

М. Р. БЄЛІКОВ

здобувач вищої освіти

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

ORCID: 0009-0004-1204-0349

А. О. БУР

здобувач вищої освіти

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

ORCID: 0009-0004-9320-7445

С. В. ПОПЕРЕШНЯК

кандидат фізико-математичних наук, доцент,

доцент кафедри інформатики та програмної інженерії

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

ORCID: 0000-0002-0531-9809

МІКРОСЕРВІСНА АРХІТЕКТУРА ДЛЯ ПОБУДОВИ КРИПТО-ФРІЛАНС БІРЖІ

У статті запропоновано архітектурне рішення для побудови крипто-фріланс біржі нового покоління, яке поєднує переваги мікросервісного підходу та блокчейн-технологій. Робота обґрунтовує актуальність теми у зв'язку з недоліками централізованих платформ, серед яких – високі комісії, затримки у виплатах та низький рівень довіри між користувачами. Поставлена мета полягає у створенні масштабованої, безпечної та прозорої архітектури, здатної забезпечити ефективну взаємодію замовників і виконавців через інтеграцію смарт-контрактів.

Архітектура системи реалізована на основі принципів предметно-орієнтованого проектування та представлена у вигляді модульного середовища з незалежними сервісами для автентифікації, управління замовленнями, транзакціями, файловим обміном і аналітикою. Для візуалізації використано UML- та C4-діаграми, що демонструють логіку роботи та взаємозв'язки між сервісами. Взаємодія мікросервісів здійснюється через gRPC-протокол, тоді як зовнішні API функціонують на основі HTTPS. Для зберігання даних застосовано багаторівневий підхід: PostgreSQL використовується для критичних транзакційних даних, MongoDB – для динамічних об'єктів, Redis – для кешування та роботи в реальному часі.

Особливу увагу приділено питанням безпеки. Запроваджено механізми захисту від SQL-ін'єкцій, XSS-атак, DDoS-навантажень і фішингових загроз. Для захисту інтеграції з блокчейн-гаманцями використано алгоритми підпису транзакцій та шифрування приватних ключів. Смарт-контракти забезпечують автоматичне виконання умов угоди, прозорість фінансових розрахунків і зниження ризиків шахрайства.

Результати дослідження підтвердили ефективність поєднання мікросервісної архітектури та блокчейну для створення сучасної фріланс-платформи. Запропоноване рішення демонструє високу масштабованість, відмовостійкість і можливість подальшої інтеграції з AI-сервісами та іншими зовнішніми інструментами. У висновках наголошено, що платформа може стати основою для розвитку децентралізованих фріланс-сервісів, DeFi-проектів і нових економічних моделей, де ключовими принципами є прозорість, довіра та автоматизація.

Ключові слова: Інтерфевебдодаток, програмна архітектура, мікросервіси, фріланс, блокчейн, криптовалюта, смарт-контракти, безпека даних, Internet of Everything.

М. R. BIELIKOV

Higher Education Student

National Technical University of Ukraine

“Ihor Sikorsky Kyiv Polytechnic Institute”

ORCID: 0009-0004-1204-0349

А. О. BUR

Higher Education Student

National Technical University of Ukraine

“Ihor Sikorsky Kyiv Polytechnic Institute”

ORCID: 0009-0004-9320-7445

S. V. POPERESHNYAK

Candidate of Physical and Mathematical Sciences, Associate Professor,
Associate Professor at the Department of Informatics
and Software Engineering
National Technical University of Ukraine
“Ihor Sikorsky Kyiv Polytechnic Institute”
ORCID: 0000-0002-0531-9809

MICROSERVICES ARCHITECTURE FOR BUILDING A CRYPTO FREELANCE EXCHANGE

The article proposes an architectural solution for building a next-generation crypto-freelance exchange that combines the advantages of a microservices approach with blockchain technologies. The study highlights the relevance of the topic, driven by the shortcomings of centralized platforms such as high service fees, payment delays, and low trust between users. The main goal is to design a scalable, secure, and transparent architecture capable of ensuring effective interaction between clients and freelancers through the integration of smart contracts.

The system architecture is developed on the principles of domain-driven design and presented as a modular environment with independent services for authentication, order management, transaction processing, file handling, and analytics. UML and C4 diagrams are employed to visualize the system's logic and inter-service interactions. Communication between microservices is established via the gRPC protocol, while external APIs operate over HTTPS. A multi-layered data storage approach is applied: PostgreSQL is used for critical transactional data, MongoDB for dynamic objects, and Redis for caching and real-time processing.

Special attention is given to security issues. Protection mechanisms against SQL injections, XSS attacks, DDoS loads, and phishing threats have been implemented. Blockchain wallet integration is safeguarded through transaction signing algorithms and private key encryption. Smart contracts ensure automatic fulfillment of agreements, transparent financial operations, and reduced fraud risks.

The research results confirm the effectiveness of combining microservices architecture with blockchain in creating a modern freelance platform. The proposed solution demonstrates high scalability, fault tolerance, and the potential for further integration with AI services and other external tools. The conclusions emphasize that such a platform can serve as a foundation for decentralized freelance services, DeFi projects, and innovative economic models, where transparency, trust, and automation are the defining principles.

Key words: web application, software architecture, microservices, freelance, blockchain, IoT.

Постановка проблеми

Сфера фріланс-послуг переживає стрімке зростання, однак наявні централізовані платформи продовжують стикатися з низкою системних проблем. Серед найбільш критичних – високі комісійні збори, затримки з виведенням коштів, а також уразливість до шахрайських дій з боку як замовників, так і виконавців. Крім того, централізований контроль таких платформ ускладнює процес арбітражу та викликає недовіру до прозорості рішень адміністрації.

