# Мережа Axelar:

# З'єднання додатків з блокчейн екосистемами

Проект 1.0
січень 2021 р.

Анотація

З'являються численні блокчейн-екосистеми, які надають унікальні та відмінні функції, привабливі для користувачів та розробників додатків. Проте зв'язок між екосистемами дуже рідкісний і фрагментарний. Щоб програми могли безперешкодно взаємодіяти між екосистемами блокчейна, ми пропонуємо Axelar. Стек Axelar надає децентралізовану мережу, протоколи, інструменти та API, які забезпечують простий міжмережевий зв'язок. Набір протоколів Axelar складається з протоколів транскордонної маршрутизації та передачі. Децентралізована відкрита мережа валідаторів живить мережу; будь-хто може приєднатися, використовувати його та брати участь. Візантійський консенсус, криптографія та механізми заохочення призначені для досягнення високих вимог до безпеки та живучості, що є унікальними для міжмережевих запитів.

## 1. Введення

Системи блокчейн швидко набирають популярності та залучають нові варіанти використання для токенізації активів, децентралізованих фінансів та інших розподілених додатків. Декілька основних платформ, таких як Ethereum, Monero, EOS, Cardano, Terra, Cosmos, Avalanche, Algorand, Near, Celo та Polkadot, пропонують різні функції та середовища розробки, які роблять їх привабливими для різних додатків, варіантів використання та кінцевих користувачів.[5, 11, 4, 21, 20, 23, 24 19, 6, 14, 25]. Проте корисні функції кожною новою платформи в даний час пропонуються менш ніж 1% користувачів екосистеми, а саме власникам власного токену на цій платформі. Чи можемо ми дозволити розробникам платформ легко підключати свої блокчейни до інших екосистем? Чи можемо ми дозволити розробникам програм створювати найкращу платформу для своїх потреб, зберігаючи при цьому зв'язок між декількома блокчейн-екосистемами? Чи можемо ми дозволити користувачам взаємодіяти з будь-яким додатком у будь-якому ланцюжку блоків прямо з їхніх гаманців?

Щоб з'єднати екосистеми блокчейна і дозволити програмам безперешкодно взаємодіяти між ними, ми пропонуємо мережу Axelar. Валідатори колективно запускають протокол візантійського консенсусу та запускають протоколи, що полегшують міжмережеві запити. Будь-хто може приєднатися до мережі, брати участь та використовувати її. Базова мережа оптимізована для високих вимог до безпеки та живучості, унікальних для запитів між ланцюжками. Мережа Axelar також включає набір протоколів та API. Основні протоколи:

- Протокол міжмережевого шлюзу (CGP). Цей протокол аналогічний до протоколу прикордонного шлюзу в Інтернеті. Цей протокол використовується для з'єднання декількох автономних блокчейн-екосистем та відповідає за маршрутизацію між ними. Блокчейнам не потрібно «говорити якоюсь спеціальною мовою», розробникам їх платформ не потрібно вносити будь-які зміни в свої ланцюжки, і їх ланцюжки можна легко підключити до глобальної мережі.

- Протокол міжмережевої передачі (CTP). Цей протокол аналогічний протоколам прикладного рівня File Transfer, Hypertext Transfer

Protocols в Інтернеті. Це стек протоколів рівня програми, що знаходиться поверх протоколів маршрутизації (таких як CGP та інші технології маршрутизації). Розробники програм можуть підключати свої децентралізовані програми до будь-якого ланцюжка для виконання запитів між ланцюжками. Користувачі можуть використовувати протокол CTP для взаємодії з програмами в будь-якому ланцюжку, використовуючи прості виклики API, аналогічні запитам HTTP GET/POST. Розробники можуть блокувати, розблокувати та передавати активи між будь-якими двома адресами на будь-яких платформах блокчейна, запускати тригери міжмережевих додатків (наприклад, децентралізовані додатки в ланцюжку A, можуть оновлювати його стан, якщо будь-яка інша програма в ланцюжку B задовольняє деяким критеріям пошуку (відсоткова ставка > X) виконувати спільні міжмережові запити між додатками в різних ланцюжках (смарт-контракт у ланцюжку A може викликати оновлення стану смарт-контракту в ланцюжку B). Цей протокол забезпечує можливість компонування програм у блокчейн-екосистемах.

