість отриманих послуг, залишати відгуки та звертатися до служби технічної підтримки у разі виникнення позаштатних ситуацій.

Для обох категорій користувачів передбачено доступ до аналітичного модуля для перегляду статистики, зокрема: пройденої відстані за визначений проміжок часу, обсягу доходів або витрат за період, пробігу конкретного транспортного засобу, а також дистанції у розрізі типів рухомого складу тощо [1]. Наприклад, надання пасажиром інформації про екологічний тип

транспортного засобу може стимулювати їх надавати перевагу компаніям із екологічно чистішим автопарком під час планування майбутніх поїздок. З іншого боку, такі аналітичні дані допомагають постачальникам приймати обґрунтовані управлінські рішення щодо керування автопарком та підвищення загальної якості обслуговування.

Аналогічно до пасажирів, транспортні провайдери володіють одним або декількома віртуальними гаманцями, на які зараховуються кошти за надані послуги [1]. Вони можуть здійснювати моніторинг доходів, ініціювати операції виведення коштів та безпечно керувати всіма фінансовими транзакціями.

Окрім фінансового контуру, інтерфейс постачальників послуг дає змогу здійснювати моніторинг активних рейсів за участю їхнього рухомого складу, керувати запитами на поїздки та оперативно диспетчеризувати транспортні засоби для оптимізації процесів надання послуг у межах «розумного міста».

2.2 Структура МaaS-платформи «розумного міста»

На рисунку 2.2 подана структура МaaS-платформи «розумного міста» із виокремленням її ключових компонентів. До складу структурного стека входять клієнтські програмні застосунки користувачів, серверна частина («back-end») платформи, а також децентралізоване середовище розподіленого реєстру – блокчейн-мережі.

Архітектура МaaS-платформи «розумного міста», яка поєднує транспортні сервіси з блокчейн-мережею для забезпечення прозорості та безпеки. Ліва частина схеми відображає «розумні сервіси платформи»: застосунки для пасажирів і транспортних провайдерів, що взаємодіють через REST API. Тут показані ключові компоненти — вебконтролери, модель даних, JWT-автентифікація, черга повідомлень, обробники, Fabric SDK та сервіс платежів. Важливим елементом є інтеграція інтелектуального платіжного провайдера, що забезпечує безпечні транзакції.

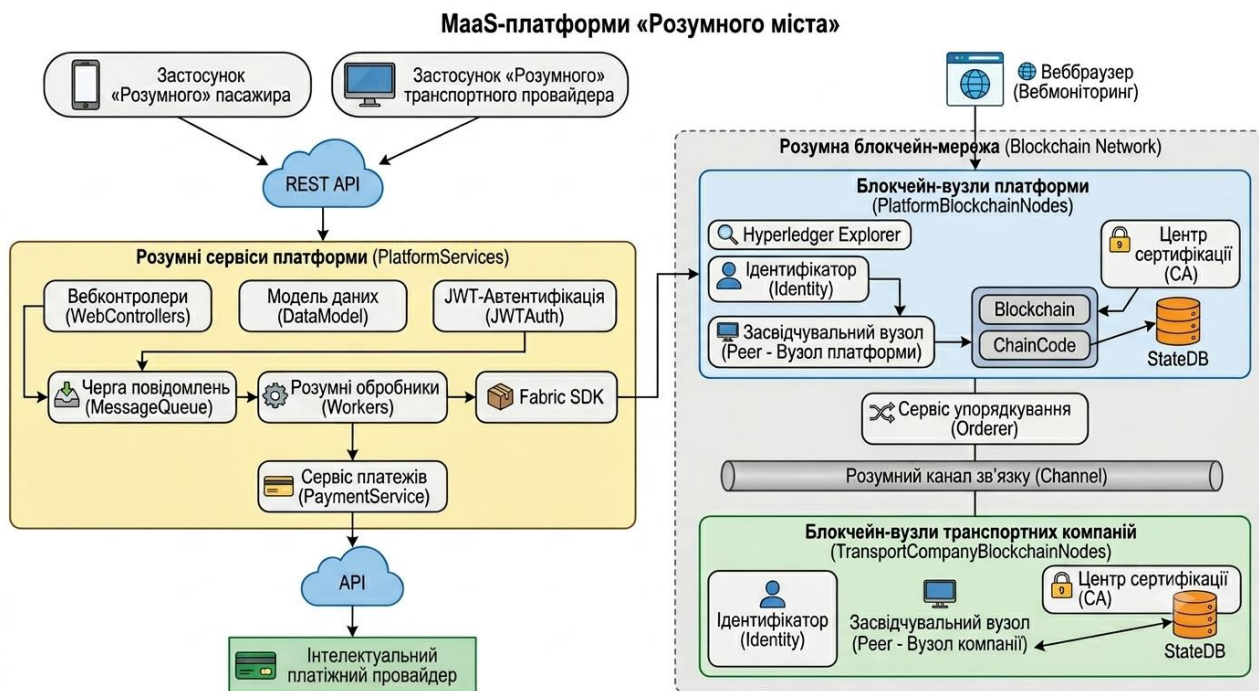


Рисунок 2.2 – Структура МaaS-платформи «розумного міста»

Права частина рисунка ілюструє «розумну блокчейн-мережу», яка складається з вузлів платформи та транспортних компаній. Вона інтегрує «Hyperledger Explorer», ідентифікатори, засвідчувальні вузли, сервіс упорядкування, канали зв'язку та бази даних «StateDB». Додатково показані центри сертифікації, що гарантують автентичність учасників. Така структура забезпечує децентралізований обмін даними між транспортними агентами, прозорість транзакцій і підвищений рівень кібербезпеки у «розумному місті».

2.2.1 Специфікація клієнтського рівня архітектури МaaS-платформи

Програмний комплекс МaaS-платформи «розумного міста» [1] містить два диференційовані клієнтські застосунки: перший розроблений безпосередньо для пасажирів, а другий – для постачальників транспортних послуг. Зазначені застосунки надають можливість користувачам ефективно взаємодіяти з ключовими функціональними модулями платформи. При цьому можна:

- здійснювати адміністрування віртуальних гаманців;

- проводити розрахункові операції;
- аналізувати статистичні дані щодо параметрів виконаних поїздок.

Завдяки впровадженню таких адаптованих програмних рішень архітектура МaaS-системи «розумного міста» [1] гарантує, що пасажери та транспортні провайдери отримують необхідний інструментарій для високоефективної взаємодії з платформою. Це дає можливість суттєво покращити користувацький досвід та забезпечує технологічну підтримку для впровадження сталіших практик міської мобільності в межах концепції «розумного міста».

2.2.2 Специфікація рівня збереження даних на базі технології блокчейн

Рівень збереження даних МaaS-системи «розумного міста» [1] побудований на основі технології блокчейн. Консорціум із декількох незалежних організацій забезпечує адміністрування та актуалізацію копій як баз даних поточного стану, так і журналів транзакцій розподіленого реєстру, що гарантує незмінність та високий рівень безпеки інформації. Реплікація даних між різними вузлами мережі підвищує відмовостійкість МaaS-системи «розумного міста» та нівелює ризики, пов'язані з наявністю єдиної точки відмови.

Вибір конкретної технології розподіленого реєстру є критично важливим аспектом для побудови ефективної та захищеної МaaS-платформи в межах концепції «розумного міста». На основі порівняльного аналізу існуючих рішень для реалізації корпоративного рівня було обрано платформу «Hyperledger Fabric 2.5». Комплексні профільні дослідження підтверджують доцільність її застосування в інфраструктурі «розумних міст» та транспортних системах. Зокрема, у [19] подано детальний огляд використання блокчейну в секторах громадського транспорту, де визначено ключові переваги «Hyperledger Fabric» щодо масштабованості, безпеки та конфіденційності. Крім того, автори [3]

дослідили специфіку інтеграції блокчейну в МaaS-платформи, продемонструвавши механізми підвищення рівня довіри та прозорості між усіма стейкхолдерами завдяки інструментам «Hyperledger Fabric».

До базових функціональних переваг, що зумовили вибір «Hyperledger Fabric» [1] як цільового блокчейн-рішення, належать:

- Мережа з регламентованим доступом та централізованим керуванням ідентифікацією: усі учасники МaaS-системи «розумного міста» є верифікованими, а їхня автентифікація реалізується на основі цифрових сертифікатів.

- Модульна та конфігурована архітектура – забезпечує можливість гнучкого підключення сторонніх реалізацій ключових компонентів, як от механізм консенсусу, база даних поточного стану та супутні мікросервіси [1].

- Висока пропускна здатність та масштабованість – реалізовано механізм паралельного виконання смартконтрактів – чейнкоду, та підтримку багатоканальних мережевих структур [20].

- Забезпечення приватності та конфіденційності транзакцій – обмін даними між визначеними контрагентами захищено за допомогою ізольованих каналів зв'язку та механізмів колекцій приватних даних [1].

- Ефективне виконання запитів та менеджмент даних – інтеграція баз даних класу «StateDB», наприклад, «CouchDB», дає змогу виконувати складні вибірки, забезпечує доступ до даних у режимі реального часу та спрощує аналітичне опрацювання [21].

Спрощена модель архітектури мережі містить дві організації-учасниці. Безпосередньо МaaS-платформа виступає як юридична особа, що здійснює загальний нагляд за функціонуванням МaaS-системи «розумного міста». Транспортна компанія функціонує як друга організація, яка забезпечує ендорсмент (засвідчення) транзакцій. Як продемонстровано в [1], кожна організація оперує власним засвідчувальним вузлом («peer»), базою даних поточного стану та локальним центром сертифікації.

Програмні модулі (смартконтракти), що описують бізнес-логіку та правила виконання транзакцій у межах блокчейн-мережі, у термінології «Hyperledger Fabric» визначаються як чейнкод («chaincode»). Саме цей код регламентує процедури взаємодії всередині консорціуму, забезпечуючи автоматизоване виконання транзакцій та модифікацію стану реєстру.

Засвідчувальний вузол («peer») є структурною одиницею (нодою) мережі, яка зберігає локальну копію розподіленого реєстру та розгорнуті смартконтракти. Вузли цього типу виконують критично важливі функції: валідують і підписують транзакції, ведуть облік операцій та беруть безпосередню участь у процесах досягнення консенсусу.

Центр сертифікації відповідає за менеджмент цифрових сертифікатів користувачів, зокрема, процеси їхньої реєстрації та відкликання [1]. Оскільки «Hyperledger Fabric» є мережею з регламентованим доступом, для встановлення прав, ролей та індивідуальних атрибутів користувачів застосовуються сертифікати стандарту «X.509». На основі наданих прав та відповідно до визначених ролей користувачі отримують можливість виконувати запити або ініціювати транзакції у відповідних каналах.

Розподілений реєстр («ledger») є одним із фундаментальних компонентів архітектури «Hyperledger Fabric», акумулюючи записи про всі транзакції (зміни станів) та загальний стан блокчейн-мережі. Рівень реєстру структурно поділяється на безпосередньо ланцюжок блоків («blockchain»), що є незмінним журналом операцій, та базу даних актуального стану – світовий стан («world state» або «StateDB»). Ланцюжок блоків є впорядкованим списком блоків із режимом доступу «тільки для запису», що виключає можливість модифікації ретроспективних даних, який містить повний лог усіх подій у мережі для забезпечення можливості проведення аудиту. Ланцюжок блоків зберігається безпосередньо у файловій системі кожного засвідчувального вузла. База світового стану («world state») відображає актуальні параметри всіх активів у реєстрі на поточний момент часу [1]. Вона реалізована як сховище типу «ключ-значення», де фіксуються останні змінені транзакціями значення об'єктів.

Такий розподіл надає можливість відокремити операції пошуку та зчитування даних від довготривалої історії транзакцій, що істотно підвищує продуктивність МaaS-системи «розумного міста» [1] без потреби обробки всього ланцюжка історичних блоків при кожному запиті. В інфраструктурі «Hyperledger Fabric» підтримуються різні замінні реалізації баз даних поточного стану, як от «LevelDB» або «CouchDB». Базова вбудована СУБД «LevelDB» орієнтована на прості запити за ключем. Альтернативне рішення «CouchDB» підтримує розширені аналітичні запити до документів у форматі JSON, а також складні механізми вибірки з використанням індексів [21].

