



Вступ

- Шановна комісіє, представляю до вашої уваги роботу на тему «Створення блокчейн-рішення для страхових смарт-контрактів». Робота присвячена автоматизації параметричного страхування на блокчейні. Традиційне страхування має багато недоліків. Моя система CLAIMLESS вирішує це через смарт-контракти які виплачують відшкодування автоматично



Мета та завдання дослідження

- Метою роботи є створення системи страхових смарт-контрактів CLAIMLESS.
У роботі досліджено сучасні підходи до автоматизації страхових послуг. Проаналізовано використання блокчейн-рішень у сфері страхування. Розроблено архітектуру системи та смарт-контракти. Також реалізовано механізм автоматичної перевірки страхових випадків.

Аспект	Традиційні системи	Сучасні InsurTech-системи
Час обробки претензії	30–90 днів	1–48 годин
Рівень автоматизації	Низький (значна частка ручної праці)	Високий (AI (штучний інтелект) та RPA (роботизована автоматизація процесів) до 90%)
Джерела даних	Внутрішні бази, паперові документи	Інтеграція з API, IoT (Інтернет речей), зовнішніми оракулами
Прозорість операцій	Обмежена, суб'єктивна оцінка	Повна, на основі об'єктивних параметрів
Вартість адміністрування	Висока (30–40% премій)	Знижена (5–15% завдяки автоматизації)

Теоретичні аспекти автоматизації страхування

- У таблиці 1.1 наведено порівняльну характеристику традиційних та сучасних систем автоматизації страхування. Традиційні системи мають централізовану структуру та високі операційні витрати. Блокчейн-рішення забезпечують прозорість та автоматичне виконання операцій. Смарт-контракти дозволяють мінімізувати людський фактор. Проведений аналіз підтвердив ефективність децентралізованих систем.



Аспект	Переваги	Недоліки
Захист прав сторін	Високий рівень юридичної формалізації	Тривалі терміни розгляду та виплат
Оцінка збитків	Індивідуальний підхід до кожного випадку	Суб'єктивність оцінок та можливі спори
Прозорість	Наявність письмових договорів	Закритість внутрішніх процесів для страхувальника
Операційні витрати	Накопичений досвід і відпрацьовані процедури	Високі адміністративні витрати (до 30–40 % премії)
Доступність	Широка мережа агентів	Обмежена оперативність у кризових ситуаціях

Аналіз традиційних страхових моделей

- На рисунку 1.2 представлено традиційний процес розгляду страхової заяви. Процес включає перевірку документів, участь операторів та тривалий аналіз випадку. У таблиці 1.3 визначено переваги та недоліки традиційних страхових контрактів. Основними недоліками є затримки виплат та високі адміністративні витрати. Це обґрунтовує необхідність використання смарт-контрактів у страхуванні.



Платформа	EVM-сумісність	Середня вартість газу (gwei)	Приблизна швидкість TPS	Підтримка тестнету та оракулів	Зрілість екосистеми для страхових смарт-контрактів
Polygon Amoy	Повна	20–50	50–7000	Відмінна	Висока
Ethereum Mainnet	Повна	20–100+	15–30	Відмінна	Найвища
Binance Smart Chain	Повна	5–10	100–300	Добра	Середня
Solana	Ні	Дуже низька	1000+	Обмежена	Низька (інша мова)

Блокчейн-рішення для параметричного страхування

- На рисунку 1.3 показано варіанти використання блокчейн-рішення для параметричного страхування. Система передбачає взаємодію користувача, смарт-контракту та оракула. У таблиці 1.6 проведено аналіз блокчейн-платформ для реалізації проєкту. Для системи CLAIMLESS обрано платформу Polygon. Основними перевагами стали низькі gas-комісії та висока швидкість транзакцій.

Архітектура смарт-контрактів системи

- На рисунку 2.1 представлено структуру взаємодії смарт-контрактів у системі CLAIMLESS.

Система включає контракти FlightInsurance, WeatherInsurance та MockOracle.

На рисунку 2.3 показано архітектуру смарт-контракту FlightInsurance.

Контракт реалізує автоматичну перевірку затримки авіарейсу.

При затримці понад 180 хвилин система автоматично виконує виплату.