Мережа Axelar пропонує такі переваги:

- *Для розробників блокчейн-платформ*: можливість легко підключати свої блокчейни до всіх інших блокчейн-екосистем. Для підключення до мережі необхідно налаштувати лише граничний обліковий запис у ланцюжку.

- *Для розробників децентралізованих додатків*: Розробники додатків можуть розміщувати свої децентралізовані програми де завгодно, блокувати,

розблокувати, передавати активи та взаємодіяти з програмами в будь-якому іншому ланцюжку через CTP API.

- *Для користувачів*: Користувачі можуть взаємодіяти з усіма додатками в екосистемі прямо зі своїх гаманців.

**Платформа для творців.** Нарешті, мережа Axelar – це платформа для розробників та глобальної спільноти. Його модель управління відкрита всім. Розробники можуть пропонувати нові точки інтеграції, маршрутизацію та протоколи рівня додатків, а користувачі можуть вирішувати, чи приймати їх шляхом голосування за пропозиціями, і, у разі схвалення, валідатори ухвалять зміни.

## 1.1 Існуючі рішення щодо функціональної сумісності

Попередні спроби вирішити проблему взаємодії між блокчейнами відносяться до однієї з чотирьох категорій: централізовані біржі, інтероперабельні екосистеми, упаковані активи та токен-мости. Нижче ми коротко підсумовуємо ці підходи.

**Централізовані системи**. Сьогодні централізовані системи є єдиними по-справжньому масштабованими рішеннями для функціональної сумісності.

потреби екосистеми. Вони можуть легко перерахувати будь-який актив або підключити будь-яку платформу. Тим не менш,

централізовані системи, як відомо, мають різні проблеми з безпекою і недостатньо хороші для підтримки децентралізованої фінансової системи, що формується, яка вимагає надійної безпеки, прозорості та відкритого управління. Самі собою вони можуть управляти децентралізованими додатками у міру їх зростання.

**Центри взаємодії.** Такі проекти, як Cosmos, Polkadot, Avalabs, спрямовані на забезпечення взаємодії між бічні ланцюги рідні для їх екосистем, що використовують протоколи, що настроюються. міжмережевого зв'язку23, 25, 24]. Наприклад, можна розкрутити сайдчейн (Cosmos Zone), який може взаємодіяти з Cosmos Hub. Сайдчейн має бути заснований на консенсусі Tendermint та використовувати протокол, спочатку зрозумілий Cosmos Hub. Підключення до інших блокчейнів та екосистем, що говорять різними мовами, залишається за зовнішніми технологіями.

**Парні мости.**Обгорнуті активи (наприклад, обгорнуті біткойни) намагаються заповнити недостатній пробіл у міжмережевій сумісності в екосистемі. Одним із прикладів є tBTC [9], Котрий є спеціальним протоколом, в якому для захисту перекладів використовується розумна комбінація смарт-контрактів та забезпечення. Ці рішення вимагають значних інженерних зусиль для створення — для кожної пари ланцюжків розробники повинні створити новий смарт-контракт у ланцюжку призначення, який аналізує докази стану з вихідного ланцюжка (аналогічно тому, як кожен бічний ланцюжок може, в принципі, аналізувати стан інших ланцюжків). . З використанням цього підходу було розгорнуто лише кілька мостів. Ці підходи не масштабуються, коли один із базових блокчейнів хоче оновити свої правила консенсусу чи формат транзакцій. Це пов'язано з тим, що всі смарт-контракти, які залежать від цих ланцюжків, повинні бути оновлені.

Ми також бачили кілька інших одноцільових мостів від розробників платформ, які листують логіку переходу стану в смарт-контрактах, щоб з'єднуватися з іншими екосистемами.1, 7]. Вони страждають від численних проблем масштабування, не дозволяють екосистемі масштабуватися рівномірно і вводять додаткові залежності для додатків. Наприклад, при зміні однієї платформи необхідно оновити всі смарт-контракти на всіх мостах. Це зрештою поставить екосистему в безвихідь, де ніхто не зможе оновитися. Нарешті, якщо один одноцільовий міст з'єднує платформи A і B, а другий одноцільовий міст з'єднує B і C, це не означає, що програми на A можуть взаємодіяти з додатками на C. Можливо, потрібно створити ще один одноцільовий міст. ціль мосту або перепрограмувати логіку програми.