Канал («channel») є ізольованою та конфіденційною лінією зв'язку між чітко визначеними учасниками консорціуму, яка використовується для розмежування транзакційних даних та забезпечення безпечного інформаційного обміну всередині окремого сегмента мережі.

Вузол служби упорядкування («orderer») виконує базову функцію структурування верифікованих транзакцій у блоки, забезпечує досягнення консенсусу між учасниками та здійснює розсилку сформованих блоків для їхнього подальшого внесення до розподіленого реєстру.

Інструмент «Hyperledger Explorer» є опціональним компонентом інформаційно-технологічної системи «Hyperledger Fabric», проте його інтеграція суттєво підвищує рівень прозорості мережі. Цей програмний засіб із відкритим вихідним кодом, створений у межах проєкту «Hyperledger» під егідою «Linux Foundation», призначений для моніторингу та візуалізації даних розподіленого реєстру. Платформа надає веб-інтерфейс, за допомогою якого адміністратори та розробники можуть ефективно отримувати доступ, виконувати запити та проводити аналіз транзакційних даних, активованих смартконтрактів та поточного статусу вузлів мережі «розумного міста» [22].

2.2.3 Технологічний стек та специфікація серверної частини MaaS-платформи

Серверна частина («back-end») MaaS-системи «розумного міста» реалізована мовою програмування Java з використанням програмно-алгоритмічного комплексу «Spring», який було обрано завдяки його розвиненому корпоративному функціоналу та широкому застосуванню під час розробки високонавантажених масштабованих застосунків [23]. Платформна незалежність Java та розгалужена інформаційно-технологічна система стандартних бібліотек є оптимальними для інтеграції складної бізнес-логіки та забезпечення високої продуктивності в MaaS-середовищі. Вибір програмно-алгоритмічного комплексу «Spring» зумовлений його гнучкістю, легковажністю архітектури та наявністю вбудованих інструментів для створення захищених масштабованих інтерфейсів REST API [24]. Ці інтерфейси є критично важливими для організації інформаційного обміну як із клієнтськими застосунками, так і з блокчейн-мережею «Hyperledger Fabric», оскільки в архітектурі останньої безпосередній доступ клієнтських застосунків до засвідчувальних вузлів («peers») є регламентованим та обмеженим. Для забезпечення стабільності операційних процесів серверна частина також інтегрує чергу повідомлень та платіжний шлюз.

Взаємодія серверної підсистеми з клієнтськими застосунками базується на використанні архітектурного стилю REST API із застосуванням текстового формату обміну даними JSON.

Для забезпечення програмного зв'язку між створеним API та розподіленим реєстром інтегровано інструментарій розробника «Java Fabric SDK 2.1» [25]. Базові операції, як от розрахунки за поїздки або перевірка поточного балансу гаманця, безпосередньо координуються інструментом API через «Fabric SDK». Натомість виконання складніших та обчислювально місткіших завдань, зокрема, агрегація та генерація аналітичної статистики, делегується асинхронній системі обміну повідомленнями («RabbitMQ»).

Під час надходження аналітичного запиту інтерфейс API реєструє його у відповідному сервісі черги повідомлень, після чого вільний фоновий обробник («worker») приймає повідомлення до виконання. Використання черги повідомлень дає можливість ефективно розподіляти завдання між декількома паралельними обробниками, реалізуючи балансування навантаження, забезпечуючи масштабованість і раціональне використання обчислювальних ресурсів, а також запобігаючи виникненню критичних затримок у MaaS-системи «розумного міста» [1].

2.2.4 Специфікація підсистеми фінансових операцій та інтеграції платіжного шлюзу

Сервіс платежів («Payment Service») є базовим компонентом архітектури MaaS-системи «розумного міста» [1], який забезпечує можливість безперешкодної інтеграції із зовнішніми платіжними провайдерами. Він пов'язує MaaS-платформу з фінансовими інституціями, даючи змогу користувачам безпечно виконувати депонування коштів, розрахункові операції та виведення безготівкових активів із MaaS-системи «розумного міста». Даний сервіс гарантує обробку транзакцій у режимі реального часу, забезпечуючи верифікацію та авторизацію всіх платежів відповідно до загальноприйнятих фінансових регламентів та стандартів безпеки, зокрема PCI-DSS.

Крім того, для забезпечення безпечного та безперебійного проведення розрахунків, платіжний шлюз здійснює шифрування конфіденційних фінансових даних під час їхнього транспортування, що мінімізує ризики шахрайства та несанкціонованого доступу. Підсистема також підтримує процеси токенизації, замінюючи вразливі реквізити платіжних карток унікальними ідентифікаторами (токенами), що надає можливість підтримувати високий рівень приватності користувачів та гарантує відповідність нормативним вимогам у межах інфраструктури «розумного міста» [1].

2.3 Модель даних прототипу МaaS-платформи «розумного міста»

На рисунку 2.3 подано візуалізацію моделі даних, що зберігається в розподіленому реєстрі блокчейн-мережі [1]. Схема деталізує архітектуру даних МaaS-системи «розумного міста», відображаючи ключові класи активів (сутностей) та логічні зв'язки між ними в інфраструктурі «розумного міста».

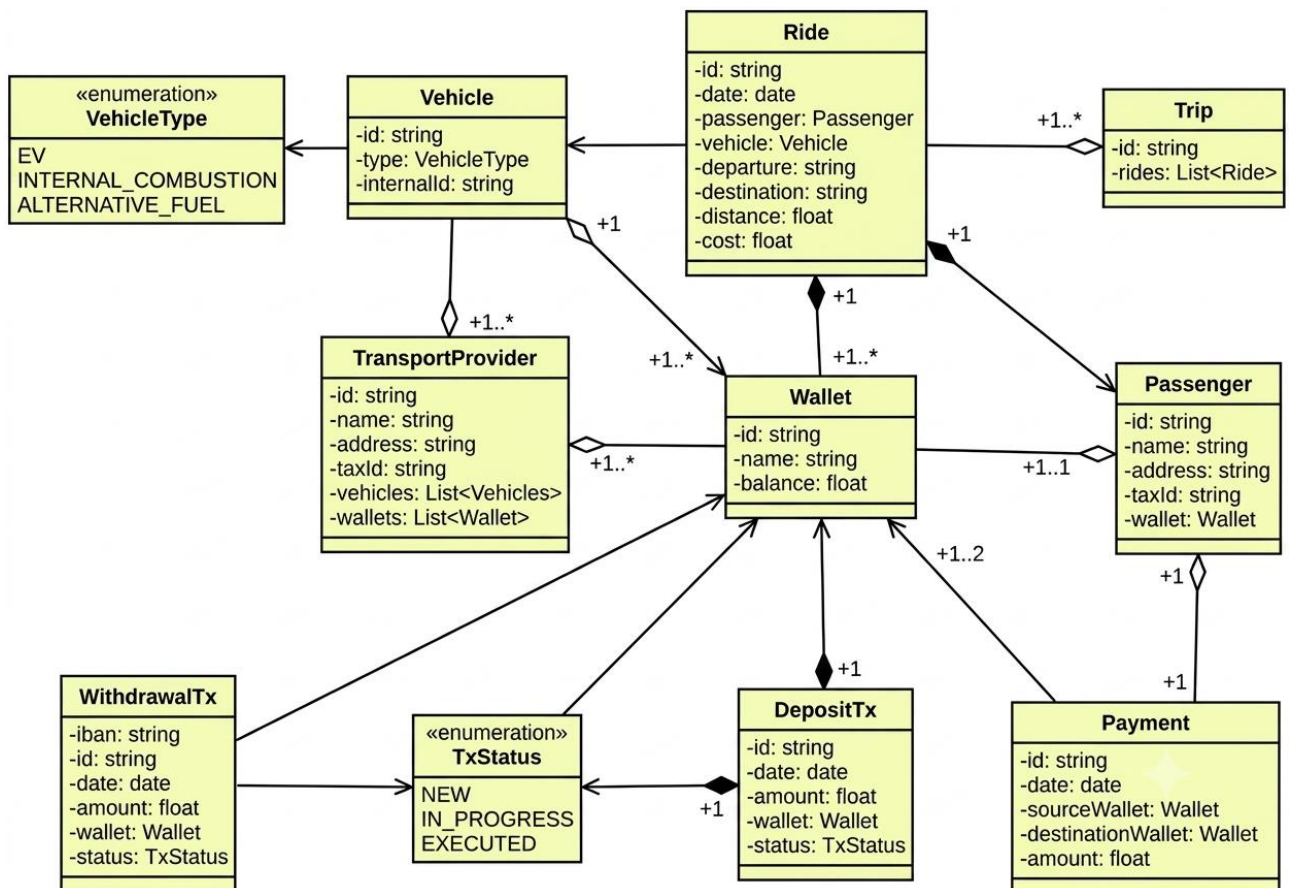


Рисунок 2.3 – Модель даних платформи [1]

Класи активів «TransportProvider» та «Passenger» призначені для агрегації та збереження облікових даних, що ідентифікують відповідно транспортні компанії – провайдери послуг, та безпосередніх користувачів транспортної мережі. Сутність «Vehicle» відповідає за збереження та систематизацію даних, пов'язаних із рухомим складом автопарку конкретної компанії. Опис типу рухомого складу базується на перерахуванні «VehicleType», яке класифікує транспортні засоби за трьома базовими категоріями: електромобілі (EV),

транспорт із двигунами внутрішнього згоряння та транспортні засоби на альтернативних видах палива.

Об'єкт реєстру «Wallet» (віртуальний гаманець) логічно пов'язаний зі своїм власником, у ролі якого може виступати або транспортне підприємство, або окрема фізична особа, та забезпечує адміністрування поточного балансу коштів. Під час виконання операцій депонування (внесення коштів) або виведення безготівкових активів здійснюється динамічне оновлення балансу із одночасною фіксацією відповідного транзакційного запису в MaaS-системі «розумного міста» [1]. Аналогічним чином, при здійсненні розрахунку за надані послуги, баланси обох залучених гаманців синхронно коригуються, а сама фінансова операція документується в розподіленому реєстрі.

Сутність «Ride» акумулює вичерпні параметри щодо кожного окремого факту надання транспортної послуги, зокрема: ідентифікатор пасажирів, задіяний транспортний засіб, дату та час здійснення поїздки, координати початкового й кінцевого пунктів маршруту, підсумкову вартість та пройдену дистанцію. Окрім цього, послідовна сукупність логічно пов'язаних між собою записів «Ride» групується в єдину структуру «Trip», яка транслює серію послідовних взаємопов'язаних поїздок у межах одного цілісного маршруту переміщення користувача в інфраструктурі «розумного міста».

2.4 Ланцюговий код MaaS-платформи «розумного міста»

Чейнкод («chaincode») у середовищі «Hyperledger Fabric» є функціональним еквівалентом смартконтрактів, що застосовуються в інших блокчейн-платформах. Він регламентує правила та логіку, які керують виконанням транзакцій у мережі [1].

На рисунку 2.4 зображена компонентна структура чейнкоду для запропонованого архітектурного рішення, програмне втілення якого виконано на базі платформи Node.js у межах концепції «розумного міста».