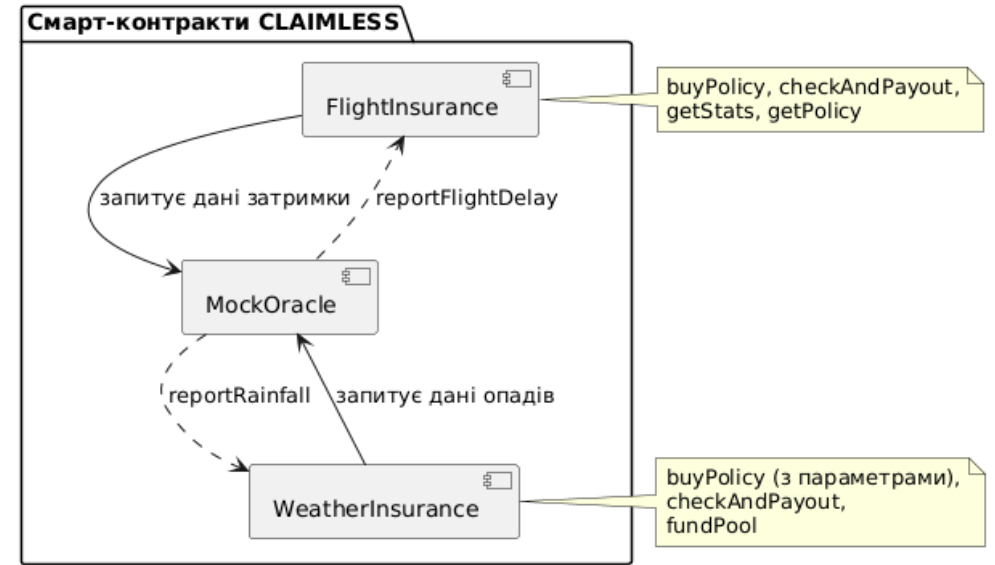


Рисунок 2.1 – Структурна схема взаємодії смарт-контрактів у системі CLAIMLESS

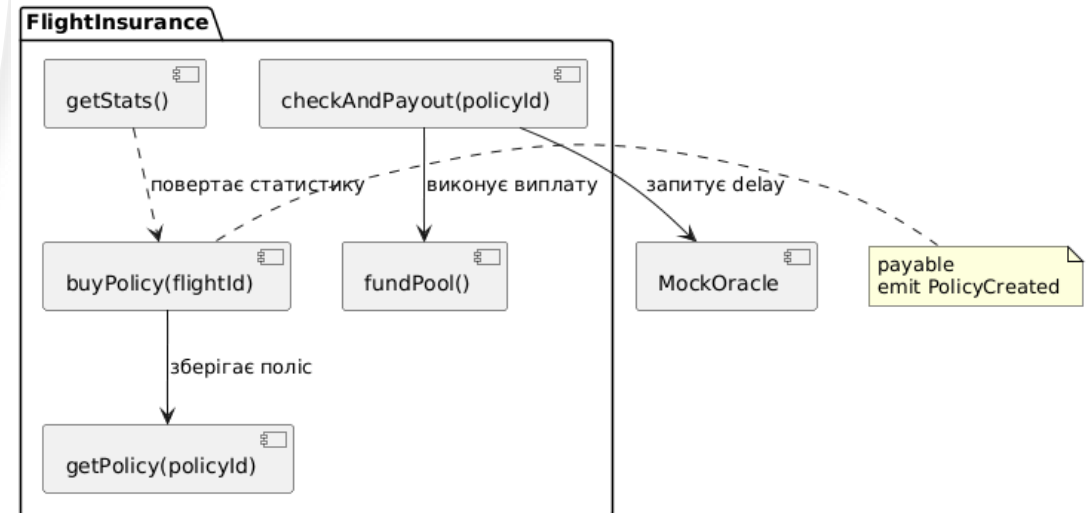


Рисунок 2.3 – Архітектура смарт-контракту FlightInsurance

Програмна архітектура системи

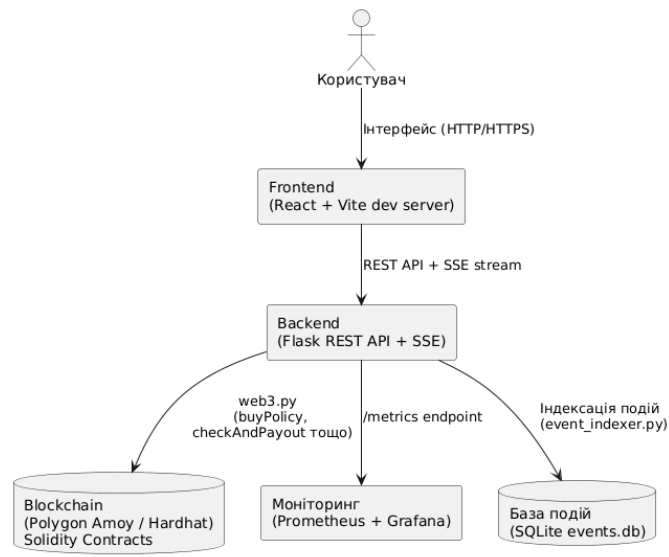


Рисунок 2.6 – Архітектурна схема програмної системи страхових смарт-контрактів

- На рисунку 2.6 наведено архітектурну схему програмної системи.
Система складається з frontend, backend та blockchain-рівня.
Рисунок 2.9 демонструє діаграму компонентів програмної системи.