Зважаючи на це, все більш актуальними стають рішення, що базуються на децентралізованих технологіях, таких як блокчейн та смарт-контракти. Вони дозволяють автоматизувати виконання умов контракту, забезпечити прозорість транзакцій, зменшити вплив посередників і підвищити рівень довіри між сторонами. Однак повна децентралізація часто знижує продуктивність системи, ускладнює масштабування та створює технічні труднощі, пов'язані з інтеграцією зі звичними веб-інтерфейсами.

Класичні архітектурні рішення для побудови цифрових платформ (монолітні або багатофункціональні гібридні системи) часто не здатні забезпечити гнучкість, масштабованість і швидке оновлення окремих модулів. Це створює потребу у впровадженні мікросервісної архітектури, яка дозволяє розподілити функціональність на незалежні сервіси з чітко визначеними зонами відповідальності, що легко масштабуються, тестуються та підтримуються. Водночас відсутність узагальнених підходів до інтеграції мікросервісів із блокчейн-рішеннями в контексті фріланс-платформ створює наукову прогалину, яку необхідно заповнити.

Отже, постає актуальна інженерна та наукова проблема – розробка архітектурного рішення, що поєднує масштабовану мікросервісну інфраструктуру з надійною криптографічною логікою на базі блокчейн-технологій для створення безпечної, прозорої та ефективної фріланс-платформи нового покоління.

Формулювання мети дослідження

Метою дослідження є проектування масштабованої, безпечної та ефективної мікросервісної архітектури для крипто-фріланс платформи, яка усуває ключові недоліки сучасних фріланс-сервісів – зокрема, високі комісії, затримки виплат і ризики шахрайства – шляхом інтеграції блокчейн-технологій і смарт-контрактів.

Для досягнення цієї мети ставляться такі основні завдання:

1. Сформулювати вимоги до архітектури платформи з урахуванням функціональних і нефункціональних аспектів, зокрема безпеки, масштабованості та гнучкості розгортання.

2. Розробити архітектуру, побудовану на мікросервісному підході, з розподілом на незалежні сервіси для ключових бізнес-функцій.

3. Інтегрувати блокчейн-механізми, зокрема смарт-контракти, для управління замовленнями, транзакціями та взаємодією між користувачами.

4. Побудувати діаграми архітектури (UML, C4) для візуалізації компонентів системи, їхньої взаємодії та логіки роботи.

Запропоноване рішення має слугувати технічною основою для створення децентралізованої, надійної та прозорої цифрової платформи для фріланс-взаємодії.

Аналіз останніх досліджень і публікацій

Сучасні дослідження зосереджені на використанні блокчейн-технологій та смарт-контрактів у цифрових платформах, зокрема у сфері фрілансу. У роботі [1] запропоновано поєднання блокчейну з біометричними технологіями для зменшення шахрайства на фріланс-сайтах. У роботі [2] акцентують увагу на методології Domain-Driven Design, що дозволяє більш чітко пов'язати бізнес-вимоги з архітектурою мікросервісних систем. Аналітичний звіт [3] демонструє значне зростання обсягів ринку фріланс-послуг у 2025 році, що підтверджує актуальність пошуку масштабованих і безпечних платформ. У дослідженні [4] подано узагальнений огляд архітектури смарт-контрактів, їхніх застосувань і перспектив. У роботі [5] описали платформу HireChain на базі Ethereum, орієнтовану на скорочення затримок у виплатах. Систему Boomerang розробили в [6], яка використовує Hyperledger для підвищення довіри й безпеки, проте характеризується зниженими показниками продуктивності. У роботі [7] акцентують увагу на довірі та прозорості у децентралізованих фріланс-системах. Окрему групу становлять праці, що торкаються суміжних питань захисту даних і обробки інформації. Так, у [8] розглянули методи підвищення криптографічної надійності шляхом контролю генерації випадкових чисел, а в [9] дослідили методи доступу й обробки мультимедійного контенту у хмарних середовищах.

Таким чином, наявні роботи доводять перспективність інтеграції блокчейну та мікросервісних підходів у побудові децентралізованих платформ, але водночас вказують на проблеми продуктивності, масштабованості й захисту, які потребують подальших досліджень.

Викладення основного матеріалу дослідження

Архітектура системи. Запропонована архітектура системи для крипто-фріланс біржі базується на мікросервісній архітектурі з використанням принципів предметно-орієнтованого проектування. Основною моделлю є багаторівнева архітектура, яка забезпечує гнучкість, масштабованість та зручність обслуговування. Для візуалізації системи використовується модель C4 на трьох рівнях абстракції, що демонструє компоненти, контейнери та внутрішню структуру сервісів. На найвищому рівні (рисунок 1) система представлена як єдине рішення, що взаємодіє із зовнішніми учасниками та сервісами, такими як блокчейн Ethereum, сервіси генерації тексту на основі великих мовних моделей, які використовуються для автоматичного покращення описів завдань, узагальнення контрактів та інтелектуальних пропозицій, а також зовнішні поштові провайдери. Поняття «Користувач» узагальнене і включає адміністраторів, замовників, фрілансерів та гостей – усі вони взаємодіють із системою через єдину точку входу.