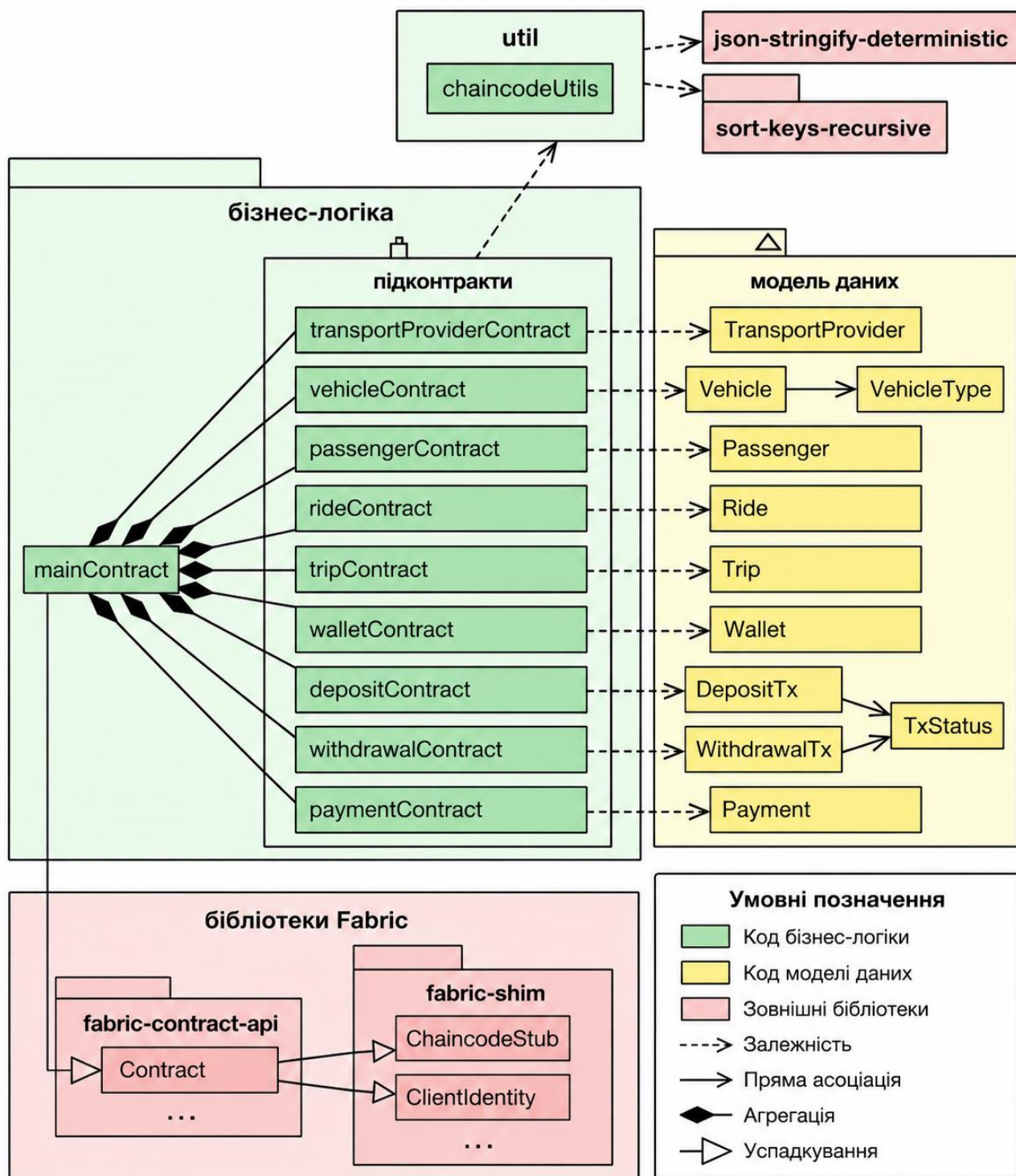


Рисунок 2.4 – Діаграма структури ланцюгового коду МaaS-платформи «розумного міста» [1]

Чейнкод, програмна реалізація якого може бути здійснена мовами Go, Node.js або Java, є застосунком, що функціонує відповідно до зумовленого інтерфейсу (API), наданого бібліотекою `fabric-contract-api` [1]. Іншою базовою бібліотекою для розробки та розгортання чейнкоду є `fabric-shim`, яка забезпечує

необхідний низькорівневий інструментарій для організації взаємодії між смартконтрактами, створеними за допомогою fabric-contract-api, та безпосередньо засвідчувальними вузлами («peers») інфраструктури «Hyperledger Fabric» [1]. Внутрішня бізнес-логіка чейнкоду декомпонована на декілька програмних модулів. Кожен модуль відповідає за реалізацію доступних операцій над певним класом активів, визначених у загальній моделі даних, наприклад, файл «vehicleContract.js» координує сутність «Vehicle», «rideContract.js» – сутність «Ride», «walletContract.js» – сутність «Wallet» тощо. Зазначені компоненти визначаються як модулі доменної логіки MaaS-системи «розумного міста».

Допоміжний модуль «chaincodeUtils.js» інтегрується модулями доменної логіки, оскільки він містить узагальнену (типізовану) реалізацію базових операцій маніпулювання даними, як от створення («create»), зчитування («read»), оновлення («update»), видалення («delete»), отримання всіх записів («getAll»), збереження масиву даних («saveAll»), перевірка наявності об'єкта («exists»), а також вибірка за селектором («getBySelector»), що застосовується для виконання складних аналітичних запитів.

Повний комплекс бізнес-логіки стає доступним для зовнішніх викликів через інкапсульований модуль фасаду «chainCode.js» [1]. Цей модуль консолідує всі публічні методи та операції, які підтримуються модулями доменної логіки, і реалізований у вигляді класу, що успадковує базовий клас «Contract» із системної бібліотеки fabric-contract-api.

2.5 Компонентно-модульна структура та багаторівнева архітектура серверної частини MaaS-платформи «розумного міста»

Як зазначалося раніше, серверна частина («back-end») платформи реалізована мовою програмування «Java 21» із використанням програмно-алгоритмічного комплексу «Spring Boot 3.1.4». На рисунку 2.5 подано модульну

архітектуру, розроблену для серверної підсистеми MaaS-системи «розумного міста».

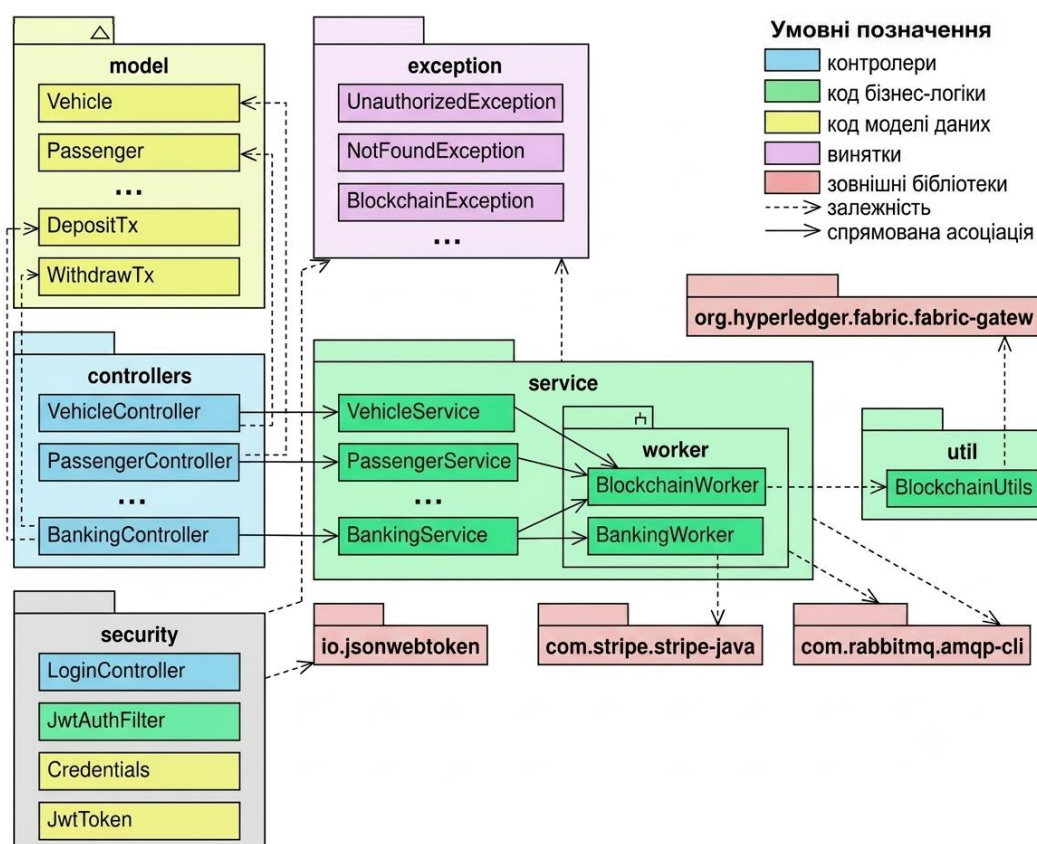


Рисунок 2.5 – Діаграма структури коду бекенду [1]

Наведена схема класифікує компоненти за окремими функціональними рівнями, відображаючи їхні ролі, системні залежності та логіку взаємодії. Дане архітектурне рішення базується на фундаментальних принципах проектування програмного забезпечення, що забезпечує масштабованість, піддатливість до модифікації та високий рівень безпеки MaaS-системи «розумного міста» [1].

Активи, які зберігаються в розподіленому реєстрі блокчейн-мережі, деталізовані на рис. 2.3, піддаються обробці за допомогою спеціалізованих компонентів у межах кожного функціонального рівня. Слід зауважити, що на рис. 2.5 відображено лише певну підмножину цих компонентів, безпосередньо пов'язаних із процесами маніпулювання активами в інфраструктурі «розумного міста» [1].

Рівень моделі даних інкапсулює фундаментальні інформаційні сутності платформи, які відображають доменні об'єкти МaaS-системи «розумного міста». Зазначені класи формують базис для збереження, структурування та маніпулювання даними на всіх етапах функціонування програмного комплексу.

Рівень контролерів виступає як вхідна точка для обробки мережових HTTP-запитів, делегуючи кінцеві точки архітектурного стилю REST API для взаємодії із зовнішніми клієнтами. Кожен окремий контролер перенаправляє запити до відповідного сервісного рівня для їхньої подальшої обробки, що гарантує чітке дотримання принципу розділення відповідальності.

Сервісний рівень акумулює в собі ключову бізнес-логіку програмного застосунку, забезпечуючи модульність та повторне використання функціоналу. Для оптимізації продуктивності сервіси делегують асинхронні, обчислювально місткі або тривалі за часом виконання завдання спеціалізованим модулям фонові обробки – воркерам («BlockchainWorker» та «BankingWorker»). Ці модулі відповідають за безпосередню інтеграцію з блокчейн-мережею та координацію платіжних операцій. Впровадження таких компонентів підвищує рівень модульності та масштабованості МaaS-системи завдяки розвантаженню основного сервісного рівня [1]. Декупінг – послаблення архітектурного зв'язку, між сервісами та фоновими обробниками завдань реалізовано за допомогою брокера черг повідомлень на базі «RabbitMQ».

Компоненти підсистеми безпеки керують механізмами автентифікації та авторизації в межах платформи. Даний рівень гарантує регламентований і захищений доступ до функціоналу МaaS-системи «розумного міста», підтримуючи цілісність користувацьких даних.

Спеціалізовані класи винятків визначені для коректної обробки специфічних системних помилок. Наявність цих винятків підвищує стійкість програмного забезпечення до відмов та спрощує трасування помилок завдяки наданню деталізованого зворотного зв'язку під час виконання процесів.

Для розширення базового функціоналу та оптимізації процесу розробки МaaS-платформа інтегрується із зовнішніми програмними бібліотеками.

Зокрема, модуль «BlockchainUtils» взаємодіє з інструментарієм розробника «Hyperledger Fabric Gateway SDK» для організації безпечного та ефективного інформаційного обміну з розподіленим реєстром. Стороння бібліотека «com.rabbitmq.amqp-client» керує чергами повідомлень під час асинхронної обробки даних, а компонент «com.stripe.stripe-java» забезпечує проведення захищених платіжних транзакцій [1]. Нарешті, бібліотека «io.jsonwebtoken» застосовується для генерації та валідації маркерів доступу «JSON Web Tokens» (JWT) з метою аутентифікації користувачів та безпечного менеджменту сесій в інфраструктурі «розумного міста».

2.6 Висновок до другого розділу

В другому розділі кваліфікаційної роботи сформовано функціональну архітектуру МaaS-платформи. Запропоновано структуру МaaS-платформи «розумного міста». Створено модель даних прототипу МaaS-платформи «розумного міста». Розглянуто ланцюговий код МaaS-платформи «розумного міста». Проаналізовано компонентно-модульну структуру та багаторівневу архітектуру серверної частини МaaS-платформи «розумного міста».

