МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Факультет інформатики
Кафедра мережних технологій

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему:

Платформа для децентралізованого кредитування з автоматизованими механізмами управління ризиками на базі смартконтрактів

Виконав:
студент 2 року навчання
освітнього ступеня «Магістр»
спеціальності 121 «Інженерія програмного забезпечення»
освітньо-наукової програми
«Інженерія програмного забезпечення»
Санін Михайло Олександрович

Науковий керівник:
Гороховський Семен Самуїлович,
кандидат наук, доцент

Рецензент:
Віровець Денис Володимирович,
доктор філософії з економіки, доцент

Київ 2026

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Факультет інформатики
Кафедра мережних технологій

ЗАТВЕРДЖУЮ
Завідувач кафедри мережних технологій
_____
(підпис)
«___» _____ 2026 р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на магістерську роботу

студенту 2-го курсу, факультету інформатики
Саніну Михайлу Олександровичу

Розробити платформу для децентралізованого кредитування з автоматизованими механізмами управління ризиками на базі смартконтрактів

Вихідні дані:
- нормативні вимоги НаУКМА до кваліфікаційної магістерської роботи;
- матеріали Aave, Compound, Morpho, Fluid, Chainlink і OpenZeppelin;
- специфікація ERC-20 і підходи до реалізації смартконтрактів у мережах, сумісних з EVM;
- інструментальні засоби Solidity, Foundry та засоби інтеграції децентралізованого застосунку.

Зміст ТЧ до магістерської роботи:
Зміст
Анотація
Перелік умовних позначень та скорочень
Вступ
1 Аналіз предметної області децентралізованого кредитування
2 Проєктування платформи та механізмів управління ризиками
3 Проєкт реалізації та перевірки платформи
Висновки
Список використаних джерел
Додатки

Строк подання роботи студентом 20 травня 2026 року

Дата видачі «08» жовтня 2025 р. Керівник _____
(підпис)
Завдання отримав _____
(підпис)

КАЛЕНДАРНИЙ ПЛАН

Тема: «Платформа для децентралізованого кредитування з автоматизованими механізмами управління ризиками на базі смартконтрактів»

Студент: Санін Михайло Олександрович
Науковий керівник: Гороховський Семен Самуїлович

№	Назва етапу роботи	Термін виконання	Примітка
1	Вибір теми, погодження напряму дослідження та формування індивідуального завдання.	жовтень 2025 року	Початок роботи.
2	Аналіз предметної області децентралізованого кредитування, збір і систематизація джерел, вивчення аналогів.	жовтень – листопад 2025 року	Огляд аналогів.
3	Формування структури магістерської роботи та уточнення функціональних і нефункціональних вимог до платформи.	листопад 2025 року	План і вимоги.
4	Проєктування архітектури платформи, ролей учасників і потоків взаємодії між компонентами.	грудень 2025 року – січень 2026 року	Архітектура системи.
5	Розроблення математичної моделі керування ризиками, правил добору заставних активів і параметрів кредитування.	лютий 2026 року	Модель ризику.
6	Написання першого і другого розділів магістерської роботи, узгодження результатів із науковим керівником.	лютий – березень 2026 року	Розділи 1–2.
7	Підготовка проєкту реалізації, опису перевірки платформи та супровідних матеріалів практичної частини.	березень – квітень 2026 року	Реалізація і тести.
8	Доопрацювання тексту магістерської роботи, оформлення результатів та підготовка до подання.	квітень – травень 2026 року	Оформлення.
9	Підготовка доповіді, ілюстративного матеріалу та завершальної редакції роботи.	травень 2026 року	Матеріали до захисту.

Студент Санін М. О. _____ Науковий керівник Гороховський С. С. _____

ЗМІСТ

АНОТАЦІЯ **5**

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ **6**

ВСТУП **10**

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ DeFi-КРЕДИТУВАННЯ **13**

1.1 Особливості заставного DeFi-кредитування 13

1.2 Класи ризиків у децентралізованому кредитному протоколі . . . 14

1.3 Огляд існуючих підходів і технологічних рішень 15

1.4 Порівняльний аналіз платформ і постановка задачі 17

1.5 Формування вимог до платформи 21

Висновки до розділу 1 23

2 ПРОЄКТУВАННЯ ПЛАТФОРМИ ТА УПРАВЛІННЯ РИЗИКАМИ **24**

2.1 Архітектура системи 24

2.2 Ролі учасників і сценарії взаємодії 26

2.3 Математична модель добору застави та оцінювання позиції . . . 28

2.3.1 Автоматичний добір застави та ефективні параметри ризику 29

2.4 Відсоткова модель, ліквідність і правила ліквідації 34

2.5 Технічні запобіжники та керування параметрами моделі 36

2.6 Вибір параметрів ризику для демонстраційного ринку 37

2.7 Числові сценарії поведінки позиції 40

Висновки до розділу 2 42

3 ПРОЄКТ РЕАЛІЗАЦІЇ ТА ПЕРЕВІРКИ ПЛАТФОРМИ **44**

3.1 Структура практичної реалізації 44

3.2 Модель зберігання стану 46

3.3 Алгоритми реалізації основних операцій 48

3.3.1 Добове оновлення **PairRegistry** 48

3.3.2 Відкриття позики з ланцюговим швидким блокуванням . 51

3.3.3 Погашення, виведення застави та ліквідація 52
3.4 Інтеграція з децентралізованим застосунком і безсерверний рівень 53
3.5 Організація тестування та перевірки 53
3.6 Стійкість реалізації до типових інцидентів 55
3.7 Критерії практичної придатності 56
Висновки до розділу 3 57

ВИСНОВКИ **58**

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ **59**

ДОДАТОК А. Лістинг структур стану та зовнішніх викликів **63**

ДОДАТОК Б. Лістинг добового пакета оновлення PairRegistry **65**

ДОДАТОК В. Лістинг сценарію перевірки швидкого блокування **66**

АНОТАЦІЯ

У роботі розв'язано задачу проєктування та реалізації платформи для децентралізованого заставного кредитування на базі смартконтрактів. Предмет дослідження охоплює моделі оцінювання забезпечення, правила нарахування відсотків, механізми ліквідації позицій, використання цінових оракулів і технічні засоби захисту протоколу. Метою роботи є проєктування та реалізація програмної системи, яка забезпечує автоматизоване керування кредитним і ліквіднісним ризиком, автоматичний добір безпечних заставних активів та коректну перевірку цінових даних без централізованого посередника.

У першому розділі подано аналіз предметної області децентралізованого кредитування, виокремлено основні групи ризиків і виконано порівняння сучасних протоколів та допоміжних технологій. У другому розділі запропоновано архітектуру платформи, формалізовано правило автоматичного добору допустимих заставних активів для заданого активу позики, ефективний LTV, ефективний ліквідаційний поріг, правила перевірки актуальності ціни, відсоткову модель і процедуру ліквідації. У третьому розділі викладено практичну реалізацію платформи: структуру ланцюгового ядра, модель зберігання стану, алгоритми критичних операцій, добовий контур переоцінювання пар, інтеграцію безсерверного децентралізованого застосунку та програму перевірки.

Практична цінність роботи полягає в тому, що запропоновані рішення можна використати як основу для реалізації DeFi-протоколу з прозорими параметрами ризику, відтворюваними перевірками та придатною до аудиту структурою контрактів. Практичний результат охоплює архітектуру смартконтрактів, орієнтовану на окремі пари активів, окремий реєстр допустимих пар, добовий контур оновлення меж ризику для конкретної пари та набір сценаріїв перевірки критичних режимів роботи.

КЛЮЧОВІ СЛОВА:
DEFI, ДЕЦЕНТРАЛІЗОВАНЕ КРЕДИТУВАННЯ, СМАРТКОНТРАКТ, ЗАСТАВНА ПОЗИЦІЯ, ЛІКВІДАЦІЯ, УПРАВЛІННЯ РИЗИКАМИ, ДЕЦЕНТРАЛІЗОВАНИЙ ЗАСТОСУНОК.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

- **DeFi** — децентралізовані фінанси.
- **dApp** — децентралізований застосунок, що взаємодіє зі смартконтрактами.
- **EVM** — віртуальна машина Ethereum.
- **ERC-20** — стандарт взаємозамінних токенів у мережах, сумісних з EVM.
- **LTV** — відношення суми позики до вартості забезпечення.
- **LT** — ліквідаційний поріг заставної позиції.
- **LLTV** — граничне відношення суми позики до вартості застави для конкретного ринку пари активів у Morpho, при перевищенні якого позиція стає доступною для ліквідації.
- **HF** — показник платоспроможності позиції (*Health Factor*).
- **APY** — річна дохідність з урахуванням капіталізації.
- **APR** — річна ставка без урахування капіталізації.
- **RPC** — інтерфейс віддаленого виклику процедур для взаємодії децентралізованого застосунку з вузлом мережі.
- **ERC-4626** — стандарт токенізованих сховищ активів у мережах, сумісних з EVM.
- **IRM** — модель відсоткової ставки (*Interest Rate Model*).
- **Oracle** — оракул, тобто компонент, що передає зовнішні дані в блокчейн-середовище.
- **Ціновий фід** — публічний потік цінових даних, який протокол використовує як зовнішнє джерело ринкової ціни.
- **bps** — базисні пункти, тобто сота частка відсотка, яку використано для цілочисельного подання меж ризику та ставок.
- **ray** — формат фіксованої коми з масштабом 10^{27}, який використано для індексів і відношень у ланцюгових розрахунках.

- **Мультипідпис** — механізм підтвердження адміністративних дій кількома уповноваженими ключами.
- **RISK_ADMIN** — роль, уповноважена подавати пакетне оновлення параметрів допустимих пар.
- **PARAM_ADMIN** — роль, уповноважена змінювати базові пороги моделі ризику та системні параметри.
- batchUpdatePairs — пакетний виклик оновлення параметрів допустимих пар у **PairRegistry**.
- **Resolver** — допоміжний контракт або інтерфейс читання агрегованого стану системи.
- **Швидке блокування** — ланцюговий механізм тимчасового блокування ризикових дій.
- **Коефіцієнт закриття** — частка боргу, яку дозволено погасити в одній операції ліквідації (*close factor*).
- **Ліквідаційний поріг** — рівень забезпечення, нижче якого позиція переходить у стан ліквідації.
- **Ліквідаційний бонус** — додаткова частка застави, яку отримує ліквідатор.
- **Резервний коефіцієнт** — частка відсоткового доходу, що резервується протоколом.
- **aToken** — токенізоване подання депозитної позиції в Aave.
- **cToken** — токенізоване подання депозитної позиції в Compound v2.
- **Comet** — ринок Compound III з одним базовим активом позики.
- $\mathbf{Q_c}$ — кількість заставного активу в позиції.
- $\mathbf{B_d}$ — обсяг боргу в активі позики.
- $\mathbf{P_c}$, $\mathbf{P_d}$ — ціни заставного активу та активу позики.
- $\mathbf{LTV_c^{base}}$, $\mathbf{LT_c^{base}}$ — базові межі запозичення і ліквідації для заставного активу.
- Δ_c — базовий буфер між LTV_c^{base} і LT_c^{base}.

- $\rho_{cd}(W)$ — коефіцієнт узгодженості цінової динаміки пари «застава – актив позики».
- $\mathbf{q}_{\mathbf{cd}}^{(\alpha)}(\mathbf{W})$ — нижній квантиль відносної дохідності пари.
- $\sigma_{cd}^{rel}(W)$ — відносна волатильність пари «застава – актив позики», обчислена за рядом $z_{cd}^{(k)}$.
- $\mathcal{D}^{stb}$ — множина низьковолатильних активів позики, для яких кореляція доходностей не використовується як основний критерій добору застави.
- $\sigma_{\max}^{rel}$ — максимально допустима відносна волатильність для пари з низьковолатильним активом позики.
- $\mathbf{a_{cd}}$ — коефіцієнт сумісності пари активів.
- ϕ_{cd} — коефіцієнт зниження базових параметрів ризику для конкретної пари.
- $\mathbf{U_t}$ — рівень використання пулу ліквідності в момент часу t.
- $\mathbf{g_U}$, $\mathbf{g_n}$ — швидкість зростання коефіцієнта використання ліквідності та її нормована форма.
- s_ρ, s_σ, s_q — нормовані складові узгодженості цінової динаміки, відносної волатильності та допустимого відносного просідання.
- U_k — точка перегину відсоткової кривої.
- g_{crit} — критична швидкість зростання використання ліквідності для нормування g_U.
- r_b, r_s — позикова ставка та ставка постачальника ліквідності.
- rf — резервний коефіцієнт.
- cf — коефіцієнт закриття однієї ліквідації.
- $bonus$ — ліквідаційний бонус.
- R_{cd}^{ref}, R_{cd}^{now} — опорне та поточне відношення ціни застави до ціни активу позики.
- γ_{cd} — поріг швидкого блокування для пари активів.
- $\gamma_{\min}$, $\gamma_{\max}$ — нижня та верхня межі порогу швидкого блокування.
- λ_γ — коефіцієнт, який перетворює історичне відносне просідання пари на поріг швидкого блокування.

- τ_{stale} — максимально допустимий вік відповіді цінового фіду.

ВСТУП

Децентралізоване кредитування стало одним із базових сценаріїв використання блокчейн-мережі Ethereum та сумісних з нею платформ. У цій моделі користувачі передають активи до пулу ліквідності, отримують дохід від розміщення коштів або залучають позику під заставу без участі банку, брокера чи іншого централізованого посередника. Смартконтракт бере на себе облік часток, перевірку лімітів, нарахування відсотків і виконання правил ліквідації. Перевага такого підходу полягає в прозорості логіки, відкритому доступі до стану системи та автоматичному виконанні закладених умов. Разом із цим зростає вимога до точності моделей ризику, бо помилка в параметрах забезпечення, джерелі ціни або реалізації контракту безпосередньо впливає на збереження коштів користувачів.

DeFi-протоколи працюють у середовищі високої цінової волатильності, нерівномірної ліквідності та постійного технічного тиску на смартконтракти. Заставна модель кредитування не усуває ризик повністю, а переводить його в площину кількісного контролю. Протокол повинен оцінювати поточну вартість застави, обмежувати надмірний борг, реагувати на дефіцит ліквідності, перевіряти придатність цінових даних і містити механізм аварійного зупинення окремих функцій. Ці вимоги визначають практичну складність побудови децентралізованої кредитної платформи.

Мета роботи полягає у проєктуванні та реалізації програмної платформи для децентралізованого кредитування з автоматизованими механізмами управління ризиками на базі смартконтрактів.

Для досягнення поставленої мети в роботі розв’язано такі **завдання**:

1. проаналізувати предметну область заставного децентралізованого кредитування та практики, які застосовують сучасні DeFi-протоколи;
2. виокремити кредитний і ліквіднісний ризики, ризик некоректно підібраного заставного активу та вимоги до коректності цінових даних, які визначають стійкість кредитної платформи;
3. формалізувати автоматичний добір допустимих заставних активів для заданого активу позики та виведення параметрів ризику для конкретної пари активів;

4. побудувати модель перевірки актуальності ціни, відсоткову модель і процедуру ліквідації для платформи із заставним кредитуванням;
5. сформувати функціональні та нефункціональні вимоги до системи;
6. спроєктувати архітектуру платформи, її модулі, ролі та потоки даних;
7. реалізувати смартконтракти платформи, інтегрувати децентралізований застосунок і перевірити критичні сценарії функціонування платформи.

Об'єктом дослідження є процеси заставного кредитування в децентралізованих фінансових протоколах.

Предметом дослідження є методи та програмні засоби автоматизованого управління ризиками в системі децентралізованого кредитування на базі смартконтрактів.

Методи дослідження охоплюють системний аналіз предметної області, порівняльний аналіз існуючих протоколів, оцінювання відносної цінової динаміки пари «застава – актив позики», формалізацію параметрів ризику, математичне моделювання показників позиції та архітектурне проєктування програмної системи.