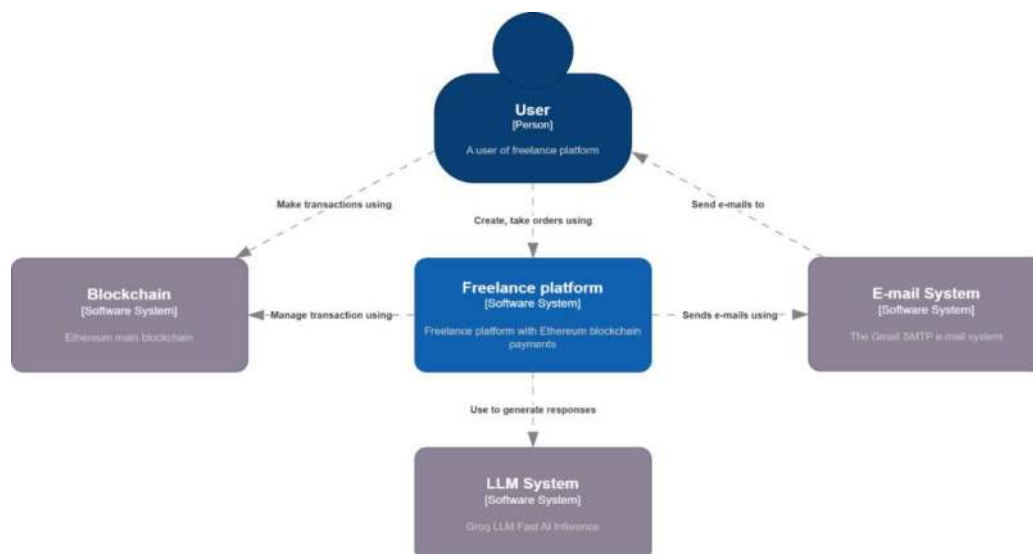


Рис. 1. C4 рівень 1, діаграма крипто-фріланс біржі

Джерело: розроблено авторами

Контейнери.

На рівні C4 Рівень 2 (Рисунок 2) кожен контейнер розкладається на компоненти, з акцентом на модульність та розділення відповідальностей через gRPC для внутрішніх викликів та HTTPS/JSON для зовнішніх API.

Архітектура включає:

1. Вебзастосунок – фронтенд-інтерфейс, побудований на Next.js, що забезпечує адаптивний інтерфейс користувача для всіх типів користувачів платформи.

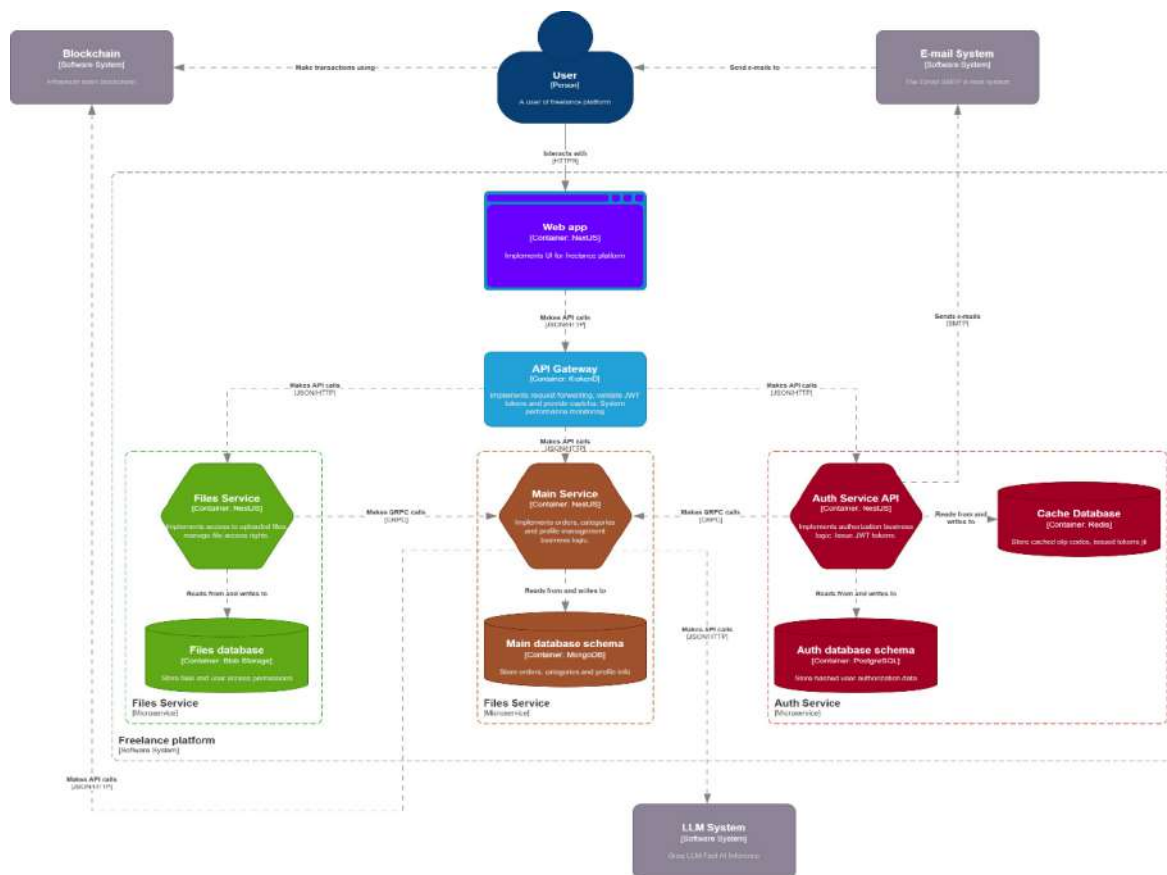


Рис. 2. C4 рівень 2, діаграма контейнерів основних мікросервісів

Джерело: розроблено авторами

2. API Gateway – центральна точка входу для маршрутизації HTTP/gRPC запитів до бекенд-сервісів, обробляє автентифікацію, обмеження швидкості запитів та спостережуваність.