РОЗДІЛ 3. ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ ПРОГРАМНО-ТЕХНОЛОГІЧНИХ РІШЕНЬ ДЛЯ «РОЗУМНОГО МІСТА»

3.1 Розгортання та апаратно-програмне забезпечення експериментального середовища

Серверна частина («back-end») та інфраструктура блокчейн-мережі протестовано на хмарній віртуальній машині Amazon EC2 типу t3.large [1] із технічними характеристиками поданими в таблиці 3.1.

Таблиця 3.1 – Технічні характеристики хмарної віртуальної машини [1]

Тип віртуальної машини	Amazon EC2 t3.large.
Центральний процесор	Intel® Xeon® Platinum 8175 M CPU @ 2.50 GHz – 2 віртуальні процесори, 1 фізичне ядро з підтримкою технології Hyper-Threading.
Архітектура	64-розрядна – x86-64.
Оперативна пам'ять	8 ГБ RAM.
Операційна система	Ubuntu 24.04.1 LTS.
Версія ядра	Linux 6.8.0-1018-aws.
Дискова підсистема	20 ГіБ NVMe SSD – Amazon EBS.
Платформа віртуалізації	Amazon EC2.

Зазначена конфігурація дає можливість відтворити реальні умови хмарного розгортання, забезпечуючи репрезентативне середовище для дослідження та тестування процесів інтеграції технології розподіленого реєстру в Маас-платформи «розумного міста» [1]. Вибір обчислювального вузла типу t3.large обґрунтований необхідністю досягнення оптимального балансу між продуктивністю Маас-системи «розумного міста» та економічною доцільністю,

що гарантує достатню потужність процесора для виконання розрахунків при мінімізації витрат на інфраструктуру.

Навантажувальне та функціональне тестування програмного забезпечення [1] здійснювалося дистанційно з використанням персонального комп'ютера, конфігурація якого подана в таблиці 3.2.

Таблиця 3.2 – Технічні характеристики хмарної віртуальної машини [1]

Центральний процесор	Intel® Core™ i7-8565U CPU @ 1.80 GHz – 8 логічних ядер.
Архітектура	64-розрядна – x86-64.
Оперативна пам'ять	16 ГБ RAM.
Операційна система	Ubuntu 24.04.1 LTS.
Дискова підсистема	512 ГБ SSD.
Версія ядра	Linux 6.8.0-47-generic.

Проведення функціональних тестувань та імітаційного моделювання навантаження з віддаленого комп'ютера дало змогу симулювати реальні умови взаємодії кінцевих користувачів із МaaS-платформою «розумного міста» через глобальну мережу Інтернет.

3.2 Функціональне тестування прототипу МaaS-платформи «розумного міста»

Для верифікації коректності функціонування інтерфейсу REST API, серверних сервісів («back-end») та їхньої інтеграції з блокчейн-мережею [1] проведено комплексне функціональне тестування з використанням специфікації «OpenAPI 3.0.1» – раніше відомої як Swagger. Цей систематичний підхід надав можливість підтвердити, що кожен окремий компонент функціонує відповідно до визначених технічних вимог, а їхня міжкомпонентна взаємодія є узгодженою та захищеною від збоїв.

Специфікація «OpenAPI» визначає стандартні регламенти для опису архітектурного стилю «REST API» у машинозчитуваному форматі, що забезпечує автоматичну генерацію інтерактивної документації та клієнтських бібліотек. Інтеграція «OpenAPI» із серверною частиною на базі Java/Spring реалізована за допомогою бібліотеки «Springdoc OpenAPI». Така архітектурна конфігурація дає можливість автоматично формувати вичерпну документацію програмних інтерфейсів, доступну через веббраузер за допомогою інструменту «SwaggerUI», що суттєво спрощує процеси тестування та валідації серверних служб і транзакцій у блокчейн-мережі. Верифікацію проведено для всіх критично важливих кінцевих точок, зокрема:

- Кінцеві точки автентифікації – перевірено механізми реєстрації користувачів, авторизації та сесійного доступу на основі маркерів.

- Кінцеві точки транзакцій – протестовано процедури створення, оновлення та видалення записів для забезпечення їхньої коректної обробки та безпомилкової фіксації в розподіленому реєстрі.

- Кінцеві точки отримання даних – підтверджено працездатність функцій із вибірки базової користувацької інформації, як-от історія поїздок та актуальний баланс віртуальних гаманців.

Кожен тестовий сценарій передбачав надсилання запитів, що містили як валідні, так і завідомо некоректні дані, з метою оцінювання логіки обробки відповідей API, інформативності повідомлень про помилки та процесів валідації вхідних параметрів [1]. Функціональне тестування продемонструвало позитивні результати за такими критеріями:

- Коректність функціонування – усі кінцеві точки відпрацювали у штатному режимі, а серверні сервіси безпомилково обробили надіслані запити.

- Взаємодія з блокчейн-мережею – транзакції були успішно внесені до розподіленого реєстру із повним збереженням цілісності даних.

- Обробка виняткових ситуацій – у разі передачі некоректних вхідних параметрів інтерфейс API повертав релевантні коди помилок та відповідні статуси HTTP.

– Продуктивність – часові показники відгуку МaaS-системи «розумного міста» виявилися задовільними, що свідчить про ефективність серверної обробки та мінімальні обчислювальні затримки під час виконання операцій у блокчейн-мережі.

На рисунку 3.1 зображено знімок екрана інтерфейсу OpenAPI, який використовувався під час проведення випробувань [1]. Продемонстровано доступні кінцеві точки API, згруповані за відповідними контролерами, а також розгорнутий контролер керування транспортними засобами із переліком його функціональних методів.

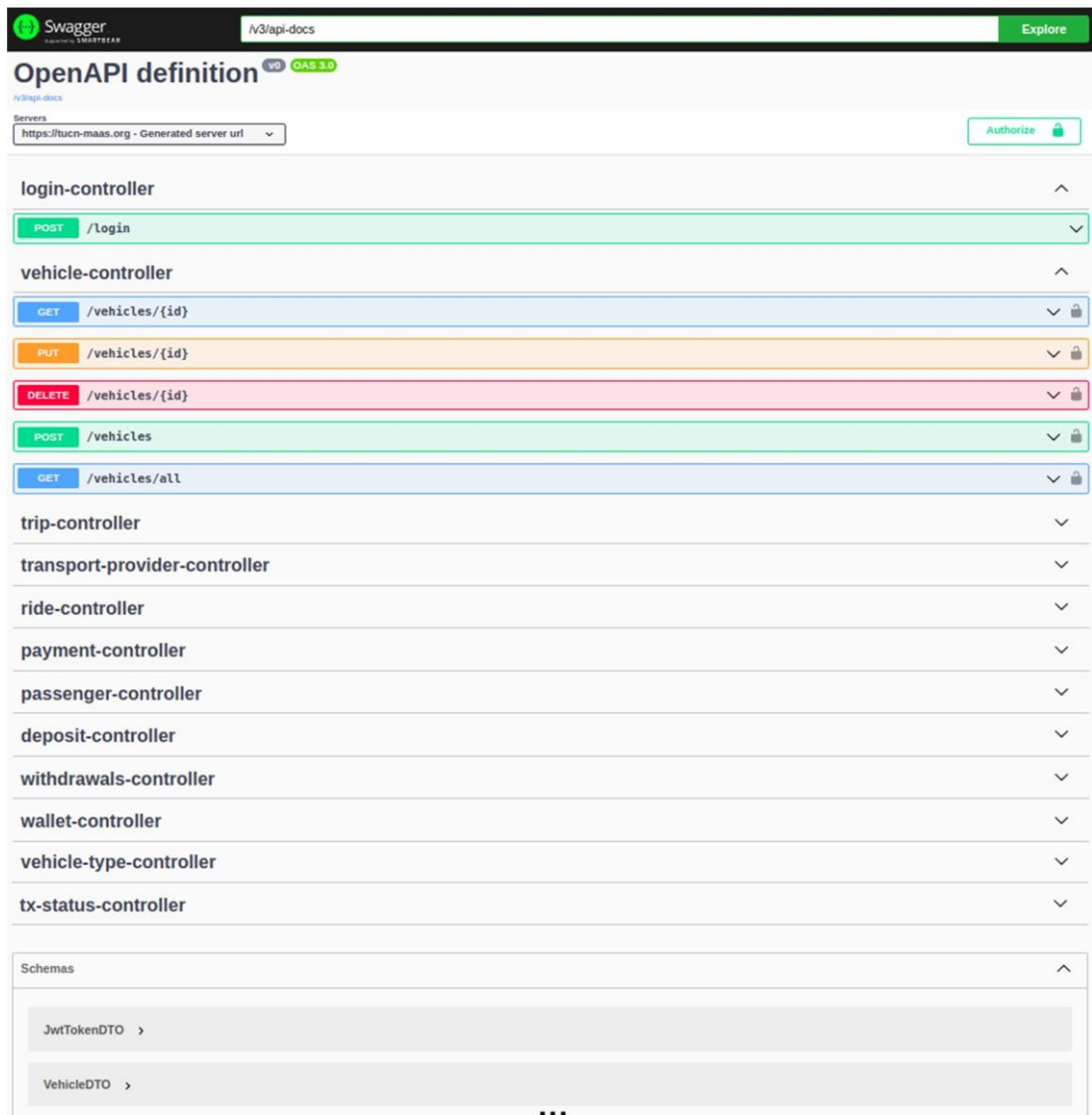


Рисунок 3.1 – Інтерфейс OpenAPI, що відображає контролери та кінцеві точки [1]

Для верифікації та безперервного моніторингу транзакцій у межах платформи було задіяно аналітичний інструмент Hyperledger Explorer. Цей засіб надає зручну графічну панель керування (див. рисунок 3.2) для візуалізації деталей транзакцій, структури блоків та загального поточного стану блокчейн-мережі.

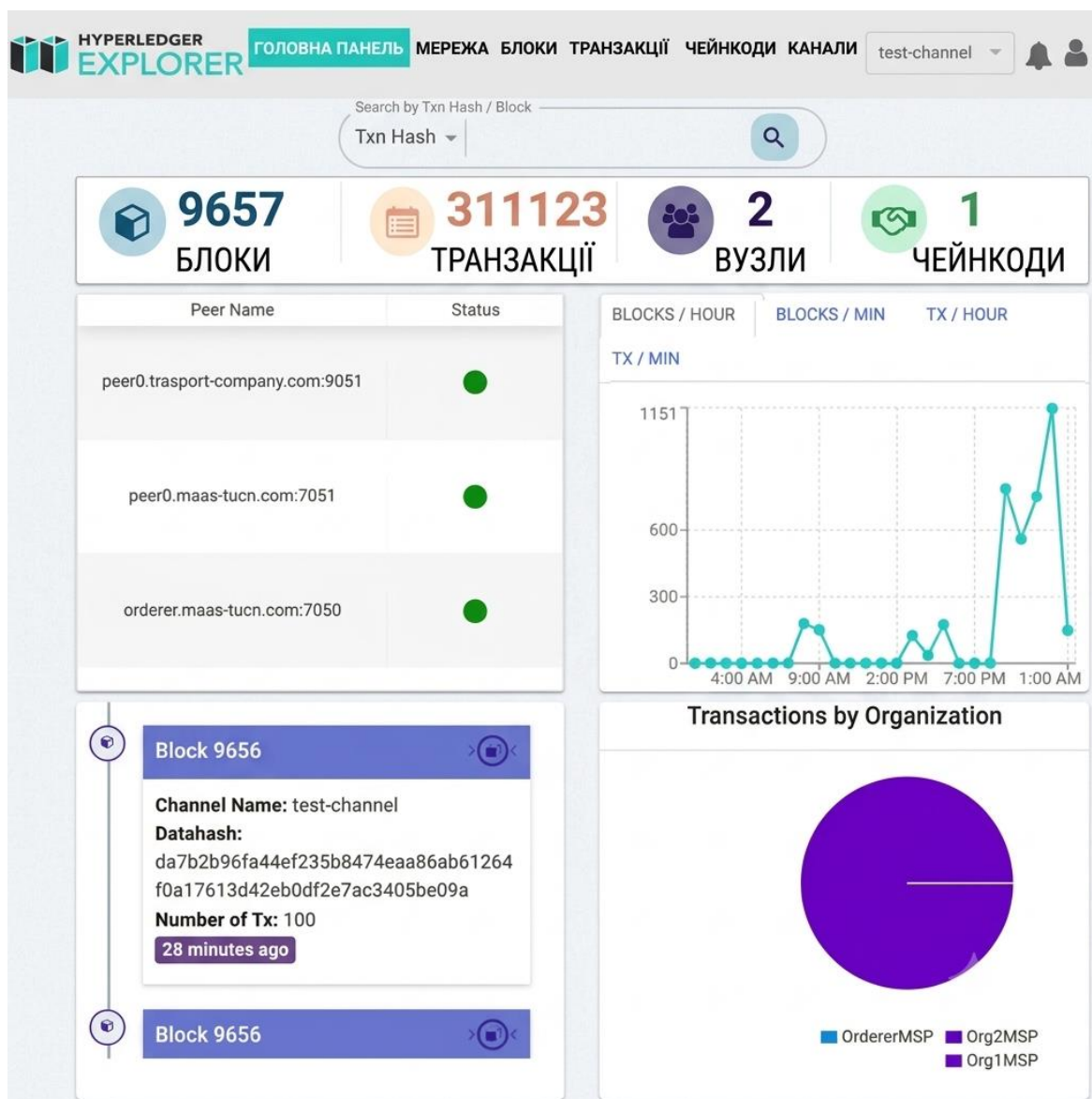


Рисунок 3.2 – Панель інструментів «Hyperledger Explorer» [1]

Завдяки використанню моніторингового інструменту було підтверджено, що всі операції, ініційовані серверними сервісами, стабільно фіксуються в блокчейні, забезпечуючи високу прозорість та незмінність інформації.