React-клієнт взаємодіє зі смарт-контрактами через web3.py та API.
Модульна архітектура забезпечує масштабованість системи.

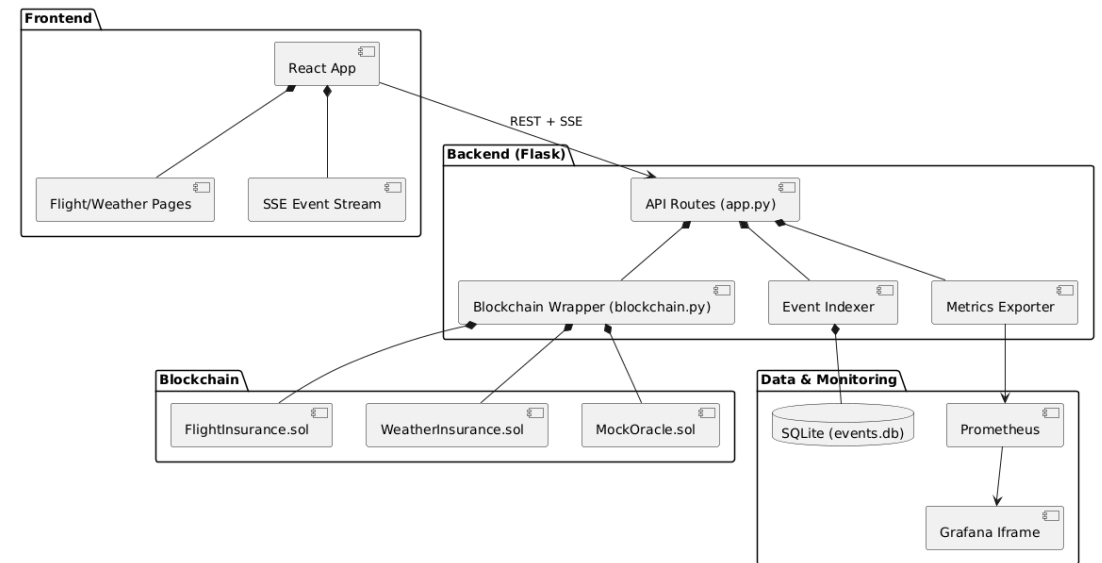


Рисунок 2.9 – Діаграма компонентів програмної системи страхових смарт-контрактів

Логіка функціонування системи

- На рисунку 2.12 показано життєвий цикл страхового полісу. Система автоматично змінює стани полісу після підтвердження страхового випадку. На рисунку 2.16 представлено технологічний стек проєкту CLAIMLESS. Для реалізації використано Solidity, Polygon, React та FastAPI. Технологічна архітектура забезпечує швидку роботу системи.