Наукова новизна роботи полягає в удосконаленні методу автоматичного добору допустимих заставних активів і виведення меж кредитного ризику для конкретної пари активів у децентралізованій кредитній платформі. На відміну від розглянутих конфігураційних підходів, у яких допустимі комбінації активів і межі ризику задаються наперед на рівні активу, категорії або вже створеного ринку, запропоновано формальне правило автоматичного добору допустимих заставних активів для заданого активу позики. Для звичайних пар воно спирається на узгодженість цінової динаміки активів і нижній квантиль відносної дохідності пари, а для низьковолатильного активу позики використовує окреме правило на основі нижнього квантиля відносної дохідності та обмеження відносної волатильності пари. На цій основі автоматично виводяться ключові межі кредитного ризику для конкретної пари активів, насамперед ефективний LTV та ефективний ліквідаційний поріг. Ціна ліквідації розраховується як похідна величина для конкретної позиції. Додатково запропоновано модифікацію відсоткової моделі, яка враховує швидкість зростання використання пулу ліквідності.

Практичне значення роботи полягає в розробленні протоколу, у якому

для кожного активу позики формально визначається допустимий набір заставних активів, а для низьковолатильного активу позики застосовується окреме правило добору без покладання на формально слабку кореляцію доходностей. Для кожної допустимої пари автоматично обчислюються похідні межі ризику. Це робить логіку керування ризиками відтворюваною, спрощує аудит правил допуску нових ринків і знижує ймовірність відкриття позицій із завідомо слабким профілем стійкості.

Робота складається зі вступу, трьох розділів, висновків, списку використаних джерел і трьох додатків. У першому розділі розглянуто предметну область та вимоги до системи. У другому розділі подано модель ризиків і архітектуру платформи. У третьому розділі викладено практичну реалізацію та підхід до перевірки ключових сценаріїв.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ ДЕЦЕНТРАЛІЗОВАНОГО КРЕДИТУВАННЯ

1.1 Особливості заставного DeFi-кредитування

Децентралізоване кредитування спирається на пул ліквідності або набір ізольованих ринків, у межах яких одні учасники надають активи, а інші залучають позику під заставу. На відміну від класичного кредитування, тут рішення ухвалює не оператор системи, а смартконтракт. Код визначає, які активи допустимі, який обсяг позики можна відкрити, коли позиція втрачає стійкість і за яких умов запускається ліквідація.

У такій системі важливе не лише саме зіставлення застави й боргу, а й спосіб подання позиції користувача. Постачальник ліквідності може отримувати окремий ринковий токен, що представляє частку в пулі та накопичує дохід, або працювати через внутрішні частки чи записи стану. Для архітектури протоколу ця модель визначає складність контрактів, механізм нарахування доходу, можливість передавання прав на позицію та обсяг поверхні для аудиту.

Ключові параметри заставного кредитування мають бути задані явно. До них належать річна дохідність постачальника і позичальника, початковий ліміт запозичення LTV, ліквідаційний поріг, ліквідаційний бонус або штраф, показник стійкості позиції, а також ліквідаційна ціна. Ці величини визначають, наскільки капіталоефективною буде платформа і який запас міцності вона зберігатиме за несприятливого руху ринку.

Для багаторинкової системи окремого значення набуває вибір пари «застава – актив позики». Якщо заставний актив може швидко дешевшати відносно активу позики, позиція втрачає стійкість навіть тоді, коли в момент відкриття позики базові параметри ринку виглядали прийнятними. Аналіз кредитної платформи має враховувати не лише поточну надлишковість забезпечення, а й поведінку відносної ціни пари, величину буфера до ліквідації та здатність відсоткової моделі реагувати на стрес ліквідності.

1.2 Класи ризиків у децентралізованому кредитному протоколі

Автоматично керовані ризики кредитних операцій і технічні обмеження безпеки реалізації належать до різних рівнів моделі. Таке розмежування відділяє фінансову модель від засобів її безпечного втілення.

Першу групу становить **кредитний ризик позиції**. Він виникає тоді, коли приведена вартість застави вже не покриває суму боргу разом із накопиченими відсотками. Якщо протокол продовжує дозволяти збільшення боргу або зменшення застави в такому стані, він починає накопичувати збиток. Цей ризик контролюють параметри LTV, ліквідаційний поріг, ліквідаційна ціна та правило переходу позиції в стан ліквідації.

Друга група стосується **коректності цінових даних**. Протокол не обчислює ринкову ціну самостійно, а спирається на зовнішні цінові потоки даних (цінові фіди). Тому прострочені або некоректні відповіді призводять до хибної оцінки застави та боргу. Для кредитної платформи коректність цінових даних є обов'язковою передумовою: перед будь-яким розрахунком система повинна перевіряти допустимий вік відповіді, джерело ціни та формат подання.

Третя група — **ризик ліквідності**. Він проявляється тоді, коли частка вже виданих позик стає надто великою, а вільний резерв протоколу скорочується настільки, що постачальники ліквідності втрачають можливість вийти з ринку без затримки. Для цього випадку недостатньо одного статичного ліміту. Потрібна ставка, яка реагує на поточне використання пулу та на швидкість погіршення ліквідності.

Четверта група — **ризик некоректно підібраного заставного активу**. Якщо актив застави для конкретного активу позики має нестійку спільну динаміку або демонструє глибокі короткі падіння у відносній ціні, позиція може швидко перейти до ліквідації навіть без загального стресу ринку. Для багатопарної платформи цей ризик визначає не лише значення LTV і ліквідаційного порогу, а й сам допустимий набір заставних активів для кожного активу позики.

Технічні вимоги безпеки реалізації лежать поза межами автоматично керованого ринкового ризику. Помилки доступу, повторного входу, неконтрольовані зовнішні виклики або надмірні адміністративні повноваження не належать до фінансової моделі, але визначають, чи може протокол безпечно її втілити. Далі технічні запобіжники описуються як інженерні обмеження платформи, а не як окремий клас ризику для математичного керування.

Керування ризиком у децентралізованому кредитуванні має охоплювати достатність забезпечення, вибір безпечної пари «застава – актив позики» та динаміку ліквідності, а коректність цінових даних слугує необхідною передумовою для цих рішень. Поряд із цим архітектура повинна враховувати технічні вимоги безпеки реалізації.

1.3 Огляд існуючих підходів і технологічних рішень

Порівняння платформ у цій роботі спирається на характеристики, які безпосередньо визначають межі ризику і капіталоефективність: спосіб організації ринків, форму подання депозитної позиції, правило визначення межі запозичення, буфер до ліквідації, механізм ліквідації, спосіб добору допустимої застави та поведінку відсоткової ставки за напруження ліквідності. Ці ознаки зведено в табл. 1.1; далі їх послідовно розглянуто для кожного аналога.

Aave розвивається від моделі спільного пулу до дедалі детальнішого сегментування ризику. У V1 і V2 користувач постачає актив до спільного пулу ліквідності та отримує *aToken*, тобто токен, який відображає його частку в депозиті та накопичений дохід. Межа боргу визначається не конкретною парою активів, а параметрами заставного активу, а стійкість позиції контролюється через *Health Factor*, тобто показник платоспроможності позиції, що порівнює приведену вартість застави з урахуванням ліквідаційного порогу з поточним боргом [1, 2]. У V3 з'являються два важливі режимні механізми. *E-mode* підвищує *LTV* і ліквідаційний поріг для категорій активів зі схожою поведінкою, наприклад для різних стейблкоїнів або похідних від ETH [3]. *Isolation Mode*, навпаки, вводить обмеження для ризиковішої застави: такий актив можна використовувати лише проти заздалегідь дозволеного боргу, а обсяг запозичення обмежується стелею боргу (*debt ceiling*) [4]. Конфігураційна модель Aave V3 є ключовою для порівняння, оскільки поєднує спільний пул ліквідності з режимним керуванням ризиком на рівні категорій активів і окремих заставних активів.

Compound проходить іншу еволюцію. Whitepaper і Compound v2 закріплюють класичну схему *cToken+Comptroller*. *cToken* є токеном депозитної позиції, а *Comptroller* — центральним контрактом, який обчислює межу запозичення через *collateral factor* і показник *account liquidity*, тобто запас до перевищення ліміту [6, 7, 8]. Compound III переходить до моделі *Comet*, де кожен ринок має один базовий актив позики, а решта активів можуть виступати заставою

для цього боргу [10]. Тут уже чітко розділено два коефіцієнти: *borrow collateral factor* обмежує початковий борг, а *liquidation collateral factor* визначає межу ліквідації [11]. Це створює явний буфер між відкриттям позиції та її ліквідацією. В обох лінійках ставка залежить від коефіцієнта використання ліквідності (*utilization*) і змінюється за кривою з точкою перегину [9, 12]. Compound чітко формалізує розрахунок буфера до ліквідації, але перелік заставних активів і тут задається адміністративно.

Morpho має дві різні за змістом лінії. *Morpho Optimizers* працювали як надбудова над чинними пулами Aave і Compound: протокол намагався звести постачальників і позичальників напряму, а якщо це не вдавалося, використовував базовий пул [14, 15]. Головний ефект *Optimizers* полягав у покращенні ставки, а не в заміні моделі ризику базового протоколу. Сучасний *Morpho* будує ізольований ринок одразу навколо пари активів. Кожний ринок задається п'ятьма параметрами: заставний актив, актив позики, *LLTV* (*Liquidation Loan-To-Value*, тобто межа боргу до ліквідації), оракул як джерело ціни та *IRM* (*Interest Rate Model*, тобто модель ставки) [16, 17]. Поверх таких ринків працюють *Morpho Vaults / MetaMorpho*, де куратор (*curator*) дозволяє або забороняє окремі ринки та керує межами розміщення ліквідності [18, 19]. Morpho робить пару активів явною одиницею ризику, але створення ринку та вибір його параметрів залишаються рішенням ініціатора ринку (*creator*) або куратора.

Fluid побудований як модульна контрактна система з єдиним шаром ліквідності (*Liquidity Layer*) у центрі, над яким працюють протоколи *Lending*, *Vault* і *DEX* [21, 22, 23]. Офіційна технічна документація прямо вказує, що базовий контракт *Liquidity* зберігає кошти і взаємодіє не з кінцевими користувачами, а з протоколами верхнього рівня [22].

Доступ до цього шару для постачальника ліквідності реалізовано через ERC-4626-сумісні *fToken*, а нові *fToken* створюються через *LendingFactory* [22]. Позика відкривається через *Vault Protocol*, де кожне сховище має одну заставу і один актив позики; нові сховища створюються через *VaultFactory* [21, 22]. У DEX V2 документація окремо описує режими *smart collateral* і *smart debt* [24]. Для читання стану системи винесено окремі *resolver*-контракти для ліквідності, *fToken*, сховищ, ліквідацій і DEX-резервів [21]. Fluid розглядається як приклад модульної архітектури. Водночас з доступних публічних матеріалів не випливає універсальне автоматичне правило добору застави для активу позики:

конфігурація нових модулів і їх параметрів задається явно.

Chainlink і **OpenZeppelin** не є прямими аналогами кредитної платформи, але визначають дві критичні технологічні залежності. Chainlink задає практичний стандарт зовнішніх цінових потоків даних і вимагає контролю свіжості відповіді. OpenZeppelin задає стандарт безпечних шаблонів для доступу, паузи й захисту від повторного входу [25, 26, 27].

1.4 Порівняльний аналіз платформ і постановка задачі удосконалення

У табл. 1.1 зведено Aave, Compound, Morpho та Fluid за параметрами, які прямо пов’язані із запропонованим удосконаленням: одиницею ризику, первинною межею боргу, межею ліквідації, способом появи нової комбінації активів і наявністю або відсутністю формального правила добору застави для заданого активу позики. Для Aave V3, Compound III та Morpho Blue ці характеристики додатково звірено з відкритими контрактними джерелами ReserveConfigurati on.sol, CometConfiguration.sol і IMorpho.sol. Для Fluid враховано лише ті архітектурні ознаки, які прямо описані в офіційній документації, файлі docs.md та відкритому репозиторії контрактів [5, 13, 20, 21, 22, 23].

Табл. 1.1: Порівняння платформ за параметрами, релевантними до автоматичного добору безпечної заставної пари

Параметр	**Aave**	**Compound**	**Morpho**	**Fluid**
Розглянуті версії / лінійки	V1, V2, V3	Whitepaper / v2, III	Optimizers, Markets, Vaults	Liquidity, Lending, Vaults, DEX V2
Організація ринку	Спільні пули ліквідності	v2 — cToken-ринки; III — Comet з одним базовим активом	Ізольований ринок для явної пари активів	Liquidity Layer + окремі протоколи Lending, Vault і DEX
Одиниця ризику	Актив / E-mode-категорія	v2 — ринок; III — базовий ринок + коефіцієнти застави	Пара застава/борг + $LLTV$ + оракул + IRM	Окремий модуль верхнього рівня; для позики — конкретне сховище
Первинна межа боргу	LTV	v2: *collateral factor* III: *borrow collateral factor*	$LLTV$ ринку	Задається конфігурацією конкретного сховища
Параметр ліквідації	LT + *Health Factor*	v2 — *account liquidity*; III — *liquidation collateral factor*	Межа задається $LLTV$	Визначається параметрами і ліквідаційною логікою конкретного сховища
Буфер між відкриттям і ліквідацією	Так: $LTV < LT$	Так; у III: *borrow collateral factor* < *liquidation collateral factor*	Окремого універсального буфера немає	Визначається конфігурацією конкретного сховища

Табл. 1.1: Порівняння платформ за параметрами, релевантними до автоматичного добору безпечної заставної пари (продовження)

Параметр	**Aave**	**Compound**	**Morpho**	**Fluid**
Оракул / ціна	Зовнішні цінові фіди	*Price oracle* / цінові фіди активів	Оракул є частиною ринку	Ціна для *Vault Protocol* береться через окремий оракул; читання даних винесено в *resolver*-контракти
Відсоткова модель	Ставка залежить від коефіцієнта використання ліквідності	Крива з точкою перегину від коефіцієнта використання ліквідності	IRM задається при створенні ринку	Офіційні матеріали описують окремі ставки *lend*, *borrow* і *yield* для різних модулів
Як з'являється нова комбінація активів	Лістинг активу і режиму через керування протоколом (*governance*)	Додавання або зміна активу через керування протоколом (*governance*)	Ініціатор ринку задає пару, $LLTV$, оракул і IRM; куратор керує сховищем	Через *LendingFactory*, *VaultFactory*, *DexFactory* та явну конфігурацію модулів
Врахування спорідненості активів	Частково: через E-mode	Як окреме правило не виділено	Автоматичного правила немає; ризик задають параметри ринку	Як окреме правило в офіційних матеріалах не виділено
Автоматичний добір безпечної застави для заданого активу позики	Не описано як універсальне правило; перелік задає керування протоколом	Не описано як універсальне правило; перелік задає керування протоколом	Не описано як універсальне правило; пару задає ініціатор ринку / куратор	Не описано як універсальне правило; конфігурацію задає модуль / фабрика

Табл. 1.1: Порівняння платформ за параметрами, релевантними до автоматичного добору безпечної заставної пари (продовження)

Параметр	**Aave**	**Compound**	**Morpho**	**Fluid**
Автоматичне виведення меж LTV і LT для конкретної пари з похідним розрахунком ціни ліквідації	Не описано як універсальне правило; параметри задаються наперед	Не описано як універсальне правило; параметри задаються наперед	Не описано як універсальне правило; $LLTV$ задається під час створення ринку	Не описано як універсальне правило; межі задає конфігурація сховища
Елемент, використаний для формування вимог	E-mode, Isolation Mode і сегментація ризику	Явний буфер між відкриттям і ліквідацією у III	Пара активів як явна одиниця ризику	Поєднання шару ліквідності з вузько налаштованими сховищами

Усі розглянуті платформи вже мають робочі механізми контролю забезпечення, ліквідації, ставки та ціни. Для цільової платформи найбільше значення мають Aave V3, Compound III та Morpho Blue: Aave задає режимні категорії активів, Compound III фіксує явний буфер між відкриттям і ліквідацією, а Morpho робить пару активів окремою одиницею ризику. Fluid у цьому розділі є додатковим архітектурним орієнтиром і показує, як вузько спеціалізовані модулі можуть працювати над спільним шаром ліквідності. Водночас поява нових ринків або підтримуваних комбінацій активів у всіх розглянутих підходах лишається наслідком явної конфігурації через керування протоколом, творця ринку, куратора або фабрику розгортання.