Функціональні можливості оновлення даних у режимі реального часу, аудит логів транзакцій та аналіз структури блоків, дали змогу остаточно валідувати коректність інтеграції розподіленого реєстру в загальну архітектуру MaaS-системи «розумного міста» [1].

Окрім цього, графічний інтерфейс відображає метадані транзакцій, часові мітки, ендорсменти – засвідчувальні підписи вузлів, та набори зчитування/запису. Часові мітки та дані ендорсменту верифікують точний час виконання операції та ідентифікують конкретні вузли, які провели її валідацію, що безпосередньо підтверджує працездатність механізму децентралізованого консенсусу. Набори зчитування/запису деталізують, які саме активи або записи були модифіковані в реєстрі, формуючи прозорий аудиторський слід еволюції даних у часі. «Hyperledger Explorer» візуалізує інформацію про логічні канали мережі, що є особливо цінним для систем із регламентованим доступом. Канали ізолюють сегменти мережі таким чином, щоб певні транзакційні дані залишалися доступними виключно авторизованим організаціям-учасникам, задовольняючи суворі вимоги до приватності MaaS-платформи «розумного міста» [1]. Шляхом моніторингу транзакцій всередині конкретних каналів стейкхолдери мають можливість переконатися в правильності партиціювання – розмежування, даних та підтвердити належне функціонування закладених механізмів конфіденційності в інфраструктурі «розумного міста».

Узагальнюючи, зазначені компоненти графічної панелі моніторингу підтвердили високу надійність архітектури MaaS-системи «розумного міста» шляхом візуальної верифікації того, що кожна транзакція була безпечно внесена до реєстру, належним чином засвідчена відповідними вузлами мережі та збережена на засадах незмінності даних [1]. Такий рівень прозорості надає провайдерам транспортних послуг та кінцевим користувачам можливість ефективно відстежувати та здійснювати аудит повного життєвого циклу даних у розподіленому реєстрі, що суттєво підвищує довіру до показників продуктивності, стабільності та загальної безпеки функціонування MaaS-платформи «розумного міста».

3.3 Дослідження продуктивності та навантажувальне тестування елементів МaaS-платформи «розумного міста»

З метою визначення характеру впливу інтенсивності одночасних запитів та обсягу записів у розподіленому реєстрі блокчейн-мережі на часові показники відгуку платформи проведено серію комплексних навантажувальних випробувань [1]. Необхідно зауважити, що під час виконання цих оцінювань уся інфраструктура МaaS-системи «розумного міста», зокрема, серверні сервіси («back-end») та компоненти блокчейн-мережі, функціонувала на базі одного обчислювального вузла. Як наслідок, отримані метрики продуктивності мають відносний характер і можуть зазнавати змін у разі розгортання архітектури в промисловому середовищі. Головна мета цих випробувань полягала у фіксації флуктуацій часу відгуку залежно від щільності конкурентних запитів та масштабу накопичених у блокчейні даних.

Для практичної реалізації сценаріїв навантажувального тестування було застосовано програмний інструментарій «Apache JMeter 5.6.3». На рисунку 3.3 продемонстровано базову компонентну структуру сформованих тест-планів.

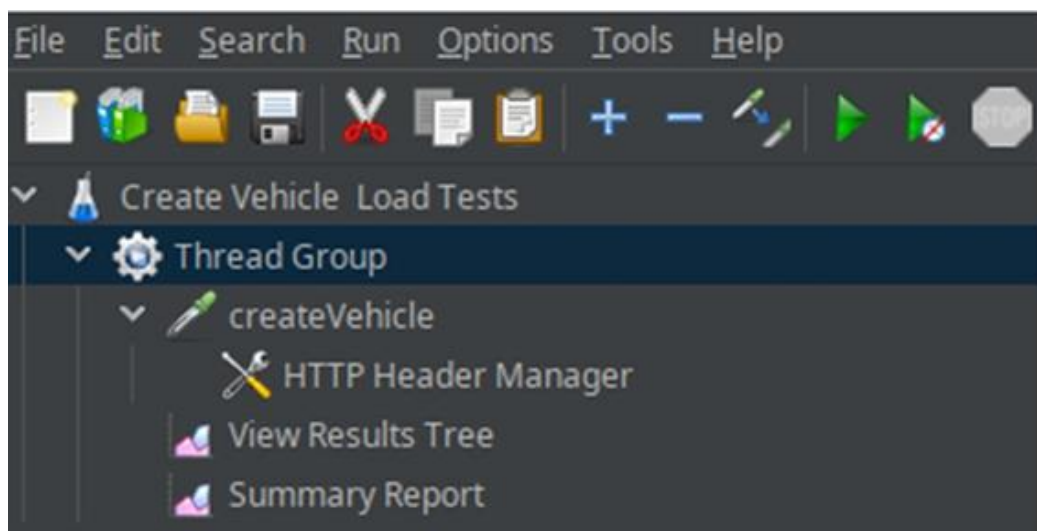


Рисунок 3.3 – Компоненти плану тестування Apache JMeter [1]

Типовий план тестування містить у собі групу потоків (див. рисунок 3.4) для імітації паралельних користувачів, модуль генерації мережевих HTTP-запитів, а також два компоненти візуалізації та аналізу отриманих результатів для подальшого аудиту продуктивності платформи в інфраструктурі «розумного міста».

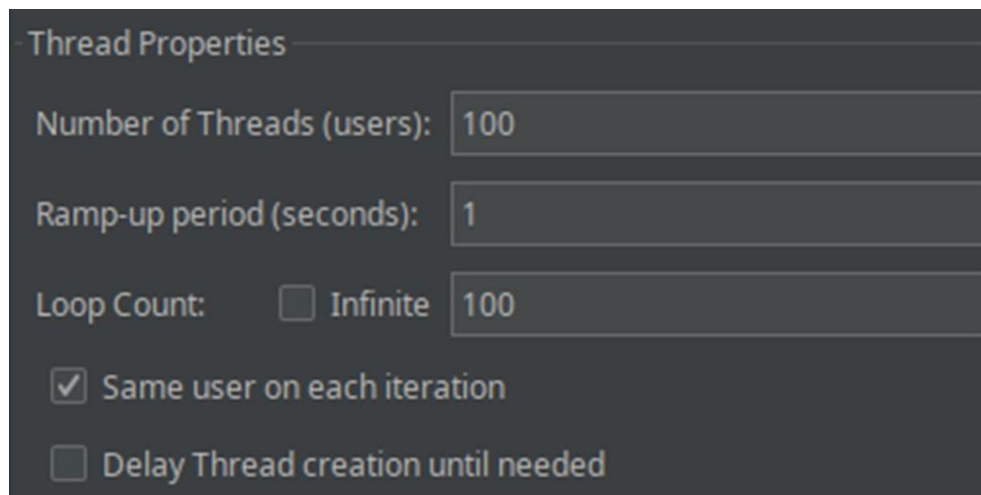


Рисунок 3.4 – Конфігурація групи потоків Apache JMeter [1]

Сформовані сценарії тестування охоплюють оцінювання продуктивності двох базових операцій: обчислення сукупних місячних витрат користувача та отримання інформації про стан віртуального гаманця за його унікальним ідентифікатором (ID). Доступ до зазначеного функціоналу кінцеві користувачі отримують через відповідні клієнтські застосунки.

У ролі бази даних поточного стану консорціуму «Hyperledger Fabric» застосовано СКБД «CouchDB» [1]. Для інтенсифікації процесу пошуку за критеріями другої операції було спроектовано та впроваджено складений індекс за полями пошуку: «[sourceWalletId, date]». Інтеграція створеного індексу «CouchDB» до структури смартконтракту (чейнкоду) вимагає розміщення конфігураційного JSON-файлу в головній директорії вихідного коду за відносним шляхом «META-INF/statedb/couchdb/indexes». Специфікація вмісту індексу подана в лістингу 3.1.

Лістинг 3.1 – Вмісту індексного файлу смартконтракту

```

{
  "index": {
    "fields": [
      "sourceWalletId",
      "date"
    ]
  },
  "ddoc": "indexSourceWalletIdDate",
  "name": "indexSWalletDate",
  "type": "json"
}

```

Після послідовного внесення до розподіленого реєстру блокчейн-мережі масивів імітаційних даних обсягом 10 000, 100 000 та 1 000 000 записів відповідно, проведені випробування продемонстрували, що за умови попереднього індексування інформації за визначеними критеріями пошуку часові показники відгуку МaaS-системи «розумного міста» [1] є інваріантними – не залежать, від загального обсягу збережених даних. Середні значення часу відгуку блокчейн-платформи для кожної фіксованої кількості паралельних запитів наведено в таблиці 3.3.

Таблиця 3.3 – Результати навантажувальних випробувань МaaS-платформи «розумного міста» [1]

Кількість запитів	Доступ до гаманця, час ідентифікатора, мс	Загальний час, мс
1	5	12
10	13	29
50	65	103
100	131	160
200	370	410
300	497	640
400	716	843

При досягненні рівня інтенсивності близько 400 паралельних запитів середній коефіцієнт помилок відгуку МaaS-системи «розумного міста» [1] – перевищення ліміту часу очікування, становив орієнтовно десять відсотків для обох досліджуваних операцій. Визначена гранична межа кількості запитів може слугувати референтним орієнтиром для подальшого масштабування та динамічного коригування кількості фонових обробників у черзі повідомлень брокера.

У таблиці 3.4 подано усереднені показники часу відгуку МaaS-системи «розумного міста» [1], розраховані для масиву зі 100 запитів в один потік, під час виконання базових операцій, підтримуваних кожною окремою сутністю моделі даних, зокрема, компоненти: «Vehicle» – транспортний засіб), «Passenger» – пасажир, «Ride» – поїздка, та інші інформаційні структури, інтегровані в інформаційно-технологічну систему «розумного міста».

Таблиця 3.4 – Часові показники основних операцій МaaS-платформи «розумного міста» [1]

Попередньо існуюча кількість записів	Доступ до запису за іменем – без індексу, мс	Отримання запису за ідентифікатором, мс	Оновлення запису, мс	Видалення запису, мс
1000	45	4	2123	2013
5000	152	4	2210	2101
10,000	312	5	2175	2175
50,000	1453	4	2187	2141
100,000	2645	4	2256	2123
1,000,000	24,987	6	2187	2217

Як продемонстровано в таблиці, продуктивність базових операцій створення, зчитування, оновлення та видалення даних («CRUD») є інваріантною відносно кількості наявних записів у розподіленому реєстрі.