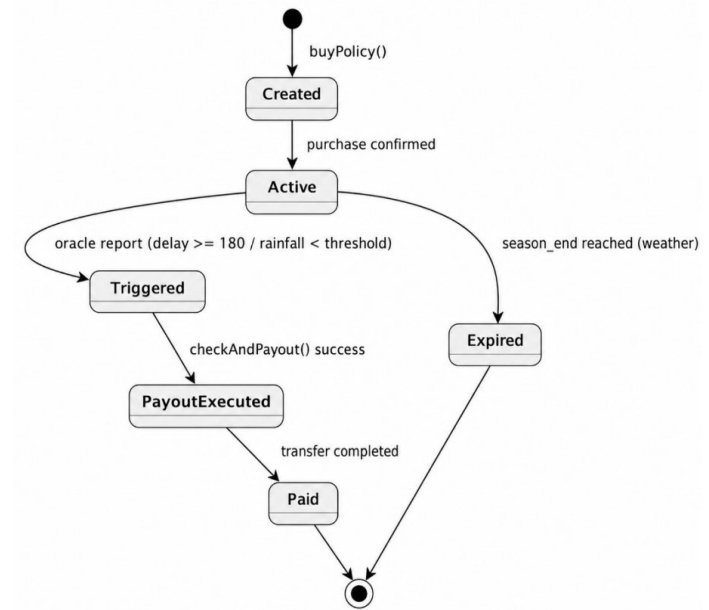


Рисунок 2.12 – Діаграма станів програмної системи страхових смарт-контрактів

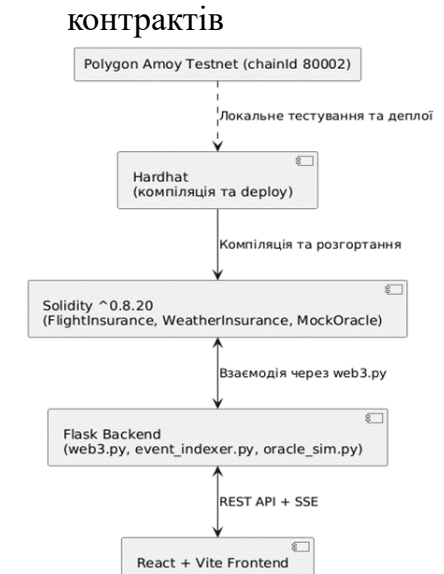


Рисунок 2.16 – Архітектура технологічного стеку блокчейн-рішення CLAIMLESS

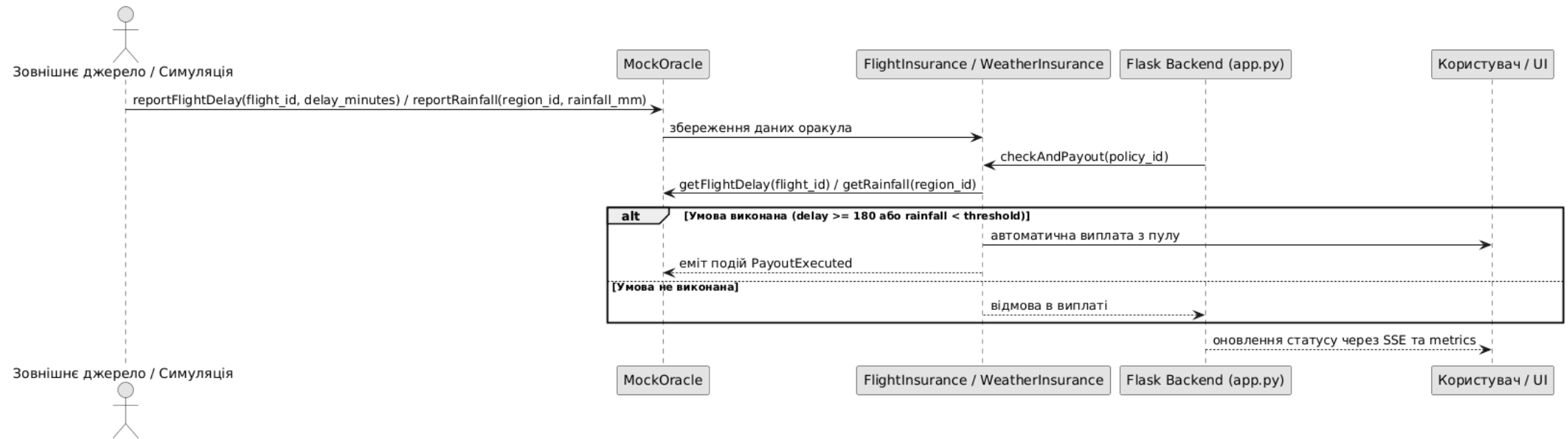
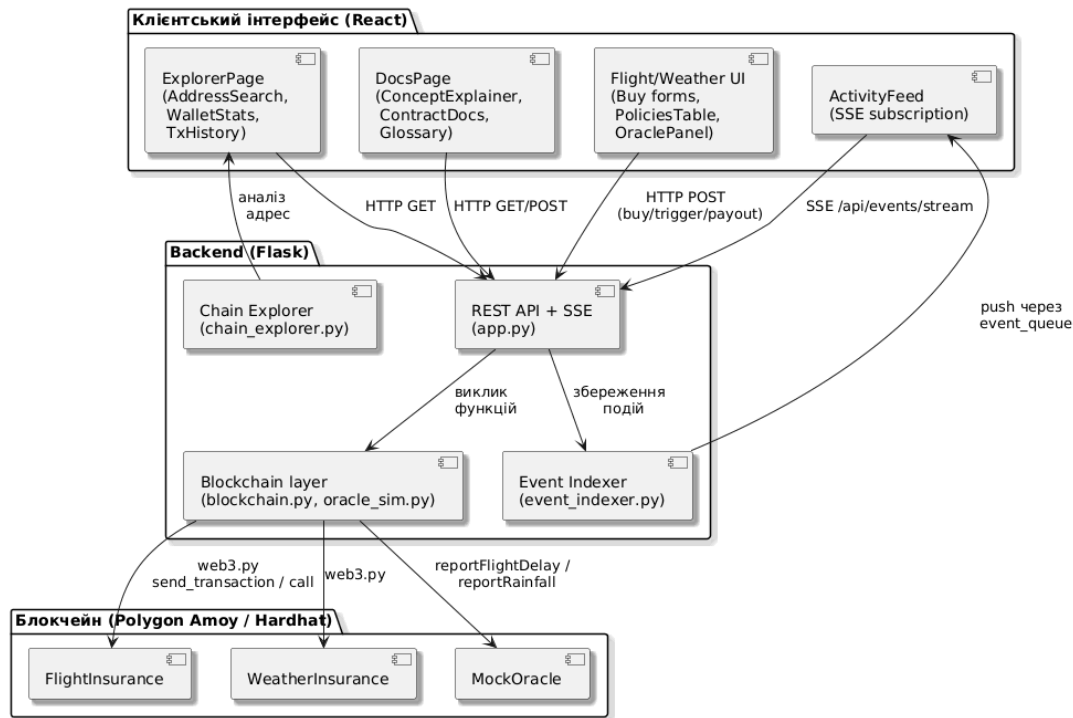