Відкриті описи платформ і доступна контрактна поверхня містять корисні часткові механізми сегментації ризику, зокрема категорії активів у Aave та явну пару як одиницю ринку в Morpho, але не задають загального публічно описаного автоматичного правила, яке б для заданого активу позики формувало безпечний набір заставних активів, відсікало пари з глибокими короткими просіданнями відносної ціни застави до активу позики і визначало межі запозичення та ліквідації для конкретної пари активів. Типові моделі ставки також реагують на поточне навантаження пулу, але не виділяють швидкість його погіршення як окремий керуючий фактор.

Запропоноване удосконалення полягає в автоматизованому керуванні ризиком у межах заставного кредитування. Його основу становлять автоматичний добір допустимих заставних активів для заданого активу позики та автоматичне виведення ключових меж кредитного ризику для конкретної пари активів, насамперед ефективного LTV і ефективного ліквідаційного порогу, з подальшим розрахунком ціни ліквідації для конкретної позиції. Додатково метод містить модифікацію відсоткової моделі з урахуванням швидкості зростання коефіцієнта використання ліквідності.

1.5 Формування вимог до платформи

Огляд аналогів дає змогу відокремити напрями удосконалення від базових технічних вимог. До напрямів удосконалення віднесено лише ті рішення, які безпосередньо усувають виявлені обмеження аналогів і становлять зміст запропонованого методу. У табл. 1.2 ці напрями подано разом із вимогами до платформи та обмеженнями аналогів, які вони усувають.

Табл. 1.2: Напрями удосконалення платформи, сформовані за результатами огляду аналогів

Напрям удосконалення	Вимога до платформи	Яке обмеження аналогів усувається
Автоматичний добір застави	Для кожного активу позики має автоматично формуватися допустимий набір заставних активів на основі формального критерію безпечності пари.	У розглянутих протоколах підтримувані пари визначаються конфігураційно через керування протоколом, ініціатора ринку, куратора або конфігуратор сховища.
Межі ризику для конкретної пари	Для кожної допустимої пари повинні автоматично виводитися ефективний LTV та ефективний ліквідаційний поріг; ціна ліквідації має розраховуватися як похідна величина для конкретної позиції.	Межі ризику в аналогах здебільшого задаються на рівні активу, категорії або наперед створеного ринку, а не обчислюються універсальним правилом для конкретної пари.
Контроль відносної цінової динаміки	Рішення про допустимість пари має враховувати не лише узгодженість динаміки активів, а й глибину коротких просідань відносної ціни застави до активу позики.	Відкриті описи платформ не задають загального правила відсікання застави, яка може швидко знецінитися саме відносно активу позики.
Адаптивне керування ліквіднісним ризиком	Відсоткова ставка має залежати не лише від поточного коефіцієнта використання ліквідності, а й від швидкості його зростання.	Типові моделі ставки, залежні від рівня використання ліквідності, реагують на вже досягнутий рівень навантаження, але слабше відображають швидке входження ринку в стресовий стан.

Окремо від цих напрямів удосконалення платформа повинна задовольняти базові технічні вимоги: перевіряти актуальність цінових даних, працювати з прозорим джерелом ціни, забезпечувати відтворюваність формул у коді й тестах та підтримувати архітектуру браузерного децентралізованого застосунку без власного бекенду. Такі вимоги важливі для практичної реалізації, але вони не становлять самостійної новизни, оскільки випливають із загальних правил побудови DeFi-протоколів.

Сформульовані напрями удосконалення охоплюють три пов’язані контури: відбір безпечної заставної пари, виведення меж кредитного ризику для конкретної пари активів та адаптивну реакцію на погіршення ліквідності. На цій основі в наступному розділі подано математичну модель добору застави, виведення параметрів ризику для конкретної пари активів та архітектуру платформи.

Висновки до розділу 1

Децентралізоване кредитування базується на надлишковому забезпеченні, але його стійкість визначають не тільки базові ліміти LTV і ліквідації. Основою порівняльних висновків цього розділу стали Aave V3, Compound III та Morpho Blue, оскільки саме для них у роботі найповніше простежуються і офіційні описи, і ключові конфігураційні параметри. Fluid використано як додатковий архітектурний орієнтир. Загальний аналіз показав, що сучасні платформи по-різному організовують ринки, подають позицію постачальника ліквідності, задають початкову межу запозичення і будують ліквідацію, проте для всіх них критичними залишаються питання добору пари активів, коректної роботи з ціновими фідами та реакції на напруження ліквідності.

Порівняння показало, що ключовим напрямом удосконалення є перехід від статичних параметрів окремого активу або категорії до формального правила автоматичного добору застави для конкретного активу позики. Це дає змогу автоматично визначати ключові межі кредитного ризику для конкретної пари, насамперед ефективний LTV та ефективний ліквідаційний поріг, а також похідну ціну ліквідації і модель ставки з поправкою на швидкість вичерпання ліквідності. На цій основі сформовано вимоги до цільової платформи.

РОЗДІЛ 2

ПРОЄКТУВАННЯ ПЛАТФОРМИ ТА МЕХАНІЗМІВ УПРАВЛІННЯ РИЗИКАМИ

2.1 Архітектура системи

Архітектура платформи повинна розділяти два контури: транзакційний і контур періодичного переоцінювання пар активів. Транзакційний контур обробляє депозит, позику, погашення, виведення застави та ліквідацію. Контур переоцінювання не стоїть на шляху користувацької транзакції: він на основі індексованих історичних даних цінових фідів періодично переобчислює показники пари «застава – актив позики», автоматично формує допустимий набір застави для кожного активу позики і виводить межі ризику для кожної допустимої пари. Прикладний рівень при цьому залишається безсерверним: браузерний децентралізований застосунок працює через гаманець користувача і публічний RPC без власного бекенду, а протокол не вимагає розгортання власних оракулів чи окремого сервісу, який стоїть на шляху транзакції.

Базовий модуль **LiquidityPool** відповідає за приймання активів, облік часток постачальників ліквідності та операції введення й виведення коштів. Він не повинен самостійно оцінювати ризик позиції, а лише виконувати зміну стану після зовнішньої перевірки. Це спрощує аудит і зменшує кількість прихованих залежностей.

Модуль **DebtEngine** обліковує суму боргу, накопичені відсотки й параметри поточної ставки. Його завдання полягає в тому, щоб приводити боргову позицію до актуального стану перед будь-якою критичною дією. Якщо ця логіка буде розпорошена по кількох контрактах, система швидко втратить узгодженість між поточним боргом і доступним лімітом.

Модуль **PairRegistry** зберігає для кожного активу позики перелік автоматично допустимих заставних активів і пов’язані з ними коефіцієнти пари та похідні межі ризику. У ньому матеріалізується результат періодичного переоцінювання: якщо пара не проходить формальний критерій безпечності, вона не потрапляє до допустимого набору для нового боргу, а якщо проходить, то для неї фіксуються автоматично обчислені параметри пари.

Модуль **RiskEngine** є центральним елементом архітектури. Він читає ціну з публічних децентралізованих фідів, звертається до **PairRegistry**, застосовує базові коефіцієнти ризику та коефіцієнти для конкретної пари, обчислює ефективний LTV, ефективний ліквідаційний поріг, ціну ліквідації, показник платоспроможності, перевіряє швидке блокування за відносною ціною $\frac{P_c}{P_d}$ і повертає висновок про допустимість операції. Розміщення цієї логіки в окремому модулі дає змогу централізовано тестувати правила, не торкаючись механізмів переказу активів.

Архітектура не передбачає власного контракту-оракула. Протокол використовує вже розгорнуті публічні фіди **Chainlink Data Feeds**, а в конфігурації ринку зберігає лише адреси дозволених фідів і параметр τ_{stale}. Історичний ряд цін, потрібний для переоцінювання сумісності пари, формується позаланцюгово з індексованих відповідей цих фідів і періодично переноситься до **PairRegistry** як набір коефіцієнтів пари. Цей процес не входить до транзакційного контуру видачі позики і не потребує постійно увімкненого бекенду.

Модуль **LiquidationManager** виконує часткове погашення боргу та вилучення відповідної частини застави. Він не приймає рішення про придатність позиції до ліквідації самостійно, а спирається на результат **RiskEngine**. Модуль **Configurator** відповідає за керування базовими параметрами ризику, мінімальним порогом кореляції $\rho_{\min}$, максимально допустимою відносною волатильністю $\sigma^{rel}_{\max}$ для ринків з $d \in \mathcal{D}^{stb}$, допустимою глибиною короткого відносного падіння $\delta_{\max}$, мінімальним коефіцієнтом зниження параметрів пари $\kappa_{\min}$, межами швидкого блокування $\gamma_{\min}$, $\gamma_{\max}$, коефіцієнтом λ_γ, інтервалом переоцінювання пар, параметрами відсоткової кривої, коефіцієнтами чутливості до швидкості використання ліквідності, списком дозволених публічних фідів, порогом застарівання ціни та режимом аварійної паузи.

Розділення модулів знижує операційний ризик: логіка оцінювання ризику, адміністрування та обліку активів має коротші інтерфейси, менше причин для зміни й простіший набір тестів. Узагальнену схему взаємодії модулів наведено на рис. 2.1.

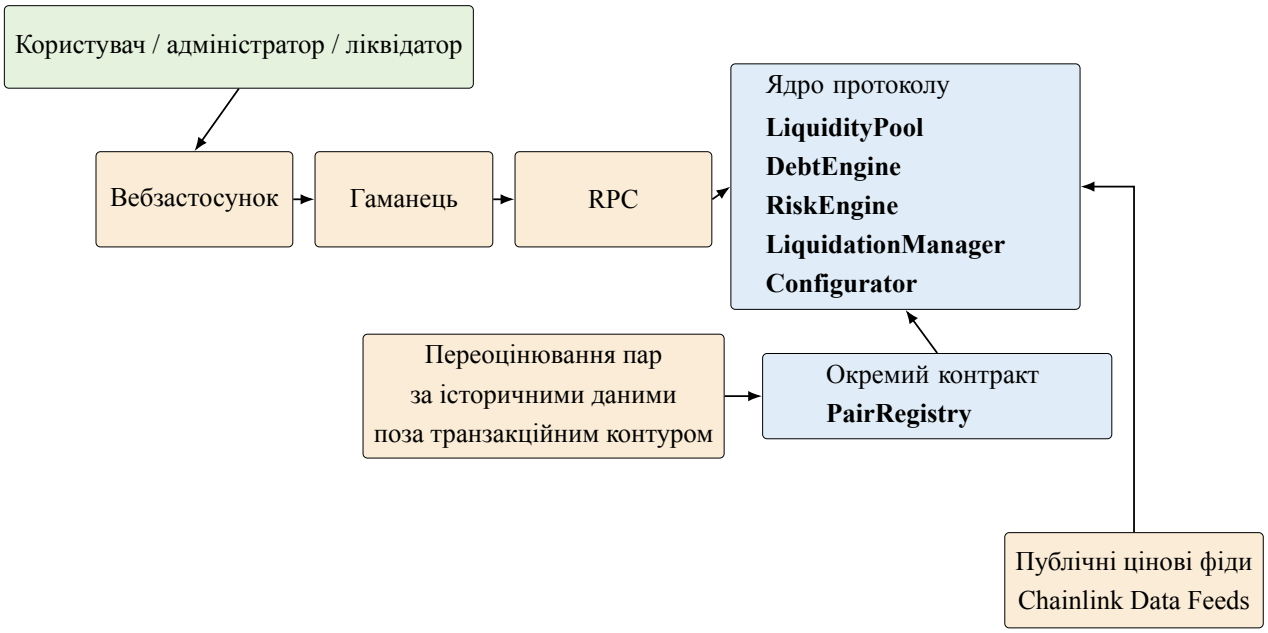

Рис. 2.1: Логічна архітектура безсерверної платформи для заставного кредитування

2.2 Ролі учасників і сценарії взаємодії

У межах платформи передбачено чотири функціональні ролі: користувач, ліквідатор, адміністратор параметрів ризику і технічний адміністратор. Звичайний користувач вносить ліквідність, відкриває позику, погашає борг і виводить забезпечення. Ліквідатор виконує лише операції, що дозволені для позицій із недостатнім забезпеченням. Адміністратор параметрів ризику не добирає пари вручну: він керує базовими порогами моделі й вводить у дію пакет автоматично обчислених параметрів пар. У практичній реалізації ці дії розділено між ролями PARAM_ADMIN і RISK_ADMIN, причому обидві ролі надано мультипідпису: перша змінює базові параметри моделі, керує паузою та списком дозволених публічних фідів, друга подає добовий пакет переоцінювання. Технічний адміністратор відповідає за технічну конфігурацію розгортання і рольову модель доступу; у практичній реалізації ці повноваження також зосереджено в тому самому мультипідписі.

Сценарій **внесення ліквідності** починається з перевірки підтримки активу та відсутності технічних обмежень, після чого протокол приймає токени, оновлює стан пулу й фіксує подію депозиту. Ця операція не змінює ризик уже відкритої позики користувача, якщо внесений актив не позначено як заставний.

Сценарій **отримання позики** є критичним. Перед його виконанням си-

Табл. 2.1: Рольова модель платформи

Роль	Повноваження
Користувач	Вносить ліквідність, використовує актив як заставу, відкриває позику, погашає борг, виводить забезпечення після проходження перевірок.
Ліквідатор	Погашає частину боргу проблемної позиції та отримує визначену правилами частку застави. Не має адміністративних прав.
Адміністратор параметрів ризику	Керує базовими параметрами LTV, ліквідаційного порогу, порогами $\rho_{\min}$, $\delta_{\max}$, $\kappa_{\min}$, параметрами ставки та вводить у дію пакет автоматично обчислених коефіцієнтів сумісності й виведених на їх основі меж ризику для пар активів. У практичній реалізації це розділено між PARAM_ADMIN і RISK_ADMIN; обидві ролі надано мультипідпису, а перша з них також керує паузою і списком дозволених публічних фідів.
Технічний адміністратор	Керує ролями доступу та технічними параметрами розгортання. У практичній реалізації ці дії виконує той самий мультипідпис, якому надано адміністративні ролі протоколу.

стема повинна перевірити, що пара «застава – актив позики» входить до поточного допустимого набору, оновити борг користувача, отримати актуальну ціну застави й активу позики, зчитати параметри ризику для конкретної пари, обчислити показник платоспроможності після гіпотетичної операції та відхилити транзакцію, якщо новий стан порушує правила. Лише після цього протокол переводить кошти користувачу та журналює подію запозичення.

Сценарій **погашення** зменшує борг і поліпшує профіль ризику. Він має бути можливим навіть у разі часткових обмежень системи, якщо це не суперечить аварійному режиму. Сценарій **виведення забезпечення** виконується лише після повторного обчислення допустимості позиції, бо саме ця операція може перевести здорову позицію у стан короткострокового дефіциту.

Сценарій **ліквідації** запускається, коли система фіксує значення показника платоспроможності нижче визначеного порогу. Ліквідатор погашає частину боргу позичальника, а протокол передає йому відповідну частину застави з ліквідаційним бонусом. Такий механізм зберігає життєздатність системи без централізованої служби стягнення.

Окремо існує сценарій **періодичного оновлення параметрів пари**. Він не є частиною користувацької транзакції. Позатранзакційний пакетний цикл переоцінювання на основі індексованого історичного ряду переобчислює $\rho_{cd}(W)$, $q_{cd}^{(\alpha)}(W)$, коефіцієнт сумісності пари та виведені на його основі межі ризику, після чого оновлює **PairRegistry**. Участь адміністратора параметрів ризику тут зводиться до задання базових порогів моделі та введення в дію нового пакета параметрів. До моменту такого оновлення транзакційний контур використовує останню затверджену конфігурацію. Узагальнений життєвий цикл заставної позиції показано на рис. 2.2.