Висока ефективність вибірки даних за унікальним ідентифікатором (ID) забезпечується завдяки механізму неявного індексування активів. Водночас виконання операцій внесення змін, модифікації та вилучення стабільно триває приблизно дві секунди для кожної процедури. Це зумовлено тим, що процеси зміни стану блокчейн-мережі потребують виконання регламентів міжвузлового інформаційного обміну та процедур взаємної валідації – консенсусу.

Така фіксована тривалість обумовлена переважно дією системного параметра «BatchTimeout», значення якого за замовчуванням становить дві секунди. Параметр «BatchTimeout» визначає інтервал часу, впродовж якого блокчейн-мережа очікує перед формуванням нового блока: усі нові верифіковані транзакції групуються в пакети та додаються до структури блока кожні дві секунди [1]. Натомість вибірка активу за альтернативними атрибутами, наприклад, за назвою, супроводжується значними часовими затримками, що є прямо пропорційними загальному масштабу досліджуваного масиву даних. Окреслена проблема нівелюється шляхом додаткового індексування активів за назвою, що суттєво підвищує загальну ефективність пошукових запитів в інфраструктурі «розумного міста». Часові показники, зафіксовані в табл. 3.3 та 3.4, були виміряні локально на рівні серверної частини («back-end») платформи і не враховують затримок, що виникають під час передачі даних через фізичні канали комп'ютерної мережі.

3.4 Налаштування мережі блокчейн для прототипу МaaS-платформи «розумного міста»

Оптимізація продуктивності мереж розподіленого реєстру є критично важливим завданням під час проектування програмних платформ корпоративного рівня. Параметр «MaxMessageCount» – розмір пакета, визначає граничну кількість транзакцій, що можуть бути об'єднані в межах одного блока, тоді як «BatchTimeout» регламентує максимальний часовий інтервал очікування перед формуванням нового блока у випадку, якщо встановлений

ліміт кількості транзакцій не був досягнутий. Тонке налаштування зазначених параметрів безпосередньо впливає на пропускну здатність обробки транзакцій та величину часових затримок у MaaS-системі «розумного міста» [1].

Оцінювання продуктивності здійснювалося за допомогою інструментарію «Apache JMeter» із використанням фіксованої конфігурації плану тестування: сто паралельних потоків, тривалість лінійного наростання навантаження («ramp-up time») одна секунда та сто ітерацій для кожного окремого потоку. В ролі базової тестової операції було обрано процедуру створення запису про транспортний засіб («create vehicle»), проте для отримання репрезентативних метрик могла бути задіяна будь-яка інша транзакція моделі даних. Кінцева мета експерименту полягала у визначенні оптимальних архітектурних конфігурацій, що забезпечують раціональний баланс між швидкістю обробки інформації та рівнем використання апаратних ресурсів.

Випробування [1] проводилися у дистанційному режимі, що дало можливість природним чином врахувати затримки фізичних каналів передачі даних комп'ютерної мережі у підсумкових метриках продуктивності. Для підтвердження валідності, відтворюваності та узгодженості отриманих результатів серії експериментів повторювалися у різний час впродовж доби. Показники ефективності залишалися стабільно подібними між усіма повтореннями, що підтверджує високу надійність, релевантність та об'єктивність сформованих висновків.

У таблиці 3.5 подано емпіричні результати оцінювання продуктивності, отримані за допомогою навантажувальних випробувань мережі «Hyperledger Fabric». Під час проведення цих тестів варіювалися значення конфігураційного параметра «MaxMessageCount», тоді як показник «BatchTimeout» залишався фіксованим і відповідав базовому значенню за замовчуванням дві секунди, що є важливим кроком для розробки стійких рішень «розумного міста» [1].

Таблиця 3.5 – Метрики продуктивності для варіювання «MaxMessageCount» [1]

Максимальна кількість повідомлень	Середня затримка, мс	Максимальна затримка, мс	Стандартне відхилення, мс
10	4671	6724	521
50	3439	6455	467
100	2881	5937	454
200	3496	5945	276
500	4023	6742	586

Подані дані демонструють, що збільшення розміру пакета з десяти до ста транзакцій в одному блоці призводить до істотного зниження середнього рівня затримки – з 4.6 до 2.9 с, що свідчить про інтенсифікацію продуктивності Маас-системи «розумного міста». Проте при перевищенні порога у сто транзакцій на пакет показники ефективності починають погіршуватися. Зокрема, для розмірів пакета у двісті та п'ятсот одиниць середня затримка зростає до 3.5 та 4 с відповідно. Це дає змогу зробити висновок, що для поточної архітектурної конфігурації розподіленого реєстру та фіксованої інтенсивності надходження у сто записів за секунду оптимальна величина розміру пакета становить близько сто транзакцій. При перевищенні цього екстремуму переваги пакетної обробки нівелюються збільшенням сумарного часу очікування на формування блоків.

Ураховуючи стабільну інтенсивність генерації транзакцій на рівні сто запитів за секунду, вибір розмірів пакетів величиною сімдесят п'ять, сто та сто двадцять п'ять повідомлень є математично обґрунтованим підходом для подальшого тонкого налаштування системного параметра «BatchTimeout» [1]. Зазначені конфігурації відповідають критичним сценаріям, коли ліміт місткості пакета є відповідно дещо меншим, рівним або більшим за динамічну швидкість надходження транзакцій. Такий підхід надає можливість провести всебічне

оцінювання того, як різні обсяги пакетів відносно вхідного потоку даних впливають на загальні метрики продуктивності мережі. У таблиці 3.6 узагальнено емпіричні метрики ефективності, зафіксовані для різноманітних комбінацій значень параметрів «MaxMessageCount» та «BatchTimeout» в інфраструктурі MaaS-системи «розумного міста» [1].

Таблиця 3.6 – Метрики продуктивності для різних значень «MaxMessageCount» та «BatchTimeout» [1]

Максимальна кількість повідомлень	Тайм-аут пакета, с	Середня затримка, мс	Максимальна затримка, мс
75	0.5	3996	7032
75	1	3733	7167
75	2	3630	7113
75	3	3765	6424
100	0.5	3122	5421
100	1	2949	5028
100	2	2881	5937
100	3	3323	4542
125	0.5	3104	5954
125	1	3208	5409
125	2	3483	5477
125	3	4451	5323

Результати проведених випробувань свідчать про те, що узгоджене налаштування параметрів «MaxMessageCount» та «BatchTimeout» чинить визначальний вплив на величину часових затримок у блокчейн-мережі. Мінімальний середній показник затримки, який становить 2.9 с, зафіксовано за умови встановлення параметра «MaxMessageCount» на рівні сто одиниць та

значення «BatchTimeout» тривалістю дві секунди. Таке співвідношення відображає оптимальний архітектурний баланс для стабільного вхідного навантаження з інтенсивністю 100 запитів за секунду.

При зменшенні параметра «MaxMessageCount» до семидесяти п'яти одиниць збільшення тривалості «BatchTimeout» від половини до двох секунд забезпечує додаткове зниження латентності «розумного міста», що вказує на ефективність пролонгованого часу акумуляції транзакцій для пакетів меншого обсягу. Натомість підвищення граничного значення «MaxMessageCount» до ста двадцяти п'яти одиниць та встановлення ліміту «BatchTimeout» понад дві секунди призводить до зростання часових затримок [1]. Це демонструє, що надлишковий обсяг пакетів у поєднанні з тривалим очікуванням на формування блоків погіршує загальну продуктивність.

Отримані емпіричні дані підтверджують критичну необхідність синхронізації параметра «MaxMessageCount» із реальними показниками інтенсивності генерації транзакцій, а також ретельного розрахунку величини «BatchTimeout» для максимізації ефективності функціонування MaaS-платформи в інфраструктурі «розумного міста».

3.5 Порівняльний аналіз з іншими інноваційними технологіями

У порівнянні з іншими перспективними технологіями, спрямованими на вдосконалення процесів міської мобільності, зокрема, системами на основі ШІ [26], сенсорними мережами IoT [27] або традиційними хмарними платформами [28], технологія розподіленого реєстру – блокчейн, вирізняється насамперед своєю здатністю формувати децентралізоване середовище довіри між багатьма незалежними стейкхолдерами. Інтелектуальні системи на базі ШІ демонструють високу ефективність під час розв'язання задач прогностичної аналітики, оптимізації маршрутів і динамічного планування графіків. Проте вони переважно функціонують із використанням централізованих сховищ даних, що актуалізує ризики порушення конфіденційності користувачів [29].

Сенсорні IoT-мережі [30] забезпечують збір метрик про стан транспортних засобів та інфраструктури в режимі реального часу, що є критично важливим для просторового планування; водночас процеси аналізу та безпечної дистрибуції масивних потоків інформації між різними суб'єктами вимагають впровадження суворих стандартів інтероперабельності [31]. Свою чергу, хмароорієнтовані архітектури [32] мають розвинений інструментарій для гнучкого масштабування, проте вони так само передбачають концентрацію даних, підвищуючи вразливість «розумного міста» до виникнення єдиної точки відмови або технологічної залежності від конкретного постачальника послуг [33].

Натомість МaaS-платформа на основі блокчейн-технологій нівелює зазначені недоліки завдяки використанню розподілених, захищених від несанкціонованої модифікації реєстрів. Це суттєво знижує ризики шахрайства та забезпечує прозору взаємодію між транспортними операторами та пасажиром. Водночас обмеження щодо забезпечення високої пропускну здатності або обсягів збереження великих за обсягом наборів та колекцій даних, які притаманні суто централізованим рішенням, зумовлюють необхідність проектування гібридних архітектур із винесенням частини операцій за межі основного реєстру («off-chain»). У таблиці 3.7 подано узагальнене порівняння технології блокчейн з іншими інноваційними рішеннями, що зазвичай впроваджуються у сферах міської мобільності.

Таблиця 3.7 – Плюси та мінуси нових технологій для міської мобільності [1]

Технології	Плюси	Мінуси
1	2	3
Системи на основі ІІТ	Прогнозна аналітика, оптимізація маршрутів, динамічне планування	Централізоване управління даними, потенційні проблеми конфіденційності, прозорість моделі «чорної скриньки»

1	2	3
Сенсорні мережі Інтернету речей	Моніторинг дорожнього руху та стану транспортних засобів у режимі реального часу	Складність обробки та безпеки великомасштабних даних, проблеми сумісності
Хмарні платформи	Масштабована інфраструктура, розширені інструментальні ланцюжки розробки	Ризики централізації даних, потенційна прив'язаність до постачальника, вразливості під час простою сервісу
Блокчейн (MaaS)	Децентралізована довіра, захищені від несанкціонованого доступу реєстри, прозорий потік даних	Обмежена пропускна здатність, складніше налаштування, часто потрібне позаланцюжкове сховище для великих за обсягом наборів та колекцій даних

Попри вагомій перевазі блокчейну в контексті гарантування прозорості та довіри, він не позиціонується як повна альтернатива згаданим технологіям, а виступає стрижневим елементом ширшої інформаційно-технологічної системи:

- III аналіз забезпечує функціонування інтелектуальних прогнозних моделей, тоді як блокчейн гарантує цілісність даних між різними учасниками ринку;

- IoT-сенсори [34] постачають оперативні навігаційні та телеметричні метрики, а розподілений реєстр забезпечує захист і валідацію цієї інформації у спільному сховищі;

- хмарні обчислювальні архітектури виконують ресурсомісткі розрахунки, тоді як блокчейн фіксує транзакції, фінансові платежі та квитки у захищений від підробки спосіб.

Завдяки синергії та інтеграції сильних сторін декількох технологічних підходів, сучасні системи міської мобільності здатні досягти раціонального балансу між продуктивністю, інформаційною безпекою та архітектурною стійкістю як для пасажирських, так і для вантажних перевезень в інфраструктурі «розумного міста» [1].

3.6 Перспективи подальших досліджень у сфері інтеграції MaaS систем

У межах дослідження [1] було проведено комплексний аналіз інтеграції технології розподіленого реєстру – блокчейн, в інтелектуальні мобільні «розумного міста» з особливим акцентом на концепцію мобільності як послуги (MaaS). Оцінено ефективність застосування блокчейну для підвищення продуктивності, безпеки та екологічної стійкості міських транспортних мереж. Отримані результати підтверджують, що блокчейн як децентралізоване, прозоре та захищене архітектурне рішення здатний ефективно нівелювати проблеми взаємної довіри та складнощі інтеграції між різнорідними транспортними технологіями й незалежними стейкхолдерами.