Рисунок 2.18 – Схема механізму верифікації страхового випадку через оракул у проєкті CLAIMLESS

Механізм роботи оракулів

- На рисунку 2.18 представлено механізм верифікації страхового випадку через оракул. MockOracle отримує дані про затримки рейсів та погодні умови. Смарт-контракти використовують ці дані для автоматичної перевірки умов страхування. Після підтвердження випадку система виконує виплату без участі оператора. Це забезпечує повну автоматизацію страхового процесу.



Контракт	Функція	Опис	Параметри
FlightInsurance	buyPolicy	Придбання поліса страхування рейсу	flight_id (string)
FlightInsurance	checkAndPayout	Перевірка умов і виконання виплати	policy_id (uint256)
FlightInsurance	getStats	Отримання статистики контракту	–
FlightInsurance	getPolicy	Отримання деталей конкретного поліса	policy_id (uint256)
WeatherInsurance	buyPolicy	Придбання поліса страхування від посухи	region_id, threshold_mm, duration_days, premium_matic
WeatherInsurance	checkAndPayout	Перевірка умов і виконання виплати	policy_id (uint256)
MockOracle	reportFlightDelay	Фіксація затримки рейсу оракулом	flight_id, delay_minutes
MockOracle	reportRainfall	Фіксація даних про опади	region_id, rainfall_mm

Реалізація функціоналу системи

- У таблиці 3.1 наведено основні функції страхових смарт-контрактів. Система підтримує функції buyPolicy, checkAndPayout та reportFlightDelay. На рисунку 3.2 показано інтеграцію клієнтського інтерфейсу зі смарт-контрактами. Користувач взаємодіє з blockchain-системою через Rabby Wallet. Система забезпечує автоматичне виконання страхових операцій.

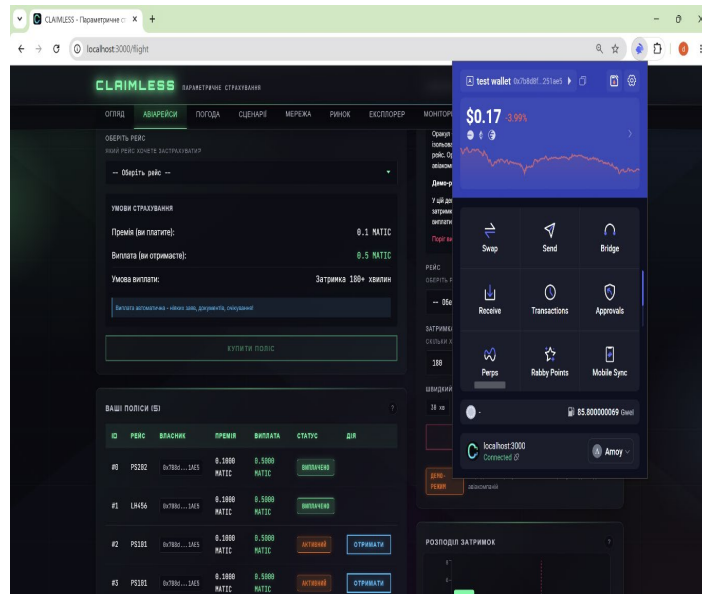


Рисунок 3.3 – Сторінка «Огляд системи»