3. Сервіс автентифікації – відповідає за автентифікацію та авторизацію користувачів, включаючи вхід, реєстрацію, багатофакторну автентифікацію, блокування акаунтів та видачу JWT. Інтегрується з PostgreSQL, Redis та Gmail SMTP для верифікації електронної пошти і MFA.

4. Головний сервіс – реалізує основну бізнес-логіку: керування проектами, ролями користувачів, вирішення суперечок, взаємодію зі смарт-контрактами та AI-перевірку. Використовує MongoDB для гнучкого та масштабованого збереження даних, а також взаємодіє з Ethereum через Web3.js і LLM API від Groq для генерації тексту та перевірок тексту на спам.

5. Сервіс файлів – керує завантаженням, зберіганням (через Blob Storage) і безпечним доступом до файлів, що завантажуються користувачами.

Взаємодія між сервісами відбувається.

Компоненти.

На рівні C4 Рівень 3 (Рисунок 3) кожен контейнер розкладається на компоненти, з акцентом на модульність та розділення відповідальностей.

Кожен мікросервіс включає:

1. Контролери – для обробки вхідних API/gRPC запитів.
2. Сервіси – які інкапсулюють бізнес-логіку.
3. Репозиторії або шари доступу до даних – для взаємодії з базами даних.
4. Клієнти або адаптери – для зовнішніх сервісів (SMTP, Ethereum, LLM).

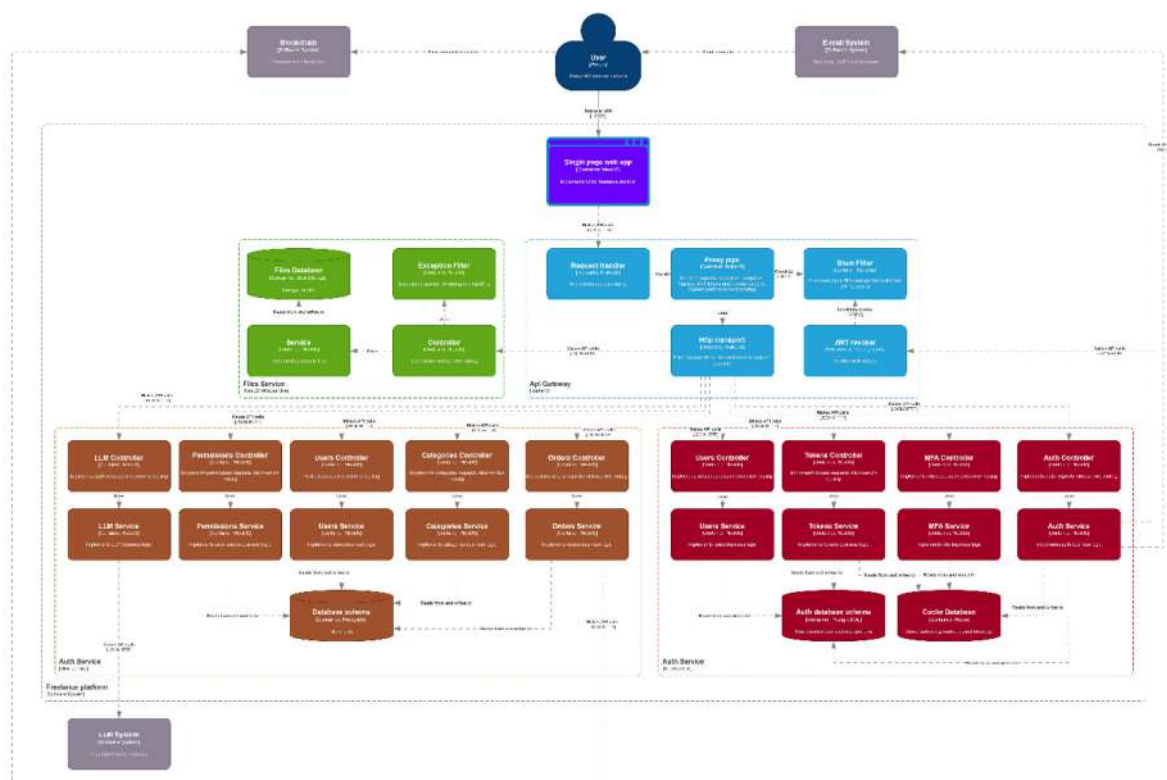


Рис. 3. C4 рівень 3, внутрішня архітектура всіх мікросервісів

Джерело: розроблено авторами

Для взаємодії між сервісами обрано gRPC замість REST через його низьку затримку та ефективний бінарний протокол (Protocol Buffers). Для зовнішніх API використовується HTTPS, що забезпечує шифрування даних і автентифікацію сервера.

Протокол gRPC використовується для передачі даних між мікросервісами через його численні переваги. По-перше, gRPC забезпечує високу швидкість обміну даними завдяки використанню протоколу HTTP/2, що підтримує мультиплексування потоків і ефективне управління з'єднаннями. Це особливо важливо в мікросервісній архітектурі, де кожен сервіс повинен взаємодіяти з іншими для виконання складних операцій. Крім того, gRPC використовує серіалізацію через Protocol Buffers, що дозволяє передавати менший обсяг даних і швидше обробляти запити. Це знижує затримки між запитами та відповідями, підвищуючи загальну продуктивність системи.

Аналіз безпеки даних. Було враховано потенційні ризики та забезпечено захист від поширених загроз шляхом інтеграції сучасних технологій безпеки.