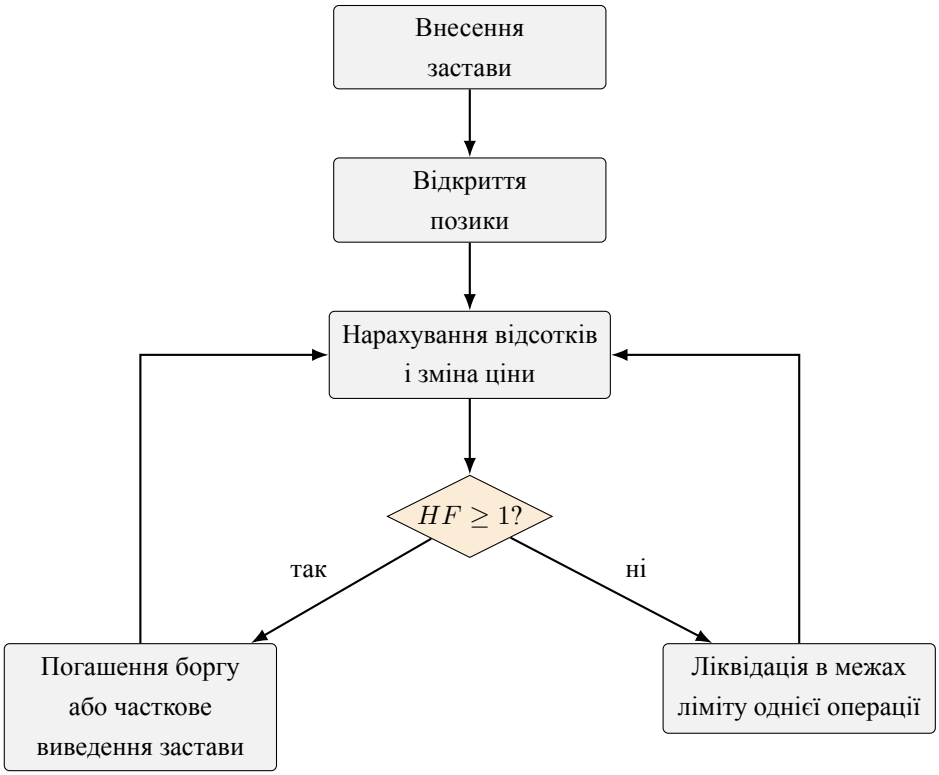

Рис. 2.2: Життєвий цикл заставної позиції в кредитному протоколі

2.3 Математична модель добору застави та оцінювання позиції

Модель має оцінювати не лише окремий актив застави, а й саму пару «застава – актив позики». Тому математичну модель побудовано для пари c/d, де c — заставний актив, d — актив позики, Q_c — кількість застави, B_d — кількість боргу, а P_c та P_d — їхні ціни в єдиній розрахунковій валюті.

Для заставного активу c задаються базові параметри ринку: LTV_c^{base} і

LT_c^{base}, причому завжди виконується умова

$$\Delta_c = LT_c^{base} - LTV_c^{base} > 0. \tag{2.1}$$

Величина Δ_c задає базовий буфер між моментом відкриття позиції та моментом її переходу в ліквідаційний стан. На відміну від статичної моделі окремого активу, у запропонованій моделі цей буфер не скасовується, а переноситься на рівень конкретної пари.

2.3.1 Автоматичний добір допустимих заставних активів і ефективні параметри ризику

Нехай для вікна з W спостережень логарифмічні доходності активів мають вигляд

$$r_c^{(k)} = \ln \frac{P_c^{(k)}}{P_c^{(k-1)}}, \qquad r_d^{(k)} = \ln \frac{P_d^{(k)}}{P_d^{(k-1)}}, \quad k = 1, \ldots, W. \tag{2.2}$$

Для оцінювання того, наскільки заставний актив поводиться безпечно відносно активу позики, однієї кореляції недостатньо. Потрібно також контролювати короткі відносні просідання ціни застави. Тому для пари c/d вводиться відносна логарифмічна дохідність

$$z_{cd}^{(k)} = r_c^{(k)} - r_d^{(k)} = \ln \frac{P_c^{(k)}/P_d^{(k)}}{P_c^{(k-1)}/P_d^{(k-1)}}. \tag{2.3}$$

Тоді коефіцієнт кореляції цінової динаміки пари визначається як

$$\rho_{cd}(W) = \frac{\sum_{k=1}^{W} \left(r_c^{(k)} - \bar{r}_c\right)\left(r_d^{(k)} - \bar{r}_d\right)}{\sqrt{\sum_{k=1}^{W} \left(r_c^{(k)} - \bar{r}_c\right)^2} \cdot \sqrt{\sum_{k=1}^{W} \left(r_d^{(k)} - \bar{r}_d\right)^2}}. \tag{2.4}$$

Нижній квантиль відносної дохідності визначається як

$$q_{cd}^{(\alpha)}(W) = \text{Quantile}_\alpha \left(\left\{ z_{cd}^{(1)}, \ldots, z_{cd}^{(W)} \right\} \right), \tag{2.5}$$

де α — рівень нижнього хвоста розподілу. Величина $q_{cd}^{(\alpha)}(W)$ показує, наскільки

глибоким може бути короткий рух відносної ціни застави до активу позики в несприятливому сегменті спостережень.

Відносна волатильність пари визначається як

$$\sigma_{cd}^{rel}(W) = \sqrt{\frac{1}{W-1}\sum_{k=1}^{W}\left(z_{cd}^{(k)} - \bar{z}_{cd}\right)^2}. \tag{2.6}$$

Використання логарифмічних дохідностей, кореляції, квантилів нижнього хвоста і волатильності відповідає стандартним інструментам кількісного ризик-менеджменту [31, 32].

У повній системі ці три величини обчислюються позаланцюгово на основі того самого потоку оракульних цін і періодично оновлюються в конфігурації ринку. Для активів позики, віднесених у конфігурації ринку до множини низьковолатильних активів $\mathcal{D}^{stb}$, кореляція доходностей не використовується як критерій допуску пари. Для таких активів визначальними є відносна волатильність $\sigma_{cd}^{rel}(W)$ і нижній квантиль $q_{cd}^{(\alpha)}(W)$, оскільки саме вони безпосередньо характеризують ризик короткого погіршення відношення $\frac{P_c}{P_d}$.

Щоб перейти від сирих ринкових характеристик до придатних для конфігурації параметрів, вводяться три нормовані складові. Перша описує узгодженість цінової динаміки:

$$s_\rho = \max\left(0, \min\left(1, \frac{\rho_{cd}(W) - \rho_{\min}}{1 - \rho_{\min}}\right)\right), \tag{2.7}$$

де $\rho_{\min}$ — мінімальний допустимий рівень кореляційної узгодженості пари.

Друга складова відображає глибину допустимого просідання відносної ціни:

$$s_q = \max\left(0, \min\left(1, \frac{q_{cd}^{(\alpha)}(W) + \delta_{\max}}{\delta_{\max}}\right)\right), \tag{2.8}$$

де $\delta_{\max} > 0$ — максимально допустима глибина короткого відносного падіння заставного активу.

Для випадку низьковолатильного активу позики $d \in \mathcal{D}^{stb}$ вводиться окрема нормована складова:

$$s_\sigma = \max\left(0, \min\left(1, \frac{\sigma_{\max}^{rel} - \sigma_{cd}^{rel}(W)}{\sigma_{\max}^{rel}}\right)\right), \tag{2.9}$$

де $\sigma^{rel}_{\max} > 0$ — максимально допустима відносна волатильність пари для випадку $d \in \mathcal{D}^{stb}$.

На цій основі для кожного активу позики d формується множина автоматично допустимих заставних активів:

$$\mathcal{C}_d^{adm} = \begin{cases} \left\{ c \in \mathcal{C} \;\middle|\; \rho_{cd}(W) > \rho_{\min},\ q^{(\alpha)}_{cd}(W) > -\delta_{\max} \right\}, & d \notin \mathcal{D}^{stb}, \\ \left\{ c \in \mathcal{C} \;\middle|\; \sigma^{rel}_{cd}(W) < \sigma^{rel}_{\max},\ q^{(\alpha)}_{cd}(W) > -\delta_{\max} \right\}, & d \in \mathcal{D}^{stb}. \end{cases} \tag{2.10}$$

Строгі нерівності узгоджують допуск пари з подальшим розрахунком a_{cd}: пара, що лежить рівно на порозі, не потрапляє до допустимого набору з нульовою межею позики. Для звичайного активу позики відсікаються кандидати, які не мають достатньої спільної динаміки або демонструють надто глибокі короткі провали відносної ціни. Для низьковолатильного активу позики протокол замінює кореляційний критерій обмеженням на відносну волатильність пари. Це зберігає підтримку типових пар на кшталт «волатильна застава – стейблкоїн», але не допускає заставу, у якої відношення $\frac{P_c}{P_d}$ коливається надто різко.

У практичній архітектурі вміст множини $\mathcal{C}_d^{adm}$ записується в **PairRegistry** для конкретного активу позики d, і саме цей реєстр використовується під час допуску нової позики.

Для допустимої пари вводиться коефіцієнт сумісності:

$$a_{cd} = \begin{cases} 0, & c \notin \mathcal{C}_d^{adm}, \\ \min\left(s_\rho, s_q\right), & c \in \mathcal{C}_d^{adm},\ d \notin \mathcal{D}^{stb}, \\ \min\left(s_\sigma, s_q\right), & c \in \mathcal{C}_d^{adm},\ d \in \mathcal{D}^{stb}, \end{cases} \tag{2.11}$$

а коефіцієнт зниження базових параметрів пари визначається так:

$$\phi_{cd} = \begin{cases} 0, & a_{cd} = 0, \\ \kappa_{\min} + (1 - \kappa_{\min}) \cdot a_{cd}, & a_{cd} > 0. \end{cases} \tag{2.12}$$

де $\kappa_{\min} \in (0, 1)$ — мінімально допустима частка від базового рівня ризикових параметрів для пари, яка вже пройшла автоматичний відбір.

Тоді ефективний LTV для пари «застава – актив позики» визначається

формулою

$$LTV_{cd}^{eff} = LTV_{c}^{base} \cdot \phi_{cd}. \tag{2.13}$$

Ефективний ліквідаційний поріг задається так:

$$LT_{cd}^{eff} = \begin{cases} 0, & \phi_{cd} = 0, \\ \min\left(LT_{c}^{base}, LTV_{cd}^{eff} + \Delta_{c}\right), & \phi_{cd} > 0. \end{cases} \tag{2.14}$$

Означення дає дві властивості. Для недопустимого заставного активу $\phi_{cd} = 0$, тому відкриття боргу блокується без додаткових евристик. Для допустимої пари зберігається базовий буфер Δ_c, але його абсолютне значення переноситься на рівень конкретної пари. Ця властивість усуває слабке місце статичної моделі, у якій однакові LTV і LT задають і для безпечної застави, і для застави з гіршою відносною динамікою до того самого активу позики.

Поріг швидкого блокування також виводиться автоматично з хвостового ризику пари:

$$\gamma_{cd} = \min\left(\gamma_{\max}, \max\left(\gamma_{\min}, \lambda_{\gamma} \cdot \max\left(0, -q_{cd}^{(\alpha)}(W)\right)\right)\right), \tag{2.15}$$

де $\gamma_{\min}$ і $\gamma_{\max}$ задають нижню та верхню межі швидкого блокування, а λ_{γ} масштабує чутливість правила до глибини несприятливого квантиля. Унаслідок цього ланцюговий захист спрацьовує раніше, ніж добове переоцінювання встигне виключити пару з $\mathcal{C}_{d}^{adm}$.

У запропонованій моделі вручну задаються лише базові параметри ринку і пороги самої процедури відбору: LTV_{c}^{base}, LT_{c}^{base}, $\rho_{\min}$, $\delta_{\max}$, $\sigma_{\max}^{rel}$, $\kappa_{\min}$, $\gamma_{\min}$, $\gamma_{\max}$, λ_{γ}, а також параметри відсоткової кривої. Для кожної конкретної пари автоматично виводяться коефіцієнт сумісності a_{cd}, коефіцієнт зниження ϕ_{cd}, ефективний LTV_{cd}^{eff}, ефективний ліквідаційний поріг LT_{cd}^{eff}, поріг швидкого блокування γ_{cd}, похідна ціна ліквідації і поточний показник платоспроможності позиції.

Умова допустимості нової позики або збільшення чинної позиції має вигляд

$$B_{d} \cdot P_{d} \leq Q_{c} \cdot P_{c} \cdot LTV_{cd}^{eff}. \tag{2.16}$$

Показник платоспроможності позиції для пари c/d визначається як

$$HF_{cd} = \frac{Q_c \cdot P_c \cdot LT_{cd}^{eff}}{B_d \cdot P_d}. \tag{2.17}$$

За відсутності боргу значення HF_{cd} вважається нескінченно великим. Якщо $HF_{cd} \geq 1$, позиція залишається в допустимому стані. Якщо $HF_{cd} < 1$, позиція переходить у стан, який дозволяє ліквідацію.

Для перевірки нової операції протокол не може покладатися на поточний стан позиції. Йому необхідно моделювати стан *після* виконання дії. Якщо користувач хоче збільшити борг на суму ΔB_d, система повинна перевірити умову

$$\frac{Q_{c,\text{new}} \cdot P_c \cdot LT_{cd}^{eff}}{(B_d + \Delta B_d) \cdot P_d} \geq 1, \tag{2.18}$$

де $Q_{c,\text{new}}$ враховує можливе виведення частини застави або зміну її складу. Тільки такий підхід блокує операції, які виглядають безпечними до виконання, але створюють небезпечний стан після оновлення балансу.

Для користувача і для децентралізованого застосунку важливо мати явну оцінку того, за якої ціни застави позиція втратить безпечний стан. Для однієї заставної пари ціна ліквідації щодо заставного активу дорівнює

$$P_{c \to d}^{liq} = \frac{B_d \cdot P_d}{Q_c \cdot LT_{cd}^{eff}}. \tag{2.19}$$

Формула показує прикладний зміст ліквідаційного порогу для конкретної пари: чим нижчий LT_{cd}^{eff}, тим раніше настане критична ціна застави для тієї самої величини боргу.

Щоб контролювати придатність цінових даних, вводиться перевірка

$$t_{\text{now}} - t_{\text{update}} \leq \tau_{\text{stale}}, \tag{2.20}$$

де t_{update} — час останнього оновлення відповіді оракула, а τ_{stale} — допустиме вікно актуальності ціни. Якщо нерівність не виконується, протокол блокує видачу нової позики, виведення застави та ліквідацію до відновлення коректного потоку даних [25].

Нова позика проходить чотири послідовні перевірки: пара має входити до

поточного **PairRegistry**; цінові дані мають бути актуальними; для пари мають бути визначені LTV_{cd}^{eff} і LT_{cd}^{eff}; змодельований стан позиції після транзакції не повинен опускати новий HF_{cd} нижче допустимої межі.

2.4 Відсоткова модель, ліквідність і правила ліквідації

Стійкість платформи залежить від застави і від здатності пулу обслуговувати попит на ліквідність. Для цього вводиться коефіцієнт використання пулу:

$$U_t = \frac{B}{A + B}, \tag{2.21}$$

де B — сумарний обсяг позичених активів, а A — доступна ліквідність пулу після урахування резерву. Така форма відповідає практиці розрахунку використання як частки позиченого активу в сумі доступної та вже виданої ліквідності.

Базова позикова ставка має зростати разом із U_t. Для цього використано кусочно-лінійну модель з точкою перегину:

$$r_b^{base}(U_t) = \begin{cases} r_0 + s_1 \cdot U_t, & U_t \le U_k, \\ r_0 + s_1 \cdot U_k + s_2 \cdot (U_t - U_k), & U_t > U_k, \end{cases} \tag{2.22}$$

де r_0 — базова ставка, s_1 — нахил до точки перегину, s_2 — прискорене зростання після точки перегину, U_k — поріг використання пулу. Така модель узгоджується з логікою Compound: до помірного рівня використання пул працює плавно, а після критичного порогу система різко збільшує вартість нового боргу [6, 9, 12].