Інші спроби вирішити питання сумісності включають федеративні оракули (наприклад, Ren [8]), та взаємодіючі блокчейни для конкретних додатків [10].

Підсумовуючи, можна сказати, що існуючі рішення для функціональної сумісності вимагають серйозної інженерної роботи як від розробників платформ, так і від розробників додатків, які повинні розуміти різні протоколи зв'язку між кожною парою екосистем. Таким чином, інтероперабельність практично відсутня у сьогоднішньому просторі блокчейну. Зрештою, розробники платформ хочуть зосередитись на створенні платформ та оптимізувати їх для своїх варіантів використання, а також мати можливість легко підключатися до інших блокчейнів. І розробники додатків хочуть створювати децентралізовані додатки на кращих платформах для своїх потреб, використовуючи користувачів, ліквідність і взаємодіючи з іншими децентралізованими додатками в інших ланцюжках.

# 2 У пошуках масштабованого міжмережевого зв'язку

По суті, кроссчейн-комунікація вимагає, щоб різноманітні мережі могли спілкуватися однією мовою. Щоб вирішити цю проблему, ми пояснимо набір протоколів Axelar, опишемо його високорівневі властивості та пояснимо, як ці властивості відносяться до ядра масштабованого міжмережевого зв'язку.

1. *“Інтеграція «підключи та працюй».* Від розробників платформи блокчейна не потрібно виконувати важку інженерну або інтеграційну роботу, щоб говорити якоюсь «мовою користувача» для підтримки кроссчейн. Кроссчейн-протокол повинен мати можливість безперешкодно підключати будь-який існуючий або новий блокчейн. Нові активи мають додаватися з мінімальними зусиллями.

2. *Кросчейн маршрутизація.*Такі функції, як виявлення мережевих адрес, шляхів маршрутизації та мереж, лежать в основі Інтернету і підтримуються BGP та іншими протоколами маршрутизації. Так само, щоб полегшити зв'язок між екосистемами блокчейна, нам необхідно підтримувати виявлення адрес у них, додатків та маршрутизацію.

3. *Підтримка оновлень.* Якщо одна з екосистем блокчейнів зміниться, це не вплине на сумісність інших блокчейнів. Система повинна розпізнавати оновлення, і для їх підтримки повинні бути потрібні мінімальні зусилля (тобто не слід переписувати «логіку переходу станів» і програми не повинні ламатися).

4. *Єдина мова для програм.* Програмам потрібен простий протокол для блокування, розблокування, передачі та зв'язку з іншими програмами, незалежно від того, в якому

ланцюжку вони знаходяться. Цей протокол повинен бути незалежним від ланцюжка та підтримувати прості виклики, аналогічні протоколам HTTP/HTTPS, які дозволяють користувачам та браузерам взаємодіяти з будь-яким веб-сервером. У міру того, як все більше мереж та активів приєднуються до протоколів маршрутизації нижчого рівня, програми повинні мати можливість використовувати їх для зв'язку без перезапису своїх програмних стеків.

Далі ми сумуємо вимоги безпеки, яким мають відповідати ці протоколи.

1. *Децентралізована довіра.* Мережа та протоколи мають бути децентралізованими, відкритими та дозволяти кожному брати участь.

2. *Висока безпека.* Система має задовольняти високі гарантії безпеки. Системі необхідно зберігати безпеку активів та стану у міру їх обробки кроссчейн-мережею.

3. *Висока живучість.* Система повинна задовольняти високі гарантії живучості, щоб підтримувати програми, що використовують її кроссчейн-функції.

Задовольнити підмножину цих властивостей легко. Наприклад, можна створити федеративний мультипідписний обліковий запис зі своїми друзями та заблокувати/розблокувати активи у відповідних ланцюжках. Такі системи за своєю природою уразливі для змови та атак цензури, і у валідаторів немає належних стимулів для їхнього захисту. Створення децентралізованої мережі та набору протоколів, у яких кожен може брати участь при правильному стимулюванні, може забезпечити безперешкодну міжмережеву комунікацію, але вирішення цієї складної проблеми потребує ретельного поєднання протоколів консенсусу, криптографії та проектування механізмів.