Можливі атаки, які можуть бути спрямовані на систему, включають SQL-ін'єкції, атаки міжсайтового скриптування (XSS), атаки грубої сили (DDoS), фішинг, експлуатацію вразливостей у сторонньому програмному забезпеченні, атаки на блокчейн-гаманці та перехоплення даних через атаку «людина посередині» (Man-in-the-Middle).

Для захисту від SQL-ін'єкцій система використовує параметризовані запити та ORM-інструменти, що мінімізують ризик виконання шкідливих команд у базі даних. Вразливості XSS нейтралізуються шляхом ретельної перевірки та очищення вхідних даних, а також шифруванням конфіденційної інформації перед її відображенням на фронтенді.

Для боротьби з DDoS атаками при реєстрації та авторизації використовуються обмеження кількості спроб входу, капчі та двофакторна автентифікація. Ризики фішингу мінімізуються шляхом шифрування даних через SSL і TLS.

Для захисту інтеграції з блокчейн-гаманцями реалізовано алгоритми підпису транзакцій, шифрування приватних ключів і взаємодію лише через офіційні API-інтерфейси. Захист від атак типу Man-in-the-Middle забезпечується використанням HTTPS. Цілісність і конфіденційність даних підтримується шляхом шифрування зберігання конфіденційної інформації з використанням AES-256 та bcrypt.

Архітектура баз даних. У розробленій архітектурі системи використовується поєднання реляційних і нереляційних баз даних для забезпечення як цілісності даних, так і гнучкості роботи. PostgreSQL виступає основним сховищем для структурованих даних, забезпечуючи надійні гарантії узгодженості, підтримку складних запитів і потужне управління транзакціями. Він використовується зокрема мікросервісом автентифікації (Рисунок 4), де

критичною є надійність, суворе дотримання схеми даних і безпека для управління обліковими записами користувачів та інформацією про сесії.

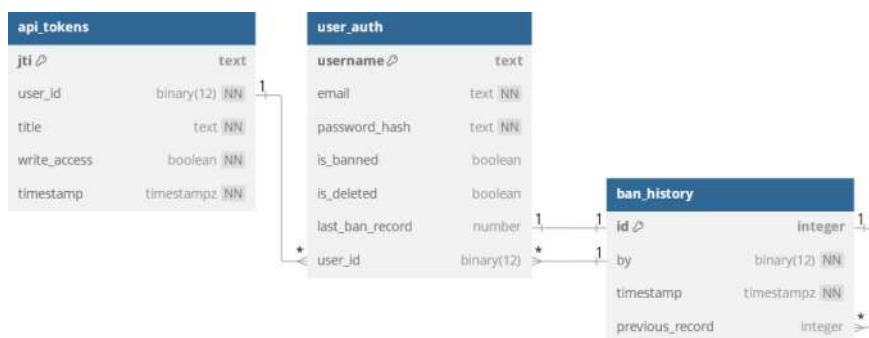


Рис. 4. ER-діаграма PostgreSQL мікросервісу автентифікації

Джерело: розроблено авторами

З іншого боку, MongoDB використовується в основному мікросервісі (Рисунок 5) для ефективної роботи з динамічними й напівструктурованими даними. Вона забезпечує легку адаптацію системи до змін у структурі даних без потреби в складних міграціях. Такий вибір особливо важливий для роботи з мінливими наборами даних, такими як інформація про паркомісця в реальному часі, результати виявлення автомобілів і журнали взаємодії користувачів.

Крім того, Redis інтегровано як сховище даних в оперативній пам'яті для кешування тимчасових і часто використовуваних даних. Зберігаючи стани сесій, проміжні результати обчислень і оновлення статусу в реальному часі в Redis, система значно знижує навантаження на запити до бази даних і прискорює час відгуку, тим самим підвищуючи загальну продуктивність і масштабованість.

Використання двох спеціалізованих баз даних – PostgreSQL для підсистеми автентифікації та MongoDB для основних функціональних сервісів – забезпечує баланс між сильною узгодженістю даних там, де це потрібно, і високою гнучкістю й продуктивністю, необхідною для динамічної роботи.

Процес обробки замовлення зі смарт-контрактами. Повний процес замовлення показано на Рисунку 6.

Замовник створює завдання на сайті платформи, і смарт-контракт створюється на адресу його гаманця, який блокує суму грошей, яку він вказав (бюджет на виконання). Замовник може скасувати замовлення. Це можна зробити як на сайті вебзастосунку, так і самостійно, викликавши функцію скасування в блокчейні через криптогаманець. У такому разі кошти будуть розблоковані, а смарт-контракт вважатиметься завершеним.

Інакше замовник чекатиме, доки один із фрілансерів прийме замовлення на нашій платформі. Коли це станеться, він отримає відповідне повідомлення. Замовник повинен відхилити або прийняти фрілансера, який хоче виконати завдання. Після вибору виконавця створюється запис у блокчейні. Безпека вибору гарантована, оскільки функція «взяти замовлення» може бути викликана лише з гаманця, який використовувався для ініціації смарт-контракту.

Фрілансер, у свою чергу, може відмовитися від виконання завдання, викликавши функцію «скасувати замовлення» через гаманець, підключений до платформи, але це погіршить його рейтинг. Потім замовник може чекати й обрати іншого фрілансера або скасувати завдання. Автоматичне скасування спрацює, коли спливе встановлений замовником термін.