Архітектура на базі консорціуму «Hyperledger Fabric» продемонструвала високу ефективність під час обробки значних масивів вхідних запитів, що позиціонує її як перспективний базис для розбудови мережевих транспортних систем. Дослідження [1] підкреслює потенціал блокчейн-технологій у контексті підвищення рівня безпеки та прозорості шляхом оптимізації процесів дистрибуції квитків та координації розрахунково-платіжних процедур. Додатково деталізовано ключові програмно-алгоритмічні засоби MaaS-платформи, де особливу увагу приділено типізації даних та загальній системній архітектурі.

Конфігурація експериментального стенда на базі одного обчислювального вузла, використана для первинних пілотних випробувань, виявилася економічно доцільною, проте виявила обмеження щодо

масштабованості та стійкості MaaS-системи «розумного міста» для «розумного міста» під навантаженням: погіршення показників продуктивності зафіксовано при досягненні порога у чотириста одночасних запитів [1]. Окреслені обмеження зумовили необхідність переходу до хмароорієнтованої розподіленої архітектури для забезпечення штатного функціонування MaaS-системи «розумного міста» в реальних умовах.

Під час проведення другої серії випробувань у хмарному середовищі встановлено, що точне налаштування параметрів «Hyperledger Fabric» із визначенням показника «MaxMessageCount» на рівні сто одиниць та параметра «BatchTimeout» тривалістю дві секунди є оптимальним для мінімізації латентності та максимізації пропускної здатності при інтенсивності сто транзакцій за секунду. Конфігурація [1] гарантує високу ефективність обробки інформації при збереженні низького рівня затримок, що є критично важливим для масштабованих MaaS-платформ. Перехід до такої архітектури дає можливість забезпечити динамічне масштабування MaaS-системи «розумного міста» в умовах зростання кількості користувачів, а впровадження механізмів балансування навантаження та кешування даних підвищує швидкість відгуку в періоди пікових навантажень. Подальше застосування стратегій очищення застарілих даних та секціонування реєстру дасть змогу гарантувати довгострокову ефективність функціонування розглянутого рішення в межах масштабного розгортання в інфраструктурі «розумного міста».

Майбутні напрями досліджень для верифікації масштабованості MaaS-систем «розумного міста» у розподілених хмарних середовищах мають пріоритетно передбачати реалізацію пілотних проектів та проведення натурних експериментів [1]. Крім того, актуальним завданням є вивчення можливостей застосування інформаційних та комунікаційних технологій для інтеграції вантажної логістики в єдину MaaS-систему «розумного міста», що надасть можливість гарантувати безпечну та прозору координацію між пасажирськими і вантажними перевезеннями.

3.7 Висновок до третього розділу

В третьому розділі кваліфікаційної роботи описано процес розгортання та апаратно-програмне забезпечення експериментального середовища. Розглянуто функціональне тестування прототипу МaaS-платформи «розумного міста». Проведено дослідження продуктивності та навантажувальне тестування елементів МaaS-платформи «розумного міста». Розглянуто процес налаштування мережі блокчейн для прототипу МaaS-платформи «розумного міста». Подано порівняльний аналіз з іншими інноваційними технологіями. Описано перспективи подальших досліджень у сфері інтеграції МaaS систем.

РОЗДІЛ 4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ, ОСНОВИ ОХОРОНИ ПРАЦІ

4.1 Державна система моніторингу довкілля як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн

З огляду на специфіку архітектурної організації міського простору, концептуальні підходи до створення сучасних MaaS-платформ вимагають врахування екологічних факторів та параметрів стану навколишнього середовища. Відтак, державну систему моніторингу довкілля доцільно розглядати як фундаментальний складник національної інформаційної інфраструктури, інтегрований у загальну екосистему мультимодальних перевезень. Раціональна організація мобільності кінцевих користувачів та сталий розвиток транспортних мереж у «розумних містах» є неможливими без належної консолідації та аналізу геоecологічних даних. Створення інтегрованої інформаційної платформи [35] забезпечує постійний доступ до оперативних екологічних показників, що дозволяє оптимізувати маршрути транспортних засобів, мінімізувати шкідливі викиди в атмосферу та гнучко керувати ресурсами міського транзиту.

Зазначена державна інфраструктура виконує функції безперервного контролю за екологічними параметрами урбанізованих територій, забезпечуючи органи місцевого самоврядування, операторів мобільності та громадськість валідованими даними про стан біосфери. У межах функціонування системи здійснюється акумуляція, обробка та архівація відомостей щодо ступеня забруднення повітряного басейну, стану водних об'єктів і ґрунтового покриву. Це дає змогу сформувати багатовимірну модель екологічної ситуації в межах транспортних коридорів. Функціональні вимоги до моніторингового комплексу передбачають високу точність вимірювань, мінімальний час затримки при передачі інформації, а також відповідність

міжнародним протоколам обміну даними для забезпечення глобальної інтероперабельності.

Технічна архітектура моніторингу на загальнодержавному рівні базується на використанні розподілених комплексів вимірювальних давачів, автоматизованих станцій спостереження, засобів дистанційного зондування Землі та високошвидкісних телекомунікаційних мереж. Сформовані інформаційні потоки спрямовуються до централізованих сховищ даних, де підлягають аналітичній обробці. Важливим завданням є розробка програмних інтерфейсів для швидкого отримання зведених звітів, що критично важливо для оперативного реагування на критичні зміни екологічного стану. Окрім того, підсистема прогнозування дозволяє моделювати сценарії трансформації міського середовища під впливом кліматичних змін, що сприяє плануванню екологічно безпечних пасажирських перевезень.

Синхронізація національних засобів контролю довкілля із закордонними аналогами та глобальними базами даних є обов'язковою передумовою для уніфікації інформаційного обміну [36]. Це гарантує стандартизацію та верифікацію процесів збору й поширення метрик, підвищуючи прозорість моніторингу. Сучасний рівень розвитку технологій міжсистемної взаємодії дає змогу різним державам створювати координовані платформи, оптимізуючи процеси спільного використання просторових та екологічних даних.

Можливість аналізу параметрів у режимі реального часу має визначальне значення для нейтралізації техногенних загроз. Завдяки взаємодії вітчизняних систем із міжнародними центрами з'являється можливість імпортувати актуальні відомості про кліматичні коливання та рівні забруднення в різних регіонах. Такі дані є корисними як для оцінювання поточного стану міського середовища, так і для побудови прогностичних моделей адаптації транспортної інфраструктури до нових умов.

Завдяки консолідації зусиль профільні інституції та міжнародні організації отримують інструменти для колективної протидії екологічним викликам, зокрема деградації природних комплексів та антропогенному

навантаженню на атмосферу. Спільне оперування інформаційними масивами дає можливість точніше оцінити масштаби негативного впливу транспорту на екосистеми міст, що допомагає визначити пріоритетні вектори розвитку екологічно чистої мобільності та впроваджувати стратегії сталого розвитку.

Міжнародна кооперація в аналізованій галузі орієнтована також на підвищення ефективності реалізації програм охорони навколишнього середовища. Створення єдиних інформаційних платформ на основі глобальних баз даних дає змогу здійснювати моніторинг виконання підписаних конвенцій та угод. Такий підхід підвищує рівень взаємної довіри між партнерами та уможливорює точне вимірювання динаміки досягнення цілей сталого розвитку.

Забезпечення інтероперабельності вітчизняних систем моніторингу з провідними світовими комплексами, такими як відповідні інфраструктурні платформи Організації Об'єднаних Націй чи європейські екологічні мережі, виступає базовим елементом безпеки життєдіяльності. Програмні рішення, що підтримують відкриті стандарти, спрощують інтеграцію локальних сервісів у глобальні ініціативи, формуючи уніфікований інформаційний простір. Такі інформаційно-технологічні комплекси мають адаптуватися до територіальних та кліматичних особливостей регіону для побудови точних локальних прогнозів. Окрему увагу слід приділити впровадженню інноваційних методів інтелектуального аналізу великих масивів інформації. Використання хмарних обчислень та алгоритмів штучного інтелекту дозволяє автоматизувати процеси прогнозування екологічних коливань на основі даних, що надходять у режимі реального часу. Програмні модулі спроможні не лише ідентифікувати потенційні загрози, а й формувати рекомендації щодо оптимізації руху міського транспорту задля мінімізації загального екологічного навантаження.

Таким чином, підсистема моніторингу природного середовища є критично важливим вузлом національної інформаційної інфраструктури. Вона забезпечує швидке реагування на динамічні зміни параметрів біосфери та стимулює міжнародну взаємодію. Інтеграція зазначених архітектурних рішень у концепцію «розумних міст» дає можливість здійснювати екологічний

контроль на локальному рівні, оперативно коригувати алгоритми функціонування транспортних мереж та забезпечувати баланс між потребами мобільності й збереженням довкілля.

У контексті функціонування МaaS-платформи «розумного міста» державна система моніторингу довкілля постає як постачальник первинних даних для модулів планування та диспетчеризації мультимодальних поїздок. Ефективність розподілу транспортних потоків та вибору оптимальних засобів пересування безпосередньо залежить від зовнішніх чинників, зокрема від метеорологічних умов, ступеня загазованості повітря та інтенсивності опадів. Оперативні відомості, отримані від екологічної інфраструктури, дозволяють інтелектуальним алгоритмам МaaS-платформи коригувати параметри маршрутизації, прогнозувати завантаженість транспортних артерій та пропонувати користувачам найбільш екологічні варіанти переміщення, наприклад, пріоритезацію електротранспорту чи засобів мікромобільності під час сприятливої погоди.

Крім того, міжнародна сумісність аналізованої системи відкриває можливості для залучення ширших аналітичних моделей та транскордонних баз даних, що суттєво підвищує точність алгоритмів прогнозування в межах МaaS-платформи. Дослідження глобальних кліматичних закономірностей та врахування досвіду іноземних операторів дає змогу адаптувати стратегії розвитку міської мобільності до довгострокових змін екологічних стандартів, забезпечуючи масштабованість та гнучкість усієї міської інфраструктури «розумного міста».

4.2 Вимоги ергономіки до організації робочого місця оператора ПК

Аналіз умов професійної діяльності під час створення інфраструктурних елементів МaaS-платформи унаочнив, що провідними ризиками для фахівця з розробки є тривалі статичні навантаження на опорно-руховий апарат і суттєве напруження зорового аналізатора. Основним інструментом для мінімізації

зазначених ризиків, а також для формування безпечного й продуктивного виробничого середовища є практичне впровадження принципів ергономіки. Цей науковий напрям орієнтований на адаптацію робочої зони, технічних засобів та алгоритмів діяльності до фізіологічних і психологічних можливостей організму людини. Вимоги ергономіки щодо просторової організації місця оператора персонального комп'ютера чітко регламентуються вітчизняними нормативними документами.

Раціональне компонування фізичних елементів робочої зони виступає базою для профілактики патологій опорно-рухової системи. Першочергова увага приділяється конструкції крісла, яке забезпечує компенсацію статичних напружень тіла. У процесі тривалого проектування модулів МaaS-платформи встановлено, що належна конфігурація сидіння є визначальним фактором тривалої працездатності. Проведення багатоодинних сесій написання програмного коду за відсутності належної підтримки поперекового відділу хребта та можливості динамічної зміни кута нахилу спинки призводить до швидкого втомлювання. Відтак, робоче крісло має бути підйомно-поворотним, із можливістю регулювання висоти, кутів нахилу сидіння і спинки, а також висоти підлокітників. Така конструкція гарантує підтримання фізіологічно оптимальної пози, що мінімізує напруження м'язів шийно-плечової області.

Окрему роль відіграє робочий стіл. Його робоча поверхня повинна мати достатні габарити для зручного розміщення обчислювальної техніки, периферійних пристроїв та документації, а простір для ніг має забезпечувати вільну зміну положення кінцівок. При розробці великих програмних систем стає очевидним, що геометричні параметри столу мають критичне значення для загального комфорту, оскільки недостатня глибина стільниці змушує розташовувати засіб відображення інформації занадто близько до очей фахівця.