Щоб врахувати швидке погіршення ліквідності, вводиться додатна швидкість зміни використання пулу:

$$g_U = \max\left(0, \frac{U_t - U_{t-\Delta t}}{\Delta t}\right), \qquad g_n = \min\left(1, \frac{g_U}{g_{crit}}\right). \tag{2.23}$$

У практичній реалізації величина $U_{t-\Delta t}$ зберігається в полі lastUtilizationBps структури MarketState і оновлюється під час нарахування відсотків. У математичній моделі Δt вимірюється в годинах, тому g_U має розмірність год$^{-1}$, узгоджену з параметром g_{crit}, заданим у табл. 2.3. У смартконтракті різниця між двома значеннями block.timestamp спочатку обчислюється в секундах, а перед

розрахунком g_U нормалізується до годин. Це зберігає однакову часову шкалу для g_U, g_n і g_{crit}. Поділ між поточним рівнем використання і швидкістю його зростання узгоджується з класичними підходами до вимірювання ринкового й ліквіднісного ризику [33, 34].

Запропонована модель додає до класичної кривої з точкою перегину поправку на швидкість зростання коефіцієнта використання ліквідності. Модифікована ставка визначається як

$$r_b(U_t, g_n) = r_b^{base}(U_t) \cdot (1 + \beta_g \cdot g_n)\,, \tag{2.24}$$

де $\beta_g \in [0, 1]$ — коефіцієнт чутливості позикової ставки до швидкості вичерпання ліквідності.

Якщо $g_n = 0$, модель зводиться до звичайної кривої ставки, що залежить від використання пулу. Якщо використання пулу зростає швидко, ставка збільшується пропорційно вже наявному рівню напруження ринку. Модифікація не переоцінює короткий сплеск у спокійному стані, але підсилює захист у ситуації, коли високий коефіцієнт використання ліквідності ще й продовжує швидко зростати.

Ставка для постачальників ліквідності визначається від позикової ставки з урахуванням використання пулу та резервного коефіцієнта:

$$r_s(U_t, g_n) = r_b(U_t, g_n) \cdot U_t \cdot (1 - rf), \tag{2.25}$$

де rf — резервний коефіцієнт. Частина відсоткового потоку надходить до резерву протоколу й може бути використана для покриття технічних витрат, майбутніх збитків або розвитку системи.

Процедура ліквідації має виконувати три вимоги. По-перше, вона активується лише після підтвердження, що $HF_{cd} < 1$. По-друге, одна операція ліквідації не повинна одразу погашати весь борг, якщо для цього немає окремих підстав. Для цього вводиться коефіцієнт закриття (*close factor*), який лімітує обсяг погашення за один виклик. По-третє, ліквідатор повинен отримувати стимул у вигляді бонусу до вилученої застави, інакше протокол не матиме зовнішнього механізму швидкого реагування.

Якщо позичальник має борг D, а гранична частка однієї ліквідації дорів-

нює cf, максимальний обсяг боргу для однієї операції ліквідації становить

$$D_{\text{liq}} = D \cdot cf. \tag{2.26}$$

Кількість застави, яку ліквідатор отримує за погашення частини боргу на суму x, задану в єдиній розрахунковій валюті, можна подати так:

$$Q_{\text{seize}} = \frac{x \cdot (1 + bonus)}{p_c}, \tag{2.27}$$

де $bonus$ — бонус ліквідатора, а p_c — ринкова ціна заставного активу. Конкретна реалізація повинна враховувати десяткові розряди токенів, округлення та внутрішній формат представлення ціни.

2.5 Технічні запобіжники та керування параметрами моделі

Смартконтрактна платформа не може покладатися лише на коректність фінансових формул. Вона повинна містити набір технічних запобіжників, які обмежують наслідки помилки або ворожої взаємодії.

Перший базовий запобіжник — захист від повторного входу. Операції депозиту, виведення коштів, позики, погашення та ліквідації працюють з активами й зовнішніми викликами, тому належать до критичних функцій. Для них необхідно або суворо дотримуватися шаблону *checks-effects-interactions*, або доповнювати його ReentrancyGuard [28, 26].

Другий запобіжник — аварійна пауза. Протокол не повинен мати режиму «все або нічого», коли будь-яка підозра на інцидент вимагає повного зупинення системи. Доцільніше зберігати окремі прапори паузи для нових позик, ліквідацій та адміністративних змін. Тоді, наприклад, у разі проблеми з оракулом можна тимчасово зупинити видачу нових позик, але не блокувати погашення чинних боргів.

Третій запобіжник — деталізована рольова модель. Для невеликого протоколу спокуса звести всі привілеї до owner є високою, але цей підхід робить систему складною для аудиту й небезпечною в експлуатації. Використання ролей RISK_ADMIN та PARAM_ADMIN дозволяє відокремити добове пакетне оновлення пар від зміни базових параметрів, аварійної паузи та списку дозволених публічних фідів [27]. У практичній реалізації обидві ролі надано мульти-

підпису, а не окремим зовнішнім адресам.

Четвертий запобіжник — повне журналювання змін параметрів ризику. Будь-яке підвищення LTV, зміна ліквідаційного порогу, оновлення адреси оракула чи зняття паузи повинні створювати подію з попереднім і новим значенням. Це спрощує аудит, індексацію стану та постінцидентний аналіз.

Сукупність описаних засобів формує технічний каркас платформи. Без нього навіть правильна модель HF і відсоткових ставок не гарантує безпечної роботи системи в публічній мережі.

2.6 Вибір параметрів ризику для демонстраційного ринку

Для практичної реалізації було зафіксовано демонстраційний набір параметрів, який одночасно є достатньо консервативним для захисту системи і достатньо ліберальним для відтворення сценаріїв позики та ліквідації.

Табл. 2.2: Параметри сумісності пари та забезпечення

Параметр	Значення	Призначення
Базовий LTV_c^{base}	70%	Максимальна частка вартості застави для референтного активу до коригування на рівні конкретної пари.
Базовий LT_c^{base}	80%	Межа, нижче якої позиція переходить у стан придатності до ліквідації до коригування на рівні конкретної пари.
$\rho_{\min}$	0,40	Мінімально допустимий рівень узгодженості цінової динаміки пари.
α	0,05	Рівень нижнього квантиля для оцінювання коротких відносних просідань застави.
$\delta_{\max}$	15%	Максимально допустиме коротке падіння відносної ціни застави до активу позики.
$\sigma_{\max}^{rel}$	12%	Максимально допустима відносна волатильність пари для ринку з низьковолатильним активом позики.
$\kappa_{\min}$	0,75	Мінімальна частка базових параметрів для пари, що вже пройшла автоматичний відбір.
Інтервал переоцінювання пар	24 години	Період оновлення $\rho_{cd}(W)$, $q_{cd}^{(\alpha)}(W)$, a_{cd} та ϕ_{cd} у **PairRegistry**.
Бонус ліквідатора	5%	Стимул зовнішньому учаснику за погашення частини проблемного боргу.
Ліміт однієї ліквідації	50%	Максимальна частка боргу, яку можна погасити в одній операції ліквідації.

Графічну форму позикової ставки для вибраного набору параметрів на-

Табл. 2.3: Параметри ставки, ліквідності та оракула

Параметр	Значення	Призначення
Резервний коефіцієнт	10%	Частина нарахованих відсотків, що резервується протоколом.
Базова ставка	2% річних	Мінімальна вартість позики в спокійному режимі ринку.
Перший нахил ставки	8%	Плавне зростання ставки до точки перегину.
Другий нахил ставки	30%	Прискорене зростання ставки після перевантаження пулу.
Точка перегину	80% використання	Межа, після якої дефіцит ліквідності вважається істотним.
β_g	0,35	Чутливість позикової ставки до швидкості зростання коефіцієнта використання ліквідності.
g_{crit}	0,10 год$^{-1}$	Швидкість зростання коефіцієнта використання ліквідності, за якої ринок переходить у виразний стресовий режим.
$\gamma_{\min}$	3%	Нижня межа швидкого блокування відносної ціни пари.
$\gamma_{\max}$	12%	Верхня межа швидкого блокування відносної ціни пари.
λ_γ	0,50	Коефіцієнт масштабування порога швидкого блокування від глибини несприятливого квантиля.
Допустима давність ціни	3 години	Максимальний вік відповіді оракула для критичних операцій.

Табл. 2.4: Приклад добору застави та розрахунку ефективних параметрів пари

Кандидат застави	$\sigma_{cd}^{rel}(W)$	$q_{cd}^{(0,05)}(W)$	LTV_{cd}^{eff}	LT_{cd}^{eff}	Інтерпретація
Високоякісна застава	4%	-3%	66,50%	76,50%	Актив автоматично допускається і працює майже з базовими параметрами.
Допустима застава зі зниженням лімітів	9%	-7%	57,75%	67,75%	Актив допускається, але межі для цієї пари помітно знижуються.
Недопустима застава	18%	-18%	0%	0%	Пара відхиляється через надто високу відносну волатильність і глибокі короткі просідання відносної ціни.

ведено на рис. 2.3. Вона показує базову криву ставки, що залежить від використання пулу, та прискорену криву для випадку, коли коефіцієнт використання ліквідності вже зростає зі швидкістю $g_n = 1$.

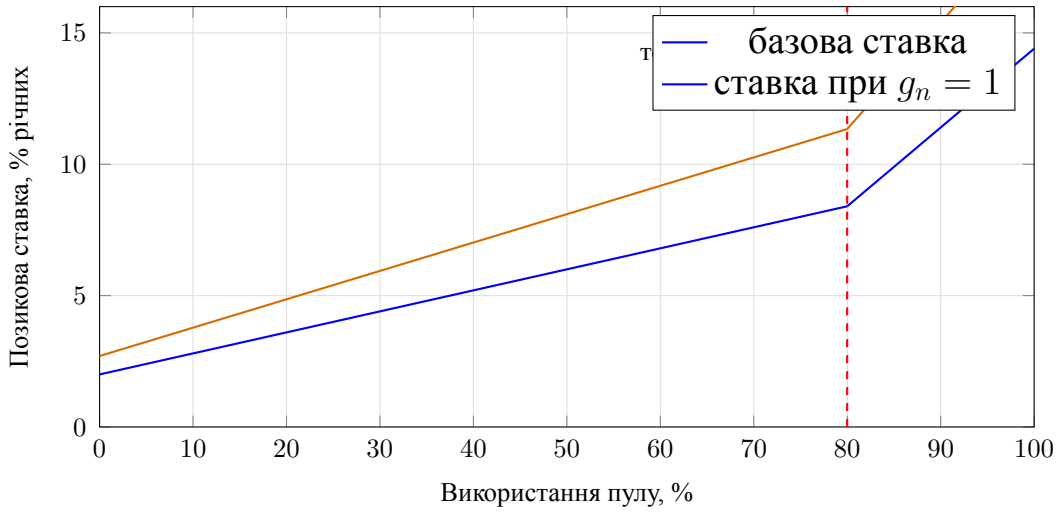

Рис. 2.3: Базова та прискорена позикова ставка для демонстраційного набору параметрів ризику

Вибір цих значень має змістове обґрунтування. Базові $LTV_c^{base} = 70\%$ і $LT_c^{base} = 80\%$ створюють 10-відсотковий буфер між моментом видачі позики й моментом втрати здорового стану позиції. Значення $\rho_{\min} = 0{,}40$ не вимагає майже тотожної поведінки двох активів, але відсікає пари без достатньої спільної динаміки в ринках, де актив позики не є низьковолатильним. Параметр $\sigma_{\max}^{rel} = 12\%$ обмежує розмах коливань відносної ціни $\frac{P_c}{P_d}$ для ринків зі стейблкоїном або іншим низьковолатильним активом позики. Параметр $\delta_{\max} = 15\%$ обмежує глибину короткого відносного падіння заставного активу до активу позики і тим самим відсікає заставу, що може занадто швидко знецінитися. Значення $\kappa_{\min} = 0{,}75$ не дозволяє парі, яка лише щойно пройшла автоматичний відбір, одразу отримати повний набір базових параметрів. Поєднання $\rho_{\min}$, $\sigma_{\max}^{rel}$, $\delta_{\max}$ і $\kappa_{\min}$ задає зміст автоматичного коригування параметрів пари, а параметри $\gamma_{\min}$, $\gamma_{\max}$ і λ_γ обмежують чутливість швидкого блокування. Бонус ліквідатора на рівні 5% забезпечує економічний стимул, але не переводить ліквідацію в режим надмірного вилучення застави.

Обрані параметри відсоткової моделі відповідають логіці керування ліквідністю. При використанні пулу нижче 80% базова ставка зростає плавно, що робить систему придатною до звичайних ринкових коливань. Після дося-

гнення точки перегину швидке зростання ставки стримує подальше запозичення та стимулює повернення ліквідності. Додатковий множник β_g не змінює логіку базової кривої, але робить її чутливою до швидкості погіршення ринку. У термінах управління ризиком це означає, що протокол не підтримує постійно завищену вартість позики, але й не ігнорує ситуацію, коли дефіцит ліквідності формується надто швидко.

2.7 Числові сценарії поведінки позиції

Числові сценарії нижче показують, як працюють вибрані параметри. Для чистоти порівняння використано еталонну автоматично допустиму пару з $a_{cd} = 1$ і $\phi_{cd} = 1$, тобто таку, для якої актив застави входить до допустимого набору для активу позики з максимальним коефіцієнтом сумісності, а ефективні параметри збігаються з базовими. Нехай користувач вносить 10 одиниць заставного активу, а ринкова ціна однієї одиниці становить 2000 умовних доларів. Тоді повна вартість застави дорівнює 20 000, а скоригована за ліквідаційним порогом вартість — 16 000.

Сценарій 1: помірна позика. Якщо користувач бере 5000 одиниць боргового активу зі стабільною ціною 1 долар, то

$$HF_{cd} = \frac{16\,000}{5\,000} = 3{,}2. \tag{2.28}$$

Позиція має значний запас стійкості. Користувач може погасити частину боргу, дочекатися зниження використання пулу або, за потреби, вивести частину застави без переходу до небезпечного стану.

Сценарій 2: агресивніша позика. Якщо за тих самих 10 одиниць застави користувач бере 12 000 одиниць активу позики, поточний показник платоспроможності становить

$$HF_{cd} = \frac{16\,000}{12\,000} \approx 1{,}33. \tag{2.29}$$

Позиція все ще є допустимою, але запас міцності вже вузький. Якщо ціна застави знижується на 25%, її повна вартість падає до 15 000, а скоригована — до 12 000. Тоді $HF_{cd} = 1$, тобто позиція переходить у прикордонний стан. Подальше навіть незначне падіння ціни робить її ліквідовуваною.

Сценарій 3: різке падіння ціни. Якщо ціна застави падає удвічі, до 1000 доларів за одиницю, повна вартість застави становить 10 000, а скоригована —

8 000. За боргу 12 000 отримуємо

$$HF_{cd} = \frac{8\,000}{12\,000} \approx 0{,}67. \tag{2.30}$$

У такому разі позиція переходить у стан, за якого ліквідатор може погасити частину боргу. Якщо він погашає 3000 доларів, а бонус становить 5%, то при ціні застави 1000 доларів вилучена кількість застави дорівнює

$$Q_{\text{seize}} = \frac{3000 \cdot 1{,}05}{1000} = 3{,}15. \tag{2.31}$$

Після цієї операції в позиції залишається $10 - 3{,}15 = 6{,}85$ одиниці застави. Її повна вартість за тієї самої ринкової ціни становить 6850 доларів, а скоригована за ліквідаційним порогом — $6{,}85 \cdot 1000 \cdot 0{,}8 = 5480$ доларів. За залишку боргу 9000 доларів новий показник платоспроможності дорівнює

$$HF_{cd} = \frac{5480}{9000} \approx 0{,}609. \tag{2.32}$$

Це показує, що часткова ліквідація з бонусом не обов'язково повертає позицію до здорового стану. Якщо ринкова ціна не відновлюється, HF_{cd} після ліквідації може залишитися нижчим за одиницю і навіть зменшитися відносно значення до операції.