Фрілансер повинен прикріпити файли з виконаним завданням. Замовник перевіряє їх, і якщо обидві сторони згодні, кошти перераховуються виконавцю. Якщо виникає конфлікт, втручається адміністратор платформи, який повинен перевірити історію чату й вирішити, чи виконав фрілансер поставлене завдання. Якщо так – гроші надсилаються виконавцю, і смарт-контракт завершується. Якщо ні – адміністратор обговорює зі сторонами конфлікту, чи дозволити поточному виконавцю внести зміни до завдання та продовжити роботу. У разі згоди виконавець має прикріпити нові файли до дедлайну або відмовитися від виконання.

Якщо сторони конфлікту не досягли згоди, адміністратор приймає рішення – чи дозволити фрілансеру брати нові замовлення на платформі, і має право його заблокувати. Виконавця виключають зі смарт-контракту. Замовник може чекати нового фрілансера. Проте, якщо завдання більше неактуальне, він може скасувати його. Кошти будуть розблоковані, і смарт-контракт вважатиметься завершеним.

Незважаючи, на досить великий список обмежень і викликів з якими стикаються науковці дана тема має перспективи, які сформовані на основі сучасного стані розвитку та прогресу.

Перспективи подальших досліджень.

1. Розвиток гібридних моделей. Поєднання різних підходів, таких як глибокі нейронні мережі та традиційні алгоритми оптимізації, може дозволити отримати кращі результати в адаптації інтерфейсів. Гібридні моделі, що

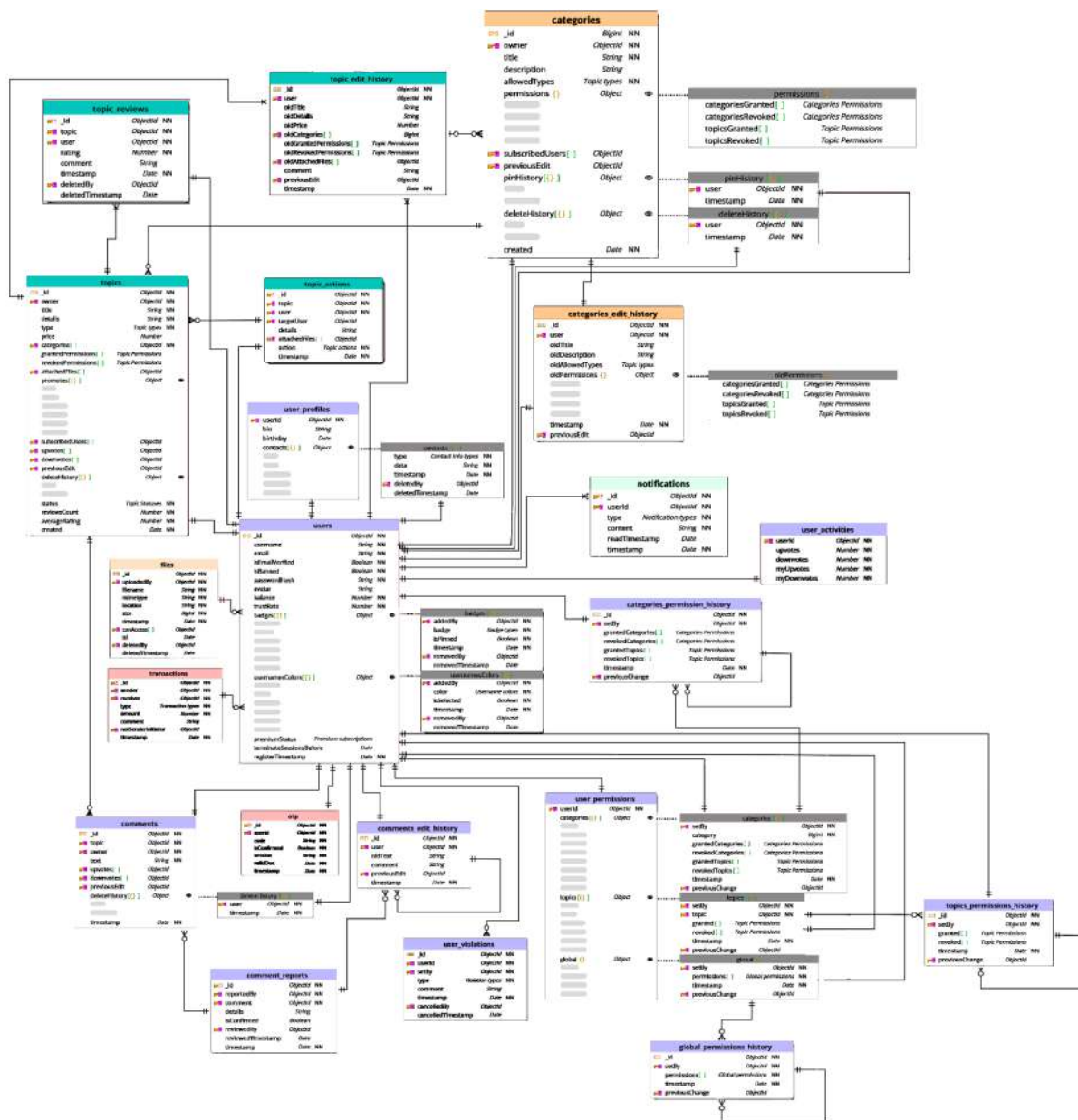


Рис. 5. ER-діаграма MongoDB головного мікросервісу

Джерело: розроблено авторами

використовують одночасно кілька методів аналізу поведінкових даних, зможуть враховувати більшу кількість змінних і краще реагувати на зміни в поведінці користувачів.