# 3 Мережа Axelar

*Мережа Axelar надає єдине рішення для кроссчейн-комунікацій, що відповідає потребам як розробників платформ – від них не вимагається жодних робіт з інтеграції, так і розробників додатків – один простий протокол та API для доступу до глобальної ліквідності та зв'язку з усією екосистемою.*

Мережа Axelar складається з децентралізованої мережі, яка об'єднує блокчейн-екосистеми, що розмовляють різними мовами, та набору протоколів з API-інтерфейсами поверх них, що дозволяє програмам легко виконувати міжмережеві запити. Мережа поєднує існуючі автономні блокчейни, такі як Bitcoin, Stellar, Terra, Algorand, та центри взаємодії, такі як рішення, такі як Cosmos, Avalanche, Ethereum та Polkadot. Наша місія полягає в тому, щоб дозволити розробникам додатків створювати такі програми простіше, використовуючи універсальний протокол та API, не розгортаючи свої пропрієтарні міжмережові протоколи або переписуючи програми з розробкою нових мостів.

Сцією метою ми розробили набір протоколів, який включає протокол міжмережевого шлюзу (див. розділ6) та протокол міжмережевої передачі (див. розділ 7).

Основним компонентом мережі є базові децентралізовані протоколи. Валідатори колективно підтримують мережу Axelar та запускають вузли, які захищають блокчейн Axelar. Вони обираються користувачами у процесі делегування. Валідатори одержують право голосу пропорційно до делегованої їм частки. Валідатори досягають консенсусу щодо стану кількох блокчейнів, яких підключена платформа. Блокчейн відповідає за підтримку та роботу міжмережевих протоколів маршрутизації та передачі. Правила керування дозволяють учасникам мережі приймати протокольні рішення, наприклад, які блокні з'єднувати і які активи підтримувати.

Блокчейн Axelar слідує моделі Delegated Proof-of-Stake (DPoS), аналогічної Cosmos Hub. Користувачі вибирають валідаторів, які мають зв'язати свою частку, щоб брати участь у консенсусі та підтримувати високу якість обслуговування. Модель DPoS дозволяє підтримувати великий набір децентралізованих валідаторів та надійні стимули, щоб гарантувати, що валідатори несуть відповідальність за підтримання мостів та часток схем криптографічних порогів. В рамках консенсусу валідатори запускають легке клієнтське програмне забезпечення інших блокчейнів, що дозволяє перевіряти стан інших блокчейнів.

Валідатори повідомляють про ці стани в блокчейн Axelar, і як тільки їх стає достатньо, стан біткойнів, ефіру і інших ланцюжків записується в Axelar.

Згодом базовий рівень Axelar знає про стан зовнішніх блокчейнів у будь-який момент часу, створюючи "вхідні мости" з інших блокчейнів. Валідатори колективно підтримують облікові записи з пороговим підписом в інших ланцюжках блоків (наприклад, 80% валідаторів повинні схвалювати та спільно підписувати будь-яку транзакцію з неї), що дозволяє їм блокувати та розблокувати активи та стан у ланцюжках, а також публікувати стан в інших ланцюжках блоків, «Вихідні мости». Загалом мережу Axelar можна розглядати як децентралізований перехресний оракул для читання/запису.

В решти документа описуються попередні відомості та будівельні блоки, що лежать у основі мережі (Розділ 4), деякі технічні деталі мережі (розділ 5), протокол міжмережевого шлюзу (розділ 6) та протокол міжмережевої передачі (розділ 7).

# 4 підготовчі заходи

## 4.1 Позначення та припущення

давайте позначимо Vr як набір валідаторів Axelar на раунді R. У кожного валідатора є вага, число в (0, 1) позначаючи право голосу цього конкретного валідатора. Для фіналізації блоків або для підпису міжмережевих запитів Axelar потрібні правильні валідатори загальної ваги > F. Ми називаємо параметр F*∈ [0.5, 1] поріг протоколу*.

Axelar може бути заснований на миттєва остаточність блокування Delegated-Proof-of-Stake. Валідатори працюють Візантійський відмовостійкий консенсус (BFT) у кожному раунді і завершити блокувати і-й. Як тільки блок завершено, виконується новий консенсус BFT для завершення і+1 блокувати тощо. Валідатори обираються шляхом делегування частки. Користувач з певною часткою може вибрати запуск вузла валідатора або делегувати своє право голосу (частку) існуючому валідатору, який голосує від його імені. Набір валідаторів можна оновлювати, валідатори приєднуються до набору або залишають його, а користувачі делегують або скасовують делегування свого права голосу.