Табл. 2.5: Числовий приклад зміни стану позиції для конкретної пари

Стан	Вартість застави	Борг	HF_{cd}
Після внесення застави, до позики	20 000	0	нескінченність
Після позики 5000	20 000	5 000	3,2
Після позики 12 000	20 000	12 000	1,33
Після падіння ціни на 25%	15 000	12 000	1,0
Після падіння ціни на 50%	10 000	12 000	0,67
Після часткової ліквідації на 3000	6 850	9 000	0,609

Ці сценарії показують, що запропоновані параметри не є формальним набором чисел. Вони створюють передбачувану траєкторію переходу позиції від безпечного стану до прикордонного, а далі до ліквідації. Ключовими тут є правило автоматичного добору застави й межі позики для конкретної пари, які не дозволяють однаково оцінювати безпечний і ризиковий заставні активи для того самого активу позики. Якщо параметри занадто м'які, система починає накопичувати дефіцит забезпечення. Якщо занадто жорсткі, позика стає практично непридатною для користувача.

Висновки до розділу 2

У розділі запропоновано модульну архітектуру платформи з виокремленням пулу ліквідності, обліку боргу, реєстру допустимих пар, оцінювання ризику, перевірки цінових даних, ліквідації та конфігурації параметрів. Визначено рольову модель і критичні сценарії взаємодії, включно з окремим контуром періодичного переоцінювання пар активів. Формалізовано правило автоматичного добору допустимих заставних активів для заданого активу позики та автоматичного виведення ключових меж кредитного ризику для конкретної пари: ефективного LTV, ефективного ліквідаційного порогу, похідної ціни ліквідації

і показника платоспроможності. Окремо подано коефіцієнт використання пулу, швидкість його зміни та модифіковану модель ставки. Окреслено технічні запобіжники, без яких система не може вважатися придатною до практичного використання.

Запропонована модель не зводить ризик до параметрів одного активу. Для кожної пари (c, d) побудовано послідовний перехід від статистичних характеристик ринку до рішення про допустимість застави, ефективних меж LTV і LT, похідної ціни ліквідації та умов швидкого блокування. Для ринків з низьковолатильним активом позики введено окреме правило, яке спирається не на формально слабку кореляцію доходностей, а на відносну волатильність пари та нижній квантиль відносної дохідності.

Числові сценарії показали, що вибрані параметри формують передбачувану траєкторію поведінки позиції: від стійкого стану до прикордонного і далі до ліквідації. Модифікована відсоткова модель додатково підсилює реакцію системи на швидке погіршення ліквідності. Це завершує проєктний і математичний фундамент платформи та створює пряму основу для практичної реалізації, поданої в наступному розділі.

РОЗДІЛ 3
ПРОЄКТ РЕАЛІЗАЦІЇ ТА ПЕРЕВІРКИ ПЛАТФОРМИ

3.1 Структура практичної реалізації

Практичну реалізацію платформи побудовано з трьох частин: ланцюгового ядра, позаланцюгового контуру добового переоцінювання пар та безсерверного прикладного рівня. Автоматичний добір допустимих пар і автоматичне виведення меж ризику для конкретної пари винесено за межі шляху кожної користувацької транзакції. Перевірка свіжості цін, показника HF_{cd} і швидкого блокування виконується безпосередньо в смартконтракті.

Табл. 3.1: Технологічний стек практичної реалізації

Компонент	Призначення
Solidity 0.8.24	Реалізація смартконтрактів платформи з використанням іменованих помилок, контрольованих модифікаторів доступу та перевірок арифметики [29].
Foundry / Forge	Компіляція контрактів, сценарні тести, часові зсуви та підготовка звітів для практичної перевірки [30].
Node.js / TypeScript	Середовище виконання позаланцюгового добового скрипта переоцінювання, який формує пакет параметрів допустимих пар для batchUpdatePairs.
Chainlink Data Feeds	Джерело поточних цін для ланцюгових перевірок свіжості даних, показника платоспроможності та швидкого блокування; історичний ряд для добового переоцінювання формується позаланцюгово з індексованих відповідей цих цінових фідів [25].
Добовий скрипт переоцінювання	Позаланцюгове обчислення $\rho_{cd}(W)$, $q_{cd}^{(\alpha)}(W)$, a_{cd}, ϕ_{cd}, LTV_{cd}^{eff} і LT_{cd}^{eff} з формуванням пакета для транзакції batchUpdatePairs, яку підтверджує мультипідпис із роллю RISK_ADMIN.
Статичний застосунок	Відображення доступних пар, поточного стану позиції, меж ризику, стану швидкого блокування і підготовка транзакцій через гаманець користувача.

На рівні розгортання система має один основний контракт протоколу й окремий контракт **PairRegistry**. В основному контракті зосереджено п’ять внутрішніх модулів: **LiquidityPool**, **DebtEngine**, **RiskEngine**, **LiquidationManager** і **Configurator**.

LiquidityPool веде облік активу позики та часток постачальників ліквідності. **DebtEngine** підтримує масштабований борг і глобальний індекс. **PairRegistry** зберігає автоматично виведені параметри конкретної пари.

RiskEngine поєднує поточні ціни з **Chainlink Data Feeds**, параметри пари та стан швидкого блокування. **LiquidationManager** виконує часткове закриття проблемної позиції. **Configurator** керує ролями, паузою, адресами фідів **Chainlink Data Feeds** і пакетними оновленнями пар. У цій реалізації мультипідпису надано обидві адміністративні ролі: RISK_ADMIN для добового пакета і PARAM_ADMIN для базових порогів моделі, паузи та списку дозволених фідів. Контур практичної реалізації з добовим пакетом, мультипідписом і ланцюговим ядром наведено на рис. 3.1.

Скорочені лістинги ключових структур стану, добового скрипту переоцінювання і тестового сценарію наведено в додатках А–В.

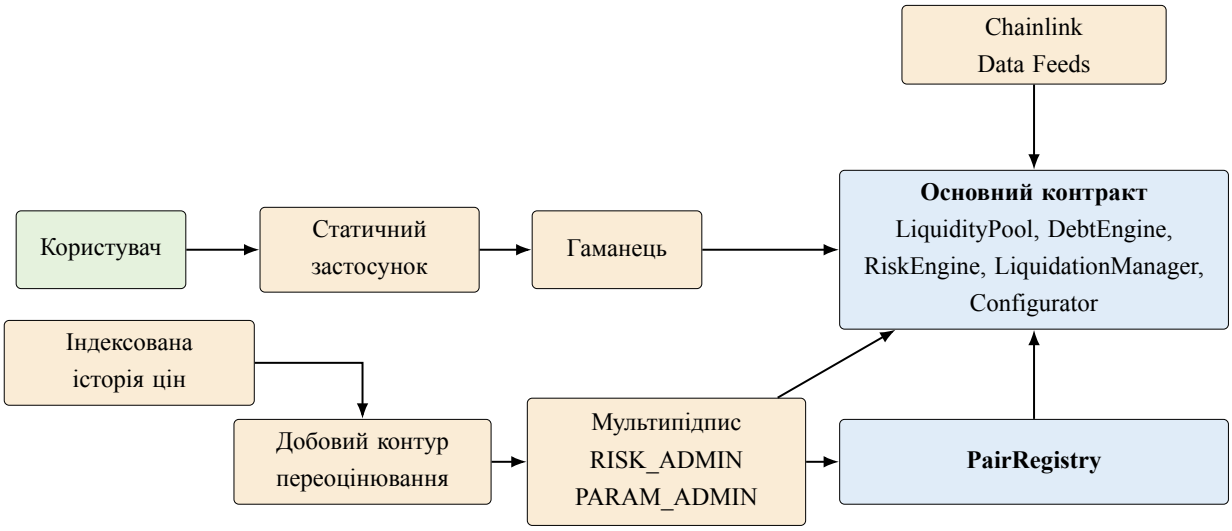

Рис. 3.1: Контур практичної реалізації платформи

3.2 Модель зберігання стану

Реалізація не зводить усю позицію до одного універсального запису на користувача. Оскільки параметри ризику залежать від конкретної пари активів, позиція ізолюється за трійкою (u, c, d), де u — користувач, c — заставний актив, d — актив позики. Ізоляція за цією трійкою усуває неоднозначність під час застосування LTV_{cd}^{eff}, LT_{cd}^{eff}, ціни ліквідації та швидкого блокування для конкретної пари.

Ринковий стан для активу позики складається з двох частин. Перша описує ліквідність: суму активу в пулі, кількість випущених часток, резерв і вільну ліквідність. Друга описує борг: масштабований борг totalScaledDebt, глобальний індекс borrowIndexRay, часову мітку останнього нарахування, попередній коефіцієнт використання ліквідності lastUtilizationBps та параметри відсоткової кривої. Така структура дозволяє оновлювати відсотки без ітерації по всіх позичальниках і прямо підтримує розрахунок g_U. Оскільки часові мітки в ланцюгу зберігаються в секундах Unix-часу, під час ланцюгового розрахунку g_U різницю між block.timestamp і lastAccrualTimestamp нормалізують до годин. Це узгоджує реалізацію з параметром g_{crit}, який у моделі задано в год$^{-1}$.

У ланцюговому поданні відсоткові параметри та пороги зберігаються в базисних пунктах (*bps*), а індекси й відношення — у форматі *ray* з масштабом 10^{27}. Тому в лістингах далі використано поля на кшталт ltvBaseBps, aCdBps, borrowIndexRay і referenceRatioRay.

Табл. 3.2: Основні сутності стану ланцюгового ядра

Сутність	Поля
AssetConfig	Адреса токена, адреса **Chainlink**-фіду, базовий LTV^{base}, базовий LT^{base}, гранична давність ціни, ознака доступності активу.
MarketState	totalLiquidityShares, totalScaledDebt, borrowIndexRay, reserveBalance, lastAccrualTimestamp, lastUtilizationBps, параметри кривої ставки, прапори паузи.
Position	Кількість застави, масштабований борг, ідентифікатор заставного активу, ідентифікатор активу позики, службові часові мітки.
PairConfig	allowed, aCdBps, phiCdBps, effectiveLtvBps, effectiveLiquidationThresholdBps, referenceRatioRay, guardThresholdBps, updatedAt, epoch.

Структура PairConfig фіксує результат автоматичного добору застави та переведення статистичних характеристик пари в ланцюгові параметри ризику. Після запису нового пакета в **PairRegistry** транзакційний контур уже не звертається напряму до кореляції чи квантилів, а працює з виведеними величинами, які обмежують позику, ліквідацію та швидке блокування.

Табл. 3.3: Ключові зовнішні виклики практичної реалізації

Виклик	Призначення
supplyLiquidity	Параметри: debtAsset, amount. Внесення активу позики до пулу ліквідності з нарахуванням часток постачальника ліквідності.
withdrawLiquidity	Параметри: debtAsset, shareAmount. Виведення частини ліквідності за умови достатнього резерву ринку та відсутності паузи.
depositCollateral	Параметри: collateralAsset, debtAsset, amount. Внесення застави до ізольованої позиції (u, c, d).
borrow	Параметри: collateralAsset, debtAsset, amount. Відкриття або збільшення позики після перевірки допустимості пари, свіжості цін, стану швидкого блокування і HF_{cd} після операції.
repay	Параметри: collateralAsset, debtAsset, amount. Погашення частини або всього боргу без блокування навіть у кризовому режимі.
withdrawCollateral	Параметри: collateralAsset, debtAsset, amount. Виведення частини застави після моделювання стану позиції та повторної перевірки меж ризику для конкретної пари.
liquidate	Параметри: borrower, collateralAsset, debtAsset, repayAmount. Часткова ліквідація позиції з урахуванням коефіцієнта закриття, бонусу ліквідатора та поточної ціни активів.
batchUpdatePairs	Параметри: batch, epoch. Пакетне оновлення **PairRegistry**, яке виконує мультипідпис із роллю RISK_ADMIN.
getPairConfig	Параметри: collateralAsset, debtAsset. Читання актуальних параметрів пари для децентралізованого застосунку і ланцюгових перевірок.

3.3 Алгоритми реалізації основних операцій

3.3.1 Добове оновлення PairRegistry

Позаланцюговий контур виконує пакетне переоцінювання пар один раз на добу. Він отримує індексовані історичні дані цінових фідів, вирівнює часову сітку, обчислює $r_c^{(k)}$, $r_d^{(k)}$, $z_{cd}^{(k)}$, а далі виводить $\rho_{cd}(W)$, $q_{cd}^{(\alpha)}(W)$, $\sigma_{cd}^{rel}(W)$, a_{cd}, ϕ_{cd}, LTV_{cd}^{eff}, LT_{cd}^{eff} та опорне значення

$$R_{cd}^{ref} = \frac{P_c^{ref}}{P_d^{ref}}. \tag{3.1}$$

Скрипт формує пакет параметрів разом із часовою міткою та номером епохи, а запис у **PairRegistry** виконує мультипідпис, якому надано роль RISK_ADMIN.

У практичній реалізації цей цикл не втручається в логіку окремої транзакції. Якщо добове переоцінювання не відбулося вчасно, система продовжує працювати з останнім затвердженим пакетом параметрів. Таке рішення зберігає працездатність платформи, але не робить її беззахисною до різкого ринкового руху, бо швидке блокування перевіряється окремо при кожному ризиковому виклику.

Табл. 3.4: Етапи пакетного оновлення параметрів пар

Крок	Зміст
1	Отримати індексовані історичні дані цінових фідів для всіх кандидатів на роль застави й усіх активів позики.
2	Обчислити $\rho_{cd}(W)$, $q_{cd}^{(\alpha)}(W)$ та $\sigma_{cd}^{rel}(W)$ для кожної потенційної пари.
3	Побудувати множину $\mathcal{C}_d^{adm}$ для кожного активу позики.
4	Для допустимих пар обчислити a_{cd}, ϕ_{cd}, LTV_{cd}^{eff}, LT_{cd}^{eff}, R_{cd}^{ref} і поріг швидкого блокування γ_{cd}.
5	Сформувати батч оновлення **PairRegistry** з номером епохи.
6	Подати транзакцію batchUpdatePairs з мультипідпису, якому надано роль RISK_ADMIN, і зафіксувати подію оновлення параметрів пар.

Кожний елемент пакета відповідає одній парі (c, d) і містить уже виведені параметри ризику, а не проміжні статистичні показники. Ланцюговий контур отримує з нього лише те, що потрібно для перевірки нової позики, ліквідації та швидкого блокування: ознаку допустимості, межі ризику, опорне відношення цін і поріг миттєвого обмеження ризикових дій.

Табл. 3.5: Поля одного елемента добового пакета оновлення

Поле	Тип	Призначення
collateralAsset	address	Адреса заставного активу c.
debtAsset	address	Адреса активу позики d.
allowed	bool	Ознака того, що пара входить до множини $\mathcal{C}_d^{adm}$.
aCdBps	uint16	Коефіцієнт сумісності a_{cd}, поданий у базисних пунктах. Його виводять з $\rho_{cd}(W)$ і $q_{cd}^{(\alpha)}(W)$ або з $\sigma_{cd}^{rel}(W)$ і $q_{cd}^{(\alpha)}(W)$ для $d \in \mathcal{D}^{stb}$.
phiCdBps	uint16	Коефіцієнт зниження ϕ_{cd}, поданий у базисних пунктах.
effectiveLtvBps	uint16	Ефективна межа запозичення LTV_{cd}^{eff} для пари (c, d), подана у базисних пунктах.
effectiveLiquidationThresholdBps	uint16	Ефективний ліквідаційний поріг LT_{cd}^{eff} для пари (c, d), поданий у базисних пунктах.
referenceRatioRay	uint128	Опорне значення $R_{cd}^{ref} = \frac{P_c}{P_d}$, зафіксоване під час останнього добового оновлення у форматі *ray*.
guardThresholdBps	uint32	Поріг γ_{cd} відносного падіння $\frac{P_c}{P_d}$ у базисних пунктах, після якого швидке блокування забороняє ризикові дії.
updatedAt	uint64	Часова мітка формування пакета для конкретної пари.