2. Розробка мультимодальних адаптивних інтерфейсів. Дослідження можливостей створення адаптивних інтерфейсів, що реагують не тільки на поведінкові дані, але й на контекст використання (наприклад, місце, час, пристрій), можуть значно покращити якість взаємодії користувача з додатком. Використання датчиків і сенсорних систем для адаптації інтерфейсів відкриває нові перспективи для інтеграції різних типів даних.

3. Етичні питання та захист даних. З огляду на зростаюче використання поведінкових даних, важливо розробляти нові підходи до забезпечення конфіденційності та безпеки даних користувачів. Дослідження в цьому напрямі можуть допомогти впровадити більш ефективні механізми захисту даних без погіршення якості адаптації інтерфейсів.

4. Автоматизація процесів збору та аналізу даних. Подальший розвиток автоматизованих систем збору поведінкових даних дозволить знизити витрати на аналітику та покращити точність отриманих результатів. Системи, що автоматично збирають і аналізують дані, а потім адаптують інтерфейси у реальному часі, є перспективним напрямком для покращення користувацького досвіду.

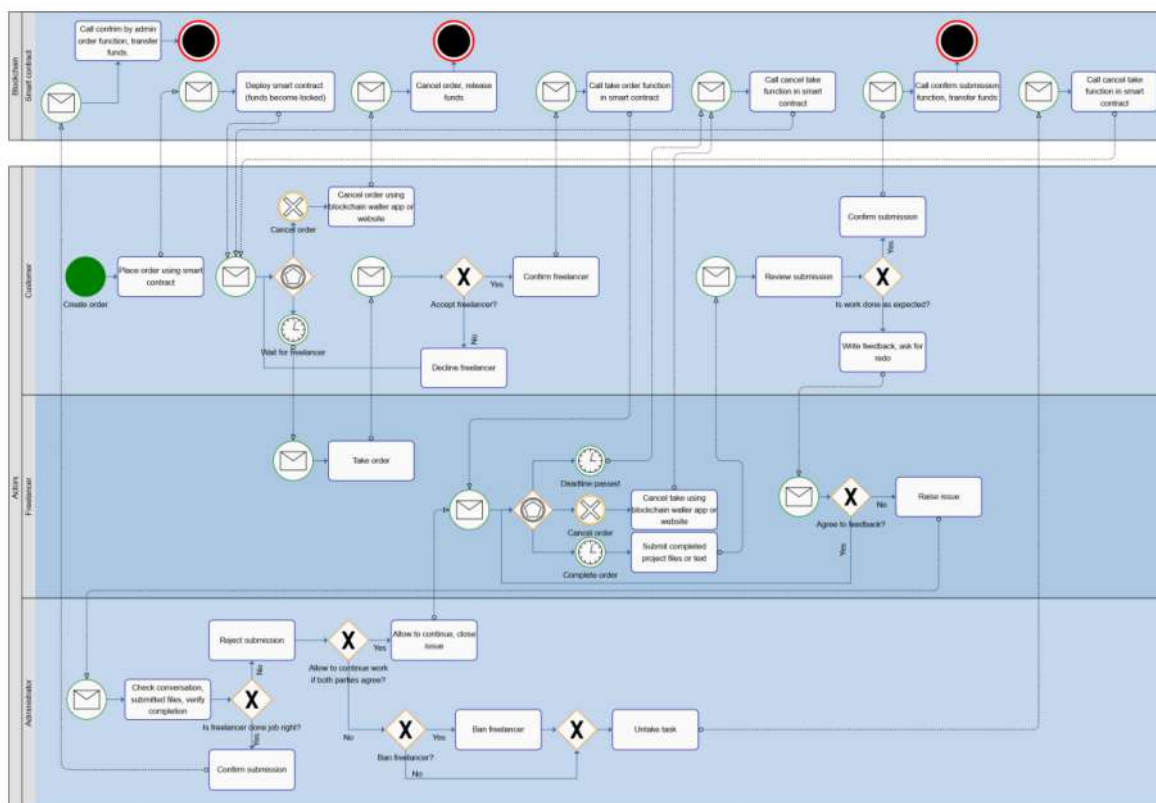


Рис. 6. BPMN-діаграма процесу виконання фріланс-замовлення

Джерело: розроблено авторами

Загалом, подальші дослідження мають бути спрямовані на покращення ефективності алгоритмів оптимізації, збільшення захисту даних та забезпечення гнучкості систем при адаптації інтерфейсів до змін у поведінці користувачів.

Висновки і перспективи подальших розвідок у даному напрямі

У межах даного дослідження розроблено архітектурне рішення для створення децентралізованої фріланс-платформи, що поєднує принципи мікросервісної інженерії та блокчейн-технологій. Запропонована система демонструє здатність ефективно масштабуватись, забезпечувати високий рівень безпеки та прозорості взаємодії між користувачами шляхом інтеграції смарт-контрактів. Завдяки модульній структурі, кожен компонент архітектури може розвиватися незалежно, що значно спрощує супровід і подальше розширення функціональності. Платформа досягає високої масштабованості та відмовостійкості завдяки модульному підходу. Інтеграція блокчейн-технологій для транзакцій забезпечує надійність надання послуг та високий рівень безпеки.

Використання gRPC-протоколу, багаторівневого підходу до зберігання даних (PostgreSQL, MongoDB, Redis) та C4-моделювання дозволило досягти гнучкості у проектуванні та наочно представити логіку роботи системи. Результати дослідження підтверджують доцільність застосування мікросервісного підходу у сфері цифрових платформ, зокрема для сервісів, що вимагають надійних фінансових розрахунків і мінімізації людського втручання. Кожен компонент системи спроектований для незалежної роботи. Розподіл обов'язків спрощує майбутнє розширення системи та полегшує інтеграцію з новими зовнішніми сервісами. Смарт-контракти додають додатковий рівень довіри та прозорості, захищаючи інтереси всіх залучених сторін. Механізм гарантує безпечне зберігання коштів та їхнє розблокування лише після успішного виконання завдання, що знижує ризики шахрайства.