Продумане розташування монітора є найбільш дієвим превентивним заходом проти зорового стомлення. Згідно з чинними санітарними нормативами, екран повинен перебувати на відстані від 50 до 70 сантиметрів від очей оператора, а його верхня межа має знаходитися на рівні зору або трохи

нижче. Під час безперервного аналізу розгалужених інтерфейсів, наприклад панелей адміністрування міських транспортних потоків, важливо орієнтувати монітор так, щоб повністю виключити появу відблисків від вікон або джерел штучного освітлення, які викликають додаткове напруження очей. Завершують організацію фізичного простору пристрої введення даних — клавіатура та миша. Для запобігання розвитку тунельних синдромів клавіатуру слід розміщувати на відстані від 10 до 30 сантиметрів від краю столу, що створює надійну опору для передпліч. Доцільним є використання ергономічних периферійних моделей, які забезпечують природне положення кистей рук.

Проте, навіть максимально раціоналізоване робоче місце не здатне повністю нівелювати негативні наслідки тривалої безперервної праці. Невід'ємною складовою ергономічного підходу є впровадження регламентованих режимів праці та відпочинку. Діючі санітарні норми рекомендують установлювати технологічні перерви тривалістю від 5 до 10 хвилин через кожні 45–60 хвилин безперервної діяльності [37]. Під час вирішення складних інженерних завдань, як-от оптимізація алгоритмів обробки транзакцій у розподіленому реєстрі MaaS-платформи, виникає тенденція до ігнорування відпочинку. Однак такий підхід є неефективним, оскільки зумовлює кумуляцію втоми та збільшує ймовірність виникнення помилок у коді. Регулярні короткі перерви, згідно з дослідженнями у сфері психології безпеки праці, дозволяють не лише розвантажити зоровий та м'язовий апарати, а й стимулюють відновлення когнітивних процесів, що підвищує загальну швидкість знаходження правильних рішень [38].

Таким чином, системне впровадження вимог ергономіки до просторової організації робочої зони та дотримання науково обґрунтованого балансу між працею й відпочинком дозволяють суттєво знизити вплив шкідливих виробничих чинників, зберегти здоров'я розробника та підтримувати високу продуктивність праці впродовж усього робочого дня.

4.3 Висновок до четвертого розділу

В четвертому розділі кваліфікаційної роботи описано державну систему моніторингу довкілля як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн. Розглянуто вимоги ергономіки до організації робочого місця оператора ПК.

ВИСНОВКИ

На основі комплексного аналізу теоретичних засад та оцінювання програмного інструментарію обґрунтовано, спроектовано та експериментально верифіковано архітектуру МaaS-платформи для «розумного міста» із застосуванням технології блокчейн, що дозволило забезпечити високу продуктивність, інтероперабельність та захищеність процесів координації мультимодальних транспортних послуг.

В першому розділі кваліфікаційної роботи освітнього рівня «Бакалавр»:

- Подано теоретико-методологічні засади проектування МaaS-платформ.
- Проаналізовано світовий досвід та інструментарій програмної реалізації МaaS-платформ.

- Висвітлено теоретико-технологічні основи застосування технології блокчейн та її класифікація.

- Проаналізовано технологічні виклики та обмеження інтеграції технології блокчейн у МaaS-платформи.

- Подано порівняльний аналіз програмних платформ розподіленого реєстру для інфраструктури «розумного міста».

В другому розділі кваліфікаційної роботи:

- Сформовано функціональну архітектуру МaaS-платформи.
- Запропоновано структуру МaaS-платформи «розумного міста».
- Створено модель даних прототипу МaaS-платформи «розумного міста».

- Розглянуто ланцюговий код МaaS-платформи «розумного міста».
- Проаналізовано компонентно-модульну структуру та багаторівневу архітектуру серверної частини МaaS-платформи «розумного міста».

В третьому розділі кваліфікаційної роботи:

- Описано процес розгортання та апаратно-програмне забезпечення експериментального середовища.

- Розглянуто функціональне тестування прототипу МaaS-платформи «розумного міста».
- Проведено дослідження продуктивності та навантажувальне тестування елементів МaaS-платформи «розумного міста».
- Розглянуто процес налаштування мережі блокчейн для прототипу МaaS-платформи «розумного міста».
- Подано порівняльний аналіз з іншими інноваційними технологіями.
- Описано перспективи подальших досліджень у сфері інтеграції МaaS систем.

У розділі «Безпека життєдіяльності, основи охорони праці» описано державну систему моніторингу довкілля як складова частина національної інформаційної інфраструктури, сумісної з аналогічними системами інших країн. Розглянуто вимоги ергономіки до організації робочого місця оператора ПК.

ПЕРЕЛІК ДЖЕРЕЛ

- 1 Miron, Radu, et al. "Integrating blockchain technology into mobility-as-a-service platforms for smart cities." *Smart Cities* 8.1 (2025): 9.
- 2 DELPHI Homepage. Available online: <https://delphi-project.eu/>.
- 3 Nguyen, T.; Nguyen, H.; Partala, J.; Pirttikangas, S. TrustedMaaS: Transforming trust and transparency Mobility-as-a-Service with blockchain. *Future Gener. Comput. Syst.* 2023, 149, 606–621.
- 4 López, D.; Farooq, B. A multi-layered blockchain framework for smart mobility data-markets. *Transp. Res. Part C Emerg. Technol.* 2020, 111, 588–615.
- 5 Indy Homepage. Available online: <https://www.hyperledger.org/projects/hyperledger-indy>.
- 6 Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online: <https://bitcoin.org/bitcoin.pdf>.
- 7 Tschorsch, F.; Scheuermann, B. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Commun. Surv. Tuts.* 2016, 18, 2084–2123.
- 8 Saputri, F.A. Regulating the Use of Smart Contract in Indonesia. *JHK J. Huk. Dan Keadilan* 2024, 1, 42–50.
- 9 Monrat, A.A.; Schelén, O.; Andersson, K. A Survey of Blockchain from the Perspectives of Applications, Challenges, and Opportunities. *IEEE Access* 2019, 7, 117134–117151.
- 10 Duda O., Matsiuk O., Kunanets N., Pasichnyk V., Rzhеuskyi A., Bilak Y. Formation of hypercubes based on data obtained from IoT devices // *International Journal of Sensors, Wireless Communications and Control*. 2021. Vol. 11(5). P. 498–504. DOI: 10.2174/2210327910999201210145151.
- 11 Cuš-Babić, N.; Guerra De Oliveira, S.F.; Tibaut, A. Interoperability of Infrastructure and Transportation Information Models: A Public Transport Case Study. *Appl. Sci.* 2022, 12, 6234.
- 12 Upadhyay, N. Demystifying blockchain: A critical analysis of challenges, applications and opportunities. *Int. J. Inf. Manag.* 2020, 54, 102120.

13 Alahmad, M.; Alfouderi, A.D.E.L.; Alonaizi, A.H.M.A.D.; Aldhamen, M. Comparison Study of the Top 5 Leading Cryptocurrencies based on General Consensus Protocol: Bitcoin, Ethereum, Tether, XRP and Bitcoin Cash. *WSEAS Trans. Comput. Res.* 2023, 11, 23–32.

14 Hyperledger Fabric Docs–Introduction. Available online: <https://hyperledger-fabric.readthedocs.io/en/latest/whatis.html>.

15 Buterin, V. Ethereum Whitepaper: A Next-Generation Smart Contract and Decentralized Application Platform. Ethereum Foundation. Available online: <https://ethereum.org/en/whitepaper>.

16 Enterprise Ethereum Alliance, Enterprise Ethereum Client Specification V7. Available online: <https://entethalliance.github.io/client-spec/spec.html>.

17 Ethereum Foundation, Solidity Documentation. Available online: <https://docs.soliditylang.org/en/latest>.

18 R3, Corda: A Distributed Ledger R3 Whitepaper. Available online: <https://www.r3.com/corda-platform>.

19 Jabbar, R.; Dhib, E.; Said, A.B.; Krichen, M.; Fetais, N.; Zaidan, E.; Barkaoui, K. Blockchain Technology for Intelligent Transportation Systems: A Systematic Literature Review. *IEEE Access* 2022, 10, 20995–21031.

20 Thakkar, P.; Nathan, S.; Viswanathan, B. Performance Benchmarking and Optimizing Hyperledger Fabric Blockchain Platform. In *Proceedings of the 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS)*, Milwaukee, WI, USA, 25–28 September 2018; pp. 264–276.

21 Hyperledger Fabric Docs–Using CouchDB. Available online: https://hyperledger-fabric.readthedocs.io/en/latest/couchdb_tutorial.html.

22 Mangrulkar, R.S.; Chavan, P.V. Hyperledger. In *Blockchain Essentials*; Springer Nature: Cham, Switzerland, 2024; pp. 167–201.

23 Blinowski, G.; Ojdowska, A.; Przybyłek, A. Monolithic vs. Microservice Architecture: A Performance and Scalability Evaluation. *IEEE Access* 2022, 10, 20357–20374.

24 Walls, C. *Spring in Action*, 6th ed.; Manning Publications: Shelter Island, NY, USA, 2022; ISBN 978-1617297571.

25 Hyperledger Fabric Gateway SDK for Java Homepage. Available online: <https://hyperledger.github.io/fabric-gateway-java/>.

26 Duda O., Shakleina I., Luchkevych M. AI and ML in DevOps efficiency // *Herald of Khmelnytskyi National University*. 2025. Vol. 351(3.1). P. 143–149. DOI: 10.31891/2307-5732-2025-351-17.

27 Orlov M. V., Duda O. M., Zhovnir Y. I., Hrybovskyi O. M. DevOps tools in IoT systems // *Комп'ютерно-інтегровані технології*. 2024. Vol. 57. P. 128–138. DOI: 10.36910/6775-2524-0560-2024-57-15.

28 Duda O., Pasichnyk V., Lypak H., Veretennikova N., Kunanets N., Matsiuk O., Mudrokha V. Integrated repositories of social and communication data // *CEUR Workshop Proceedings*. 2021. Vol. 2870. P. 1420–1430.

29 Zhang, J.; Zhang, J. Artificial Intelligence Applied on Traffic Planning and Management for Rail Transport: A Review and Perspective. *Discret. Dyn. Nat. Soc.* 2023, 2023, 1832501.

30 Duda O., Mykytyshyn A., Mytnyk M., Stanko A. TNTU Smart Campus services network // *CEUR Workshop Proceedings*. 2023. Vol. 3628. P. 93–105.

31 Xie, H.; Song, X.; Zhang, H. Chapter 8–MaaS and IoT: Concepts, methodologies, and applications. In *Big Data and Mobility as a Service*; Elsevier: Amsterdam, The Netherlands, 2022; pp. 229–243.

32 Karnaukhov O., Duda O., Martsenko S., Yatsyshyn V. Cyber-physical systems at Digital University // *CEUR Workshop Proceedings*. 2023. Vol. 3628. P. 306–314.

33 Faieq, S.; Saidi, R.; Elghazi, H.; Rahmani, M.D. C2IoT: A framework for Cloud-based Context-aware Internet of Things services for smart cities. *Procedia Comput. Sci.* 2017, 110, 151–158.

34 Orlov M. V., Hrybovskyi O. M., Zhovnir Y. I., Duda O. M. DevOps methodology in IoT ecosystems // *Вчені записки ТНУ*. 2024. Vol. 35(6). P. 163–170. DOI: 10.32782/2663-5941/2024.6.2/22.

35 Stepanova LS (2020) Environmental policy of Ukraine and international experience. Kyiv: Center for Economic Strategy.

36 Zhukovsky MP (2020) Modern trends in the utilization of municipal solid waste in cities. Journal "Scientific World", 12(4), pp 49–55.

37 Computer Vision Syndrome. American Optometric Association. 2024. URL: <https://www.aoa.org/healthy-eyes/eye-and-vision-conditions/computer-vision-syndrome>.

38 Mental health at work. World Health Organization. 2022. URL: <https://www.who.int/news-room/fact-sheets/detail/mental-health-at-work>.