Різні блокчейни працюють з різними припущеннями. Синхронний зв'язок означає, що існує фіксована верхня межа Δ часу доставки повідомлення, де Δ відомо і може бути вбудоване протокол. Асинхронний зв'язок означає, що доставка повідомлень може зайняти скільки завгодно багато часу, і відомо, що протоколи BFT не можуть бути побудовані для асинхронних мереж навіть за наявності лише одного зловмисного валідатора. Реалістичним компромісом між синхронією та асинхронією є припущення частково синхронного зв'язку. Мережа може бути повністю асинхронною до деякого невідомого часу глобальної стабілізації (GST), але після GST зв'язок стає синхронним з відомим верхнім кордоном Δ [17].

Типові блокчейни працюють у припущенні > F правильних валідаторів. Для синхронних мереж зазвичай встановлюється F = 1/2, але більш слабкого припущення про частково синхронної мережі F = 2/3. Біткойн, його форки та поточна Proof-of-Work версія Ethereum працюють лише за умови синхронності. Інші, такі як Algorand та Cosmos, вимагають лише часткової синхронізації. При з'єднанні ланцюжків через Axelar з'єднання працює, припускаючи найсильніші мережеві припущення цих ланцюжків, що є синхронністю, наприклад, у разі з'єднання Біткойна і Космосу. Сам блокчейн Axelar працює в частково синхронному режимі і тому вимагає F = 2/3, але можна покращити граничну вимогу, припускаючи, що інші ланцюжки блоків безпечні і використовуючи їхню безпеку.

## 4.2 Криптографічні попередні відомості

Цифрові підписи. А схема цифрового підпису є набір алгоритмів (Keygen, Підписати, Підтвердити). Кейген виводить кілька ключів (PK, SK). Лише власник SK може підписувати повідомлення, але будь-хто може перевірити підпис за допомогою відкритого ключа PK. Сьогодні більшість блокчейн-систем використовують одну із стандартних схем підпису, таку як ECDSA, Ed25519 або кілька варіантів.2, 3].

Порогові підписи. А схема порогового підпису дає можливість групі н сторони розділити секретний ключ для схеми підпису таким чином, щоб будь-яке підмножина t + 1 або кілька сторін можуть співпрацювати для створення підпису, але не підмножина t або менша кількість сторін може створити підпис або навіть дізнатися який-небудь інформацію про секретний ключ. Підписи, створювані пороговими протоколами для ECDSA та EdDSA, виглядають ідентично підписам, створюваним автономними алгоритмами.

Схема порогового підпису замінює Кейген і підписати алгоритми звичайної схеми підпису з розподіленими н партійними протоколами T.Кейген, T.Sign. Для цих протоколів зазвичай потрібно як загальнодоступний широкомовний канал, і приватні парні канали між сторонами, і вони зазвичай включають кілька раундів зв'язку. Після успішного проходження T.Кейген кожен користувач має частку si секретного ключа SK та відповідного відкритого ключа PK. ToT .Sign протокол дозволяє цим сторонам зробити підпис для даного повідомлення, яке дійсне під відкритим ключем PK. Цей підпис можна перевірити будь-яким користувачем Перевіряючий алгоритм оригінальної схеми підпису.

## 4.3 Властивості порогових сигнатур

Є кілька властивостей порогової схеми, які є особливо бажаними для децентралізованих мереж:

**Захист від нечесної більшості.** Деякі порогові схеми мають обмеження у тому, що вони безпечні.

тільки тоді, коли більшість n партії чесні. Таким чином, пороговий параметр повинен бути меншим, ніж n/2 [15]. Цеобмеження зазвичай супроводжується тим фактом, що 2t + 1 чесні сторони потрібні для підписання, хоча тільки t + 1 пошкоджені сторони можуть вступити в змову відновлення секретного ключа. Схеми, які не страждають від цього обмеження, називаються убезпечити себе від нечесної більшості.

Як обговорюється далі у розділі 5.2, кроссчейн-платформи повинні максимально підвищити безпеку своїх мереж і бути в змозі терпіти якнайбільше корумпованих сторін. Таким чином, необхідні схеми, які можуть зазнавати нечесної більшості.