Запропонована архітектура може слугувати основою для розробки новітніх платформ у сфері фрілансу, децентралізованих фінансових сервісів (DeFi) та смарт-економіки в цілому.

Платформа, відповідає сучасним вимогам індустрії та закладає надійну основу для постійного вдосконалення та адаптації до новітніх технологій.

Список використаної літератури

1. Batool A., Byun Y. Reduction of Online Fraudulent Activities in Freelancing Sites Using Blockchain and Biometric. *Electronics*. 2022. Vol. 11, No. 5. Art. 789. DOI: <https://doi.org/10.3390/electronics11050789>

2. Jaiswal S. K., Agrawal R. Domain-Driven Design (DDD) – Bridging the Gap between Business Requirements and Object-Oriented Modeling. *International Journal of Innovative Research in Engineering and Management (IJIREM)*. 2024. Vol. 11, No. 2. P. 79–83. DOI: <https://doi.org/10.55524/ijirem.2024.11.2.16>
3. Kumar N. Latest Freelance Statistics 2025 – Industry Size & Trends. 2025. URL: <https://www.demandsage.com/freelance-statistics/> (дата звернення: 18.09.2025).
4. Wang S., Yuan Y., Wang X., Li J., Qin R., Wang F.-Y. An Overview of Smart Contract: Architecture, Applications, and Future Trends. *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, Changshu, China, 2018. P. 108–113. DOI: <https://doi.org/10.1109/IVS.2018.8500488>
5. Deshmukh P., Kalwaghe S., Appa A., Pawar A. Decentralised Freelancing using Ethereum Blockchain. *Proceedings of the International Conference on Communication and Signal Processing (ICCSP)*, Chennai, India, 2020. P. 881–883. DOI: <https://doi.org/10.1109/ICCSP48568.2020.9182127>
6. Pallam B., Gore M. M. Boomerang: Blockchain-based Freelance Paradigm on Hyperledger. *Proceedings of the 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, Kanpur, India, 2019. P. 1–6. DOI: <https://doi.org/10.1109/ICCCNT45670.2019.8944572>
7. Gandhi M., та ін. Decentralized Freelancing System – Trust and Transparency. *International Research Journal of Engineering and Technology (IRJET)*. 2019. Vol. 6, No. 9.
8. Popereshnyak S., Novikov Y., Zhdanova Y. Cryptographic system security approaches by monitoring the random numbers generation. *CEUR Workshop Proceedings*. 2024. Vol. 3826. P. 301–309. URL: <https://ceur-ws.org/Vol-3826/short21.pdf> (дата звернення: 18.09.2025).
9. Reshetniak M., Popereshnyak S. Method for accessing and processing multimedia content in a cloud environment. *Proceedings of the IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology*. 2019. P. 71–76. DOI: <https://doi.org/10.1109/PICST47496.2019.9061463>

References

1. Batool, A., & Byun, Y. (2022). Reduction of online fraudulent activities in freelancing sites using blockchain and biometric. *Electronics*, 11(5), 789. <https://doi.org/10.3390/electronics11050789>
2. Jaiswal, S. K., & Agrawal, R. (2024). Domain-driven design (DDD): Bridging the gap between business requirements and object-oriented modeling. *International Journal of Innovative Research in Engineering and Management*, 11(2), 79–83. <https://doi.org/10.55524/ijirem.2024.11.2.16>
3. Kumar, N. (2025). *Latest freelance statistics 2025–Industry size & trends*. Retrieved September 18, 2025, from <https://www.demandsage.com/freelance-statistics/>
4. Wang, S., Yuan, Y., Wang, X., Li, J., Qin, R., & Wang, F.-Y. (2018). An overview of smart contract: Architecture, applications, and future trends. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)* (pp. 108–113). IEEE. <https://doi.org/10.1109/IVS.2018.8500488>
5. Deshmukh, P., Kalwaghe, S., Appa, A., & Pawar, A. (2020). Decentralised freelancing using Ethereum blockchain. In *Proceedings of the International Conference on Communication and Signal Processing (ICCSP)* (pp. 881–883). IEEE. <https://doi.org/10.1109/ICCSP48568.2020.9182127>
6. Pallam, B., & Gore, M. M. (2019). Boomerang: Blockchain-based freelance paradigm on Hyperledger. In *Proceedings of the 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICCCNT45670.2019.8944572>
7. Gandhi, M., et al. (2019). Decentralized freelancing system – Trust and transparency. *International Research Journal of Engineering and Technology*, 6(9).
8. Popereshnyak, S., Novikov, Y., & Zhdanova, Y. (2024). Cryptographic system security approaches by monitoring the random numbers generation. *CEUR Workshop Proceedings*, 3826, 301–309. <https://ceur-ws.org/Vol-3826/short21.pdf>
9. Reshetniak, M., & Popereshnyak, S. (2019). Method for accessing and processing multimedia content in a cloud environment. In *Proceedings of the IEEE International Scientific-Practical Conference: Problems of Infocommunications Science and Technology (PIC S&T)* (pp. 71–76). IEEE. <https://doi.org/10.1109/PICST47496.2019.9061463>

Дата першого надходження рукопису до видання: 27.09.2025
 Дата прийнятого до друку рукопису після рецензування: 24.10.2025
 Дата публікації: 28.11.2025