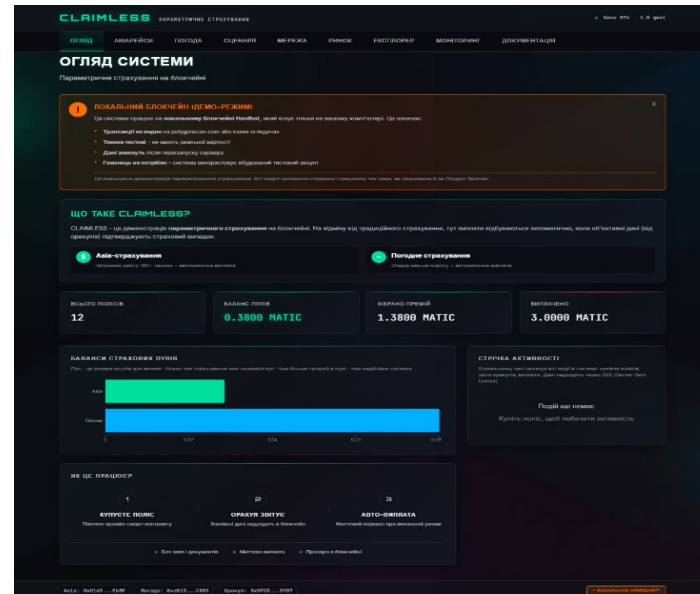


Рисунок 3.5 – Сторінка «Авіарейси» з панеллю RabbyWallet

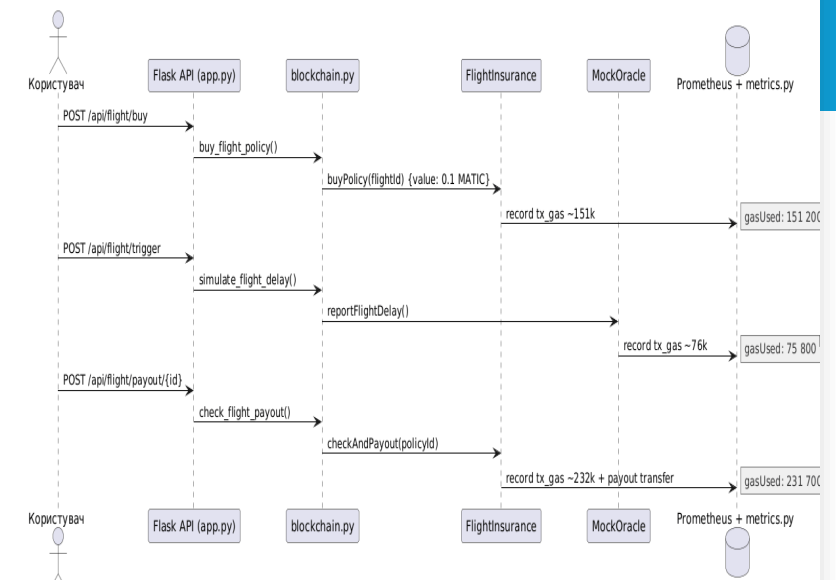


Рисунок 3.19 – Послідовність виконання операцій страхування рейсу з накопиченням газових витрат

Інтерфейс користувача та тестування системи

- На рисунку 3.3 представлено сторінку «Огляд системи». Рисунок 3.5 демонструє сторінку «Авіарейси» з інтегрованим RabbyWallet. Користувач може придбати страховий поліс безпосередньо через web-інтерфейс. На рисунку 3.19 показано накопичення gas-витрат під час страхових операцій. Тестування підтвердило низький рівень комісій у мережі Polygon.



Висновки

- У результаті роботи було створено систему страхових смарт-контрактів CLAIMLESS. Розроблене рішення забезпечує автоматизацію страхових виплат. Блокчейн-технології дозволили підвищити прозорість страхових процесів. Система характеризується низькими транзакційними витратами та масштабованістю. Отримані результати можуть бути використані для реальних blockchain-страхових сервісів.