Номер епохи не дублюється в кожному елементі пакета, а передається як спільний атрибут всього батча; після прийняття пакета його значення фіксується в записі PairConfig. Перед записом пакета контракт перевіряє п’ять інваріантів: для дозволеної пари мають бути заповнені всі поля конфігурації; для недозволеної пари значення a_{cd}, ϕ_{cd}, LTV_{cd}^{eff}, LT_{cd}^{eff}, R_{cd}^{ref} і γ_{cd} мають дорівнювати нулю; повинна виконуватися нерівність $LTV_{cd}^{eff} < LT_{cd}^{eff}$; коефіцієнти a_{cd}, ϕ_{cd}, LTV_{cd}^{eff}, LT_{cd}^{eff} і γ_{cd} не можуть виходити за межі дозволених діапазонів; новий пакет приймається лише з номером епохи, більшим за номер попереднього пакета.

3.3.2 Відкриття позики з ланцюговим швидким блокуванням

Логіка позики починається не з переказу активу, а з моделювання нового стану. Контракт спочатку перевіряє, чи входить пара (c, d) до поточної множини $\mathcal{C}_d^{adm}$, чи свіжі цінові дані та чи не спрацювало швидке блокування. Лише після цього він оновлює борг через глобальний індекс, моделює новий стан позиції й перевіряє HF_{cd} після операції.

Швидке блокування працює на основі поточної відносної ціни

$$R_{cd}^{now} = \frac{P_c^{now}}{P_d^{now}}. \tag{3.2}$$

Контракт порівнює її з R_{cd}^{ref}, що був зафіксований під час останнього добового оновлення. Нехай γ_{cd} є порогом швидкого блокування, записаним у полі guardThresholdBps. Блокування спрацьовує за умови

$$R_{cd}^{now} \leq R_{cd}^{ref} \cdot (1 - \gamma_{cd}). \tag{3.3}$$

Якщо ця умова виконується, система відхиляє нову позику й небезпечне виведення застави навіть тоді, коли добове переоцінювання ще не встигло виключити пару з $\mathcal{C}_d^{adm}$. Завдяки цьому різке падіння активу на десятки відсотків не лишає платформу беззахисною до наступної доби.

Послідовність перевірок під час відкриття позики наведено на рис. 3.2.

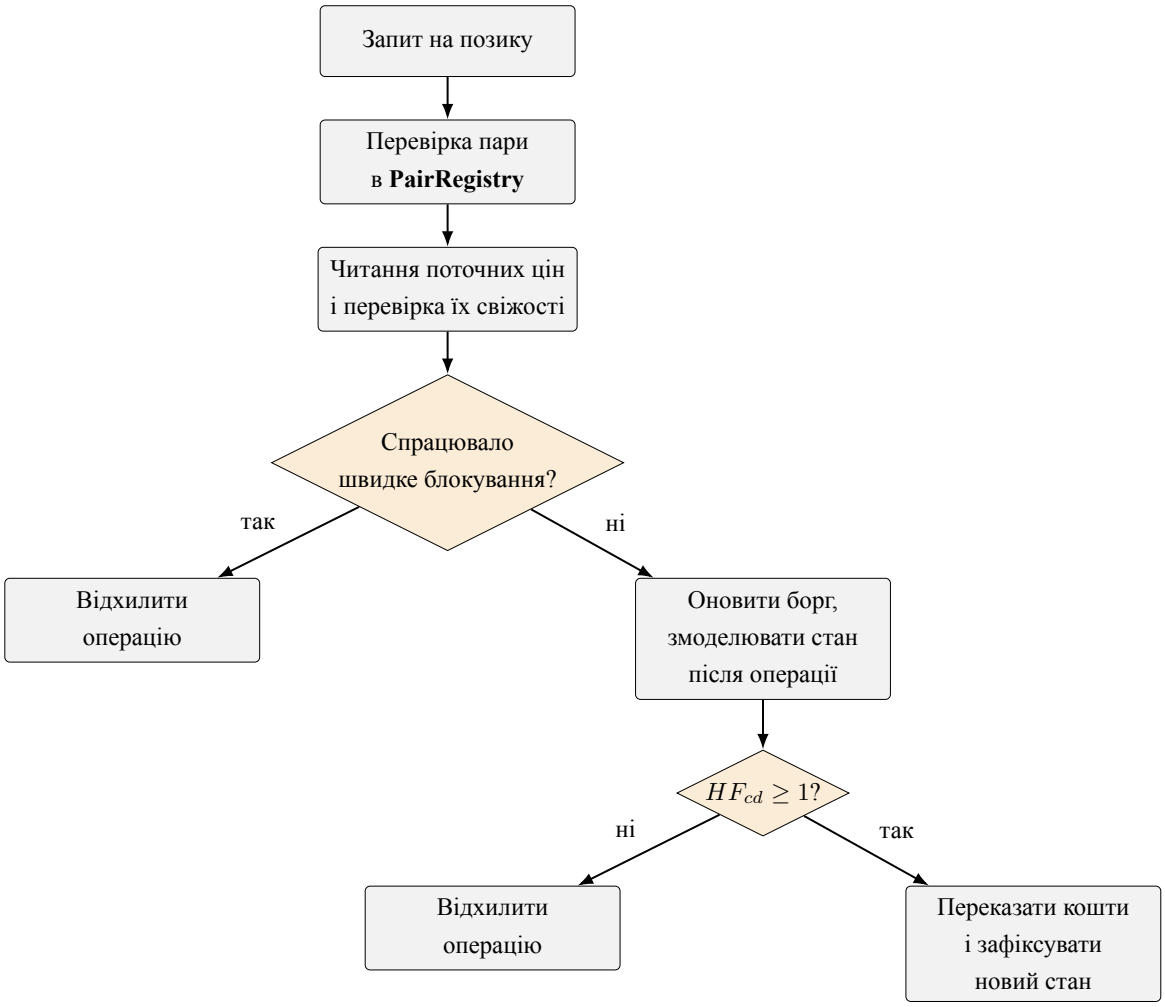

Рис. 3.2: Послідовність допуску нової позики

3.3.3 Погашення, виведення застави та ліквідація

Погашення боргу не блокується в кризовому режимі. Якщо швидке блокування вже спрацювало, система все одно дозволяє користувачеві погасити частину або весь борг і внести додаткову заставу. Швидке блокування забороняє лише дві ризикові дії: відкриття нової позики та виведення застави. Виведення застави — це ризикова дія, тому вона перевіряється за поточними цінами, станом швидкого блокування для конкретної пари і післяопераційним HF_{cd}. Натомість repay, depositCollateral і liquidate лишаються доступними навіть після спрацювання цього механізму.

Ліквідація спирається лише на ланцюгові дані й не чекає добового оновлення. Як тільки поточна ціна переводить позицію в стан $HF_{cd} < 1$, ліквідатор може погасити частину боргу в межах коефіцієнта закриття, а контракт визначає обсяг застави, яку потрібно вилучити з урахуванням бонусу ліквідатора. Це

дає системі безперервний зовнішній механізм відновлення платоспроможності позицій.

3.4 Інтеграція з децентралізованим застосунком і безсерверний прикладний рівень

Фронтенд реалізовано як статичний децентралізований застосунок, який не потребує постійного власного бекенду на шляху користувацької транзакції. Він читає стан позиції через RPC, показує користувачу допустимі пари, межу позики, очікуваний HF_{cd} після дії, стан швидкого блокування, дату останнього оновлення пари та поточну ставку. Запис у блокчейн відбувається лише через гаманець користувача.

Добовий позаланцюговий контур працює окремо від децентралізованого застосунку і не бере участі в конкретній користувацькій транзакції. Інтерфейс лише читає вже записаний ланцюговий стан і пояснює користувачеві, чому операцію дозволено або відхилено.

Табл. 3.6: Дані, які децентралізований застосунок відображає без зміни стану

Показник	Призначення
Допустимі заставні активи для вибраного активу позики	Пояснення, які пари зараз підтримуються системою.
LTV_{cd}^{eff}, LT_{cd}^{eff}	Межа позики та ліквідаційний поріг для поточної пари.
Очікуваний HF_{cd} після операції	Попередня оцінка безпечності нової транзакції.
Час останнього оновлення пари	Показ свіжості пакетного оновлення **PairRegistry**.
Стан швидкого блокування	Пояснення, чи не потрапила пара під тимчасове блокування через різкий ціновий рух.
Поточна ставка та використання пулу	Пояснення вартості боргу і напруження ліквідності ринку.

Для інтерфейсу принциповими є й точні причини відмови. Замість загального повідомлення про невдалу транзакцію контракт повертає окремі іменовані помилки: недопустима пара, застаріла ціна, спрацювання швидкого блокування, перевищення межі позики, брак ліквідності або відсутність повноважень [29]. Таке рішення спрощує і користувацький інтерфейс, і тестування.

3.5 Організація тестування та перевірки

Перевірку платформи побудовано у два шари. Перший охоплює ланцюгові тести смартконтрактів, другий — добовий позаланцюговий контур форму-

вання пакета для **PairRegistry**. Це розділення потрібне, бо смартконтракт може помилково пропустити небезпечну операцію, а пакетний скрипт — некоректно сформувати набір допустимих пар.

Табл. 3.7: Шари тестування практичної реалізації

Шар перевірки	Що перевіряється
Модульні тести ланцюгового ядра	Масштабований борг, глобальний індекс, відсоткова крива, свіжість цін, показник платоспроможності, коефіцієнт закриття, аварійна пауза та рольовий доступ.
Модульні тести **PairRegistry** і **RiskEngine**	Допуск лише пар з $\mathcal{C}_d^{adm}$, коректне застосування LTV_{cd}^{eff}, LT_{cd}^{eff}, спрацювання швидкого блокування при різкому падінні $\frac{P_c}{P_d}$.
Інтеграційні тести	Повний життєвий цикл позиції: депозит ліквідності, внесення застави, позика, погашення, виведення застави, ліквідація після падіння ціни.
Тести добового контуру	Детермінованість розрахунку $\rho_{cd}(W)$, $q_{cd}^{(\alpha)}(W)$, a_{cd}, ϕ_{cd} та коректність пакетного оновлення **PairRegistry**.
Операційні тести децентралізованого застосунку	Коректне відображення допустимих пар, причин відхилення транзакції, стану швидкого блокування і меж ризику для поточної пари.

Табл. 3.8: Ключові сценарії перевірки реалізації

Сценарій	Очікуваний результат
Відкриття позики по дозволеній парі	Позика проходить лише тоді, коли пара входить до $\mathcal{C}_d^{adm}$, ціна свіжа, швидке блокування не спрацювало, а HF_{cd} після операції лишається допустимим.
Відмова в позиці по забороненій парі	Контракт відхиляє виклик borrow без зміни стану.
Відмова через застарілу ціну	Нова позика, ліквідація або ризикове виведення застави відхиляються.
Спрацювання швидкого блокування	borrow і withdrawCollateral блокуються, тоді як repay, depositCollateral і liquidate лишаються доступними.
Часткова ліквідація	Ліквідація зменшує борг лише в дозволених межах і коректно переводить частину застави ліквідатору.
Пакетне оновлення **PairRegistry**	Новий пакет приймається лише від мультипідпису з роллю RISK_ADMIN, а значення epoch збільшується монотонно.
Відхилення суперечливого пакета	Оновлення не приймається, якщо $LTV_{cd}^{eff} \geq LT_{cd}^{eff}$ або порушено інші інваріанти.

Сценарна перевірка охоплює не лише позитивні стани. Окремо відтворюються ситуації, коли пара ще лишається допустимою в **PairRegistry**, але поточна відносна ціна вже вийшла за поріг швидкого блокування; коли добове оновлення не відбулося вчасно; коли користувач намагається зменшити заставу після погіршення відносної ціни; коли ліквідатор виконує часткове пога-

шення в межах коефіцієнта закриття; коли пакет оновлення від мультипідпису RISK_ADMIN містить внутрішньо суперечливу конфігурацію.

Для ланцюгової частини збережено середовище Foundry / Forge [30]. Воно придатне для сценарних тестів, часових зсувів, моделювання ролей і перевірки помилок доступу. Позаланцюговий контур перевіряється окремим тестовим набором, який використовує фіксовані індексовані історичні дані цін і перевіряє відтворюваність сформованих пакетів параметрів.

3.6 Стійкість реалізації до типових інцидентів

Практична реалізація захищає критичні місця, у яких кредитна платформа втрачає стійкість. Перший клас інцидентів пов'язаний з ціною. Якщо дані **Chainlink Data Feeds** застаріли, система не допускає нову позику, небезпечне виведення застави і ліквідацію. Якщо дані свіжі, але поточне відношення $\frac{P_c}{P_d}$ різко погіршилося між двома добовими оновленнями, спрацьовує швидке блокування.

Другий клас інцидентів пов'язаний з ліквідністю. Відсоткова модель реагує на використання пулу та швидкість його зростання, а модуль вільної ліквідності не дозволяє виводити актив позики, якщо це руйнувало б платоспроможність ринку. Третій клас інцидентів пов'язаний з адміністративним шаром. Зміну базових параметрів ризику, паузу, оновлення списку дозволених публічних фідів і пакетне оновлення параметрів пар рознесено між ролями PARAM_ADMIN та RISK_ADMIN, які надано мультипідпису. Це зменшує наслідки компрометації одного ключа [27].

Табл. 3.9: Типові інциденти та реакція реалізації

Інцидент	Потенційний наслідок	Реакція системи
Застаріла ціна	Хибна позика або хибна ліквідація	Перевірка updatedAt у **Chainlink Data Feeds** і відхилення ризикових дій [25].
Різке падіння відносної ціни пари	Тимчасова небезпечність нових позик до наступного добового оновлення	Ланцюгове швидке блокування забороняє лише borrow і withdrawCollateral; repay, depositCollateral і liquidate лишаються доступними.
Повторний вхід під час токен-переказу	Неконсистентний стан балансу або подвійне списання	Критичні функції захищено шаблоном *checks-effects-interactions* і блокуванням повторного входу [26, 28].
Некоректна конфігурація пари або ринку	Встановлення суперечливих меж ризику	Контракт відхиляє пакет, у якому $LTV_{cd}^{eff} \geq LT_{cd}^{eff}$ або порушено інші інваріанти.
Відмова добового скрипту	Застосування застарілого пакета параметрів	Система лишається працездатною з останнім затвердженим пакетом, а швидке блокування захищає від очевидного шоку.

3.7 Критерії практичної придатності

Реалізація вважається практично придатною, якщо одночасно виконує п’ять умов. Смартконтракт не допускає позику або виведення застави, які порушують межі ризику для конкретної пари. Добовий контур відтворювано формує однакові параметри для однакових індексованих історичних даних. Різкий ринковий шок блокує нові ризикові дії без очікування наступного циклу переоцінювання.

Погашення боргу й внесення додаткової застави лишаються доступними навіть у кризовому режимі, а інтерфейс показує не лише поточний борг, а й причину обмеження нової дії. Критерії практичної придатності безпосередньо пов’язують запропонований метод з реалізацією: автоматичний добір допустимих пар і виведення параметрів ризику для конкретної пари перенесено в код, перевірено сценаріями і використано в робочому контурі протоколу.

Висновки до розділу 3

У розділі подано практичну реалізацію платформи для децентралізованого кредитування, побудовану відповідно до аналітичної та математичної моделі попередніх розділів. Ланцюгове ядро реалізовано у вигляді одного основного контракту протоколу та окремого **PairRegistry**. Реалізація охоплює межі ризику для конкретної пари, масштабований борг, ліквідацію і швидке блокування небезпечних дій за різкого погіршення відносної ціни пари.

Окремо визначено добовий контур переоцінювання на основі поточних цін **Chainlink Data Feeds** та індексованих історичних даних їхніх відповідей. Пакетне оновлення **PairRegistry** виконується транзакцією batchUpdatePairs з мультипідпису, якому надано роль RISK_ADMIN. Описано також безсерверний децентралізований застосунок, структури стану, алгоритми критичних операцій і програму тестування.

Розділ показує узгоджену роботу двох контурів системи: ланцюгового, який працює лише з готовими параметрами пари, і позаланцюгового, який добово переобчислює ці параметри на індексованих історичних даних. Перевірка охоплює модульні, інтеграційні та сценарні тести, а також окремий тестовий набір для добового пакета переоцінювання. Це підтверджує практичну здійсненність автоматичного добору застави, автоматичного виведення меж ризику та швидкого блокування небезпечних дій у смартконтрактному протоколі.

Окремо показано, що добове переоцінювання і швидке блокування виконують різні функції та не дублюють одне одного. Добовий контур оновлює склад допустимих пар і межі ризику, а ланцюговий захист перекриває проміжок між двома циклами переоцінювання. У такий спосіб практична реалізація прямо відтворює математичну модель другого розділу на рівні структури стану, перевірок транзакції та сценаріїв тестування.

ВИСНОВКИ

У роботі розв’язано задачу проєктування та реалізації платформи для децентралізованого кредитування з автоматизованим управлінням ризиками на базі смартконтрактів. Аналіз предметної області показав, що стійкість такого протоколу визначається кредитним і ліквіднісним ризиками, коректністю цінових даних та добором пари «застава – актив позики». Тому архітектуру платформи побудовано навколо окремого шару оцінювання ризику.

У межах роботи сформовано вимоги до системи, запропоновано модульну архітектуру протоколу, визначено рольову модель учасників і формалізовано критичні сценарії: внесення ліквідності, відкриття позики, погашення боргу, виведення застави та ліквідацію. Побудовано математичну модель оцінювання позиції через автоматичний добір допустимих заставних активів для заданого активу позики, окреме правило для ринків з низьковолатильним активом позики, коефіцієнт сумісності пари, ефективний LTV, ефективний ліквідаційний поріг, похідну ціну ліквідації та показник платоспроможності. Введено також перевірку актуальності цінових даних за часовою міткою відповіді оракула.

Запропоновано кусочно-лінійну відсоткову модель, що реагує на рівень використання пулу ліквідності та швидкість його зростання, і сформульовано правила ліквідації з урахуванням коефіцієнта закриття та бонусу ліквідатора. Технічну стійкість протоколу забезпечують захист від повторного входу, аварійна пауза, рольовий доступ і журналювання адміністративних змін.

У підсумку реалізовано смартконтрактну платформу для окремих пар активів з основним контрактом протоколу, окремим **PairRegistry**, швидким блокуванням небезпечних дій і безсерверним прикладним рівнем.

Добовий контур переоцінювання спирається на поточні відповіді публічних цінових фідів Chainlink та їх індексовану історію. Він формує параметри пари, записує їх до окремого реєстру допустимих пар і запускається через мультипідпис керування ризиком. Реалізацію перевірено модульними, інтеграційними, сценарними й позаланцюговими тестами; подальший розвиток охоплює інваріантні тести, зниження gas-витрат, розширення набору активів і зовнішній аудит.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

[1] Aave. Introduction to Aave. URL: https://aave.com/help/aave-101/introduction-to-aave (дата звернення: 03.04.2026).

[2] Aave. Health Factor & Liquidations. URL: https://aave.com/help/borrowing/liquidations (дата звернення: 03.04.2026).

[3] Aave. Efficiency Mode (E-mode). URL: https://aave.com/help/borrowing/e-mode (дата звернення: 07.04.2026).

[4] Aave. Isolation Mode. URL: https://aave.com/help/supplying/isolation-mode (дата звернення: 07.04.2026).

[5] Aave Protocol. ReserveConfiguration.sol. Release v1.19.4. URL: https://raw.githubusercontent.com/aave/aave-v3-core/b74526a7bc67a3a117a1963fc871b3eb8cea8435/contracts/protocol/libraries/configuration/ReserveConfiguration.sol (дата звернення: 08.04.2026).

[6] Leshner R., Hayes G. Compound: The Money Market Protocol. Version 1.0. February 2019. URL: https://compound.finance/documents/Compound.Whitepaper.pdf (дата звернення: 03.04.2026).

[7] Compound Labs. Compound v2 Docs: cTokens. URL: https://docs.compound.finance/v2/ctokens (дата звернення: 07.04.2026).

[8] Compound Labs. Compound v2 Docs: Comptroller. URL: https://docs.compound.finance/v2/comptroller (дата звернення: 03.04.2026).

[9] Compound Labs. Compound v2 Documentation. Protocol Math and interest accrual overview. URL: https://docs.compound.finance/v2/ (дата звернення: 03.04.2026).

[10] Compound Labs. Compound III Docs: Collateral & Borrowing. URL: https://docs.compound.finance/collateral-and-borrowing/ (дата звернення: 07.04.2026).

[11] Compound Labs. Compound III Docs: Liquidation. URL: https://docs.compound.finance/liquidation/ (дата звернення: 07.04.2026).

[12] Compound Labs. Compound III Docs: Interest Rates. URL: https://docs.compound.finance/interest-rates/ (дата звернення: 07.04.2026).

[13] Compound Finance. CometConfiguration.sol. Revision 71aec025. URL: https://raw.githubusercontent.com/compound-finance/comet/71aec025e1fcf9c16a1aeb20a85eddae5a5d5581/contracts/CometConfiguration.sol (дата звернення: 08.04.2026).

[14] Morpho. Introduction. URL: https://legacy.docs.morpho.org/overview/ (дата звернення: 07.04.2026).

[15] Morpho. Morpho Optimizers [Deprecated]. URL: https://legacy.docs.morpho.org/morpho-optimizers/concepts/introduction/ (дата звернення: 07.04.2026).

[16] Morpho. Morpho Market V1. URL: https://docs.morpho.org/learn/concepts/market/ (дата звернення: 10.04.2026).

[17] Morpho. Interest Rate Model. URL: https://docs.morpho.org/learn/concepts/irm/ (дата звернення: 10.04.2026).

[18] Morpho. Morpho Vault V2. URL: https://docs.morpho.org/learn/concepts/vault-v2/ (дата звернення: 10.04.2026).

[19] Morpho. Roles and Capabilities. URL: https://docs.morpho.org/curate/concepts/roles/ (дата звернення: 10.04.2026).

[20] Morpho Blue. IMorpho.sol. Release v1.0.0. URL: https://raw.githubusercontent.com/morpho-org/morpho-blue/55d2d99304fb3fb930c688462ae2ccabb1d533ad/src/interfaces/IMorpho.sol (дата звернення: 08.04.2026).

[21] Fluid. Contracts Overview. URL: https://docs.fluid.io/ (дата звернення: 07.04.2026).

[22] Instadapp. Fluid technical docs (docs.md). URL: https://raw.githubusercontent.com/Instadapp/fluid-contracts-public/main/docs/docs.md (дата звернення: 08.04.2026).

[23] Instadapp. fluid-contracts-public. URL: https://github.com/Instadapp/fluid-contracts-public (дата звернення: 08.04.2026).

[24] Fluid. Fluid DEX V2: Liquidity Source Integration. URL: https://docs.fluid.io/integrate/dex-v2-swaps.html (дата звернення: 07.04.2026).

[25] Chainlink Foundation. Chainlink Data Feeds Documentation. URL: https://docs.chain.link/data-feeds (дата звернення: 03.04.2026).

[26] OpenZeppelin. OpenZeppelin Contracts: Utils (Pausable, ReentrancyGuard). URL: https://docs.openzeppelin.com/contracts/5.x/api/utils (дата звернення: 10.04.2026).

[27] OpenZeppelin. OpenZeppelin Contracts: Access Control. URL: https://docs.openzeppelin.com/contracts/5.x/access-control (дата звернення: 03.04.2026).

[28] Solidity Team. Security Considerations. URL: https://docs.soliditylang.org/en/v0.8.30/security-considerations.html (дата звернення: 03.04.2026).

[29] Solidity Team. Expressions and Control Structures: revert and custom errors. URL: https://docs.soliditylang.org/en/latest/control-structures.html (дата звернення: 04.04.2026).

[30] Foundry. Forge Test Reference. URL: https://getfoundry.sh/forge/reference/test/ (дата звернення: 04.04.2026).

[31] McNeil A.J., Frey R., Embrechts P. Quantitative Risk Management: Concepts, Techniques and Tools. Revised edition. Princeton University Press, 2015. URL: https://books.google.com/books/about/Quantitative_Risk_Management.html?id=REQLogEACAAJ (дата звернення: 11.04.2026).

[32] Jorion P. Value at Risk: The New Benchmark for Managing Financial Risk. 3rd ed. McGraw-Hill, 2007. URL: https://books.google.com/books/about/Value_at_Risk.html?id=FJRBPgAACAAJ (дата звернення: 11.04.2026).

[33] Hull J.C. Risk Management and Financial Institutions. 4th ed. John Wiley & Sons, 2015. URL: https://books.google.com/books/about/Risk_Management_and_Financial_Institutio.html?id=uzikBgAAQBAJ (дата звернення: 11.04.2026).

[34] Bangia A., Diebold F.X., Schuermann T., Stroughair J.D. Modeling Liquidity Risk, with Implications for Traditional Market Risk Measurement and Management. In: Figlewski S., Levich R.M. (eds.) Risk Management: The State of the Art. Springer, Boston, MA, 2001. P. 3–13. DOI: https://doi.org/10.1007/978-1-4615-0791-8_1 (дата звернення: 11.04.2026).

ДОДАТОК А

ЛІСТИНГ СТРУКТУР СТАНУ ТА ЗОВНІШНІХ ВИКЛИКІВ

Лістинг А.1. Скорочені структури стану основного контракту

```
pragma solidity ^0.8.24;
struct AssetConfig {
    address token;
    address priceFeed;
    uint32 staleAfter;
    uint16 ltvBaseBps;
    uint16 liquidationThresholdBaseBps;
    bool enabled;
}
struct MarketState {
    uint128 totalLiquidityShares;
    uint128 totalScaledDebt;
    uint128 borrowIndexRay;
    uint128 reserveBalance;
    uint64 lastAccrualTimestamp;
    uint16 lastUtilizationBps;
    uint16 optimalUtilizationBps;
    uint16 slope1Bps;
    uint16 slope2Bps;
    bool paused;
}
struct Position {
    uint128 collateralAmount;
    uint128 scaledDebt;
    address collateralAsset;
    address debtAsset;
    uint64 openedAt;
    uint64 updatedAt;
}
```

Лістинг А.2. Скорочений фрагмент PairRegistry і зовнішнього інтерфейсу

```
struct PairUpdate {
    address collateralAsset;
    address debtAsset;
    bool allowed;
```

```
    uint16 aCdBps;
    uint16 phiCdBps;
    uint16 effectiveLtvBps;
    uint16 effectiveLiquidationThresholdBps;
    uint64 updatedAt;
    uint128 referenceRatioRay;
    uint32 guardThresholdBps;
}
struct PairConfig {
    bool allowed;
    uint16 aCdBps;
    uint16 phiCdBps;
    uint16 effectiveLtvBps;
    uint16 effectiveLiquidationThresholdBps;
    uint64 updatedAt;
    uint64 epoch;
    uint128 referenceRatioRay;
    uint32 guardThresholdBps;
}
interface IPairRegistry {
    function getPairConfig(address collateralAsset, address debtAsset) external view returns (
      PairConfig memory);
    function batchUpdatePairs(PairUpdate[] calldata updates, uint64 epoch) external;
}
interface ILendingProtocol {
    function supplyLiquidity(address debtAsset, uint256 amount) external;
    function withdrawLiquidity(address debtAsset, uint256 shareAmount) external;
    function depositCollateral(address collateralAsset, address debtAsset, uint256 amount)
      external;
    function borrow(address collateralAsset, address debtAsset, uint256 amount) external;
    function repay(address collateralAsset, address debtAsset, uint256 amount) external;
    function withdrawCollateral(address collateralAsset, address debtAsset, uint256 amount)
      external;
    function liquidate(address borrower, address collateralAsset, address debtAsset, uint256
      repayAmount) external;
}
```

ДОДАТОК Б

ЛІСТИНГ ДОБОВОГО ПАКЕТА ОНОВЛЕННЯ PAIRREGISTRY

Лістинг Б.1. Скорочений фрагмент добового скрипту переоцінювання пар

```
type PairUpdate = {
  collateralAsset: string; debtAsset: string; allowed: boolean;
  aCdBps: bigint; phiCdBps: bigint; effectiveLtvBps: bigint;
  effectiveLiquidationThresholdBps: bigint; referenceRatioRay: bigint;
  guardThresholdBps: bigint; updatedAt: bigint;
};
async function rebuildPairs(epoch: bigint): Promise<void> {
  const updates: PairUpdate[] = [];
  for (const debtAsset of debtAssets) {
    for (const collateralAsset of collateralCandidates) {
      const series = await loadAlignedSeries(collateralAsset, debtAsset);
      const rho = computeCorrelation(series.collateral, series.debt);
      const qAlpha = computeLowerQuantile(series.relativeReturns, 0.05);
      const relativeVolatility = computeRelativeVolatility(series.relativeReturns);
      const stableDebt = isLowVolatilityDebtAsset(debtAsset);
      const allowed = isAdmissible({ rho, qAlpha, relativeVolatility, stableDebt });
      const aCdBps = allowed ? computeACdBps({ rho, qAlpha, relativeVolatility, stableDebt }) : 0n;
      const phiCdBps = allowed ? computePhiCdBps(aCdBps, kappaMinBps) : 0n;
      const baseLtvBpsForPair = baseLtvBps[collateralAsset];
      const baseLtBpsForPair = baseLtBps[collateralAsset];
      const baseBufferBps = baseLtBpsForPair − baseLtvBpsForPair;
      const effectiveLtvBps = allowed ? applyBps(baseLtvBpsForPair, phiCdBps) : 0n;
      const effectiveLiquidationThresholdBps = allowed
        ? minBps(baseLtBpsForPair, effectiveLtvBps + baseBufferBps) : 0n;
      updates.push({
        collateralAsset, debtAsset, allowed, aCdBps, phiCdBps,
        effectiveLtvBps, effectiveLiquidationThresholdBps,
        referenceRatioRay: allowed ? currentReferenceRatio(collateralAsset, debtAsset) : 0n,
        guardThresholdBps: allowed ? computeGuardThreshold(qAlpha, minGuardBps, maxGuardBps, guardLambda) :
        0n,
        updatedAt: BigInt(nowUnix())
      });
    }
  }
  assertBatchInvariants(updates);
  const callData = pairRegistry.interface.encodeFunctionData("batchUpdatePairs", [updates, epoch]);
  await riskAdminMultisig.submit(pairRegistryAddress, callData);
}
```

ДОДАТОК В

ЛІСТИНГ СЦЕНАРІЮ ПЕРЕВІРКИ ШВИДКОГО БЛОКУВАННЯ

Лістинг В.1. Скорочений сценарний тест швидкого блокування

```
function test_guard_blocks_risky_actions_and_allows_recovery_actions() public {
    configureAllowedPair(weth, usdc, 7_000, 8_000);
    supplyLiquidity(usdc, 500_000e6);

    depositCollateral(alice, weth, usdc, 10 ether);
    borrowAs(alice, weth, usdc, 12_000e6);

    setReferenceRatio(weth, usdc, 3_000e18);
    setCurrentPrices(weth, 1_400e8, usdc, 1e8);

    vm.prank(alice);
    vm.expectRevert(GuardTriggered.selector);
    protocol.borrow(weth, usdc, 100e6);

    vm.prank(alice);
    vm.expectRevert(GuardTriggered.selector);
    protocol.withdrawCollateral(weth, usdc, 1 ether);

    liquidateAs(liquidator, alice, weth, usdc, 1_000e6);

    vm.prank(alice);
    protocol.repay(weth, usdc, 2_000e6);

    vm.prank(alice);
    protocol.depositCollateral(weth, usdc, 1 ether);
}
```