**Попередній підпис, неінтерактивний онлайн-підпис.** Прагнення зменшити навантаження на спілкування

Коли сторони підписують повідомлення, у кількох останніх протоколах визначено значну частину роботи з підпису, яка може бути виконана «в автономному режимі», до того, як стане відомо повідомлення, яке потрібно підписати.18, 13]. Результат цієї автономної фази називається попередній підпис. Виготовлення попередніх підписів розглядається як окремий протокол T. Presign на відміну від T. Кейген і T. Sign. Виходи протоколу попереднього підпису повинні зберігатися сторонами в секреті, доки вони не використовують їх на етапі підписання. Пізніше, коли повідомлення для підпису стане відомо, залишиться зробити лише невеликий обсяг додаткової «онлайнової» роботи T.Sign для завершення підпису.

онлайнT .Sign Етап не вимагає жодного спілкування між сторонами. Кожна сторона просто виконує локальні обчислення для повідомлення та попереднього підпису, а потім оголошує свою частку підпису. (Після публікації ці підписи s1, . . . , st+1 діляться легко комбінуються будь-ким, щоб розкрити фактичний підпис s) Ця властивість називається неінтерактивний онлайн-підпис.

**Надійність.** Порогові схеми гарантують лише те, що частина зловмисників не може підписувати повідомлення або дізнатися про секретний ключ. Однак ця гарантія не виключає можливості того, що зловмисники можуть заблокувати решту всіх від створення ключів або підписів. У деяких схемах зловмисна поведінка навіть однієї сторони може призвести до T.Кейген або T.Sign перервати без корисного висновку. Єдиний вихід — перезапустити протокол, можливо, з різних боків.

Натомість для децентралізованих мереж ми хочемо T.Кейген та T.Sign досягти успіху, якщо хоча б t+ Одна зі сторін чесна, навіть якщо деякі зловмисники відправляють повідомлення у спотвореному форматі або видаляють повідомлення у протоколах. Ця властивість називається міцністю.

**Атрибуція вини.** Здатність виявляти поганих акторів у T.Кейген або T.Sign називається приписуванням провини.

Без атрибуції провини важко надійно виключити чи покарати недобросовісних учасників, й у разі витрати, накладені на недобросовісних учасників, мають нести все. Ця властивість також важлива для децентралізованих мереж, де зловмисна поведінка має бути ідентифікована та економічно дестимульована за допомогою правил скорочення.

Безпека в паралельних настройках. Схема підпису має бути безпечною в паралельних умовах, де кілька екземплярів алгоритмів генерації ключів та підписи можуть бути задіяні паралельно. (Драйверс

ин [16] наприклад, показав атаку на мультипідписні схеми Schnorr у цих налаштуваннях). Існують версії як схем ECDSA, так і схем Шнорра, які задовольняють цим властивостям13, 22].

ECDSA і EdDSA на сьогоднішній день є найбільш широко використовуваними схемами підпису у просторі блокчейну.

Отже, порогові версії обох схем були у центрі уваги недавнього відродження досліджень, і розробок.

Читачі, які цікавляться сучасними технологіями, можуть звернутися до [22, 13, 18] та недавній оглядовий документ [12]

# 5 Мережа Axelar

## 5.1 Проектування відкритої кроссчейн мережі

Мости, що підтримуються мережею Axelar, підтримуються граничними обліковими записами, так що (майже) всі валідатори повинні колективно авторизувати будь-який міжмережевий запит. Проектування мережі, в якій кожен може брати участь у забезпеченні безпеки цих мостів, вимагає виконання таких технічних вимог:

- *Відкрите членство.* Будь-який користувач повинен мати можливість стати валідатором (за правилами мережі).

- *Відновлення членства.* Коли валідатор чесно залишає систему, його ключ має бути відповідним чином відкликано.

- *Заохочення та слешинг.* Зловмисні валідатори повинні бути ідентифіковані, а їхні дії повинні бути ідентифіковані та розглянуті протоколом.

- *Консенсус.* Порогові схеми власними силами визначаються як автономні протоколи. Для поширення повідомлень між вузлами нам потрібні як широкомовні, і двоточкові приватні канали. Більше того, валідатори повинні узгоджувати останній стан кожного виклику порогових схем, оскільки вони мають кілька раундів взаємодії.

- *Ключовий менеджмент.* Як звичайні валідатори у будь-якій системі PoS повинні ретельно охороняти свої ключі, так і валідатори Axelar повинні охороняти свої межі. Ключі потрібно обертати, розділяти між онлайн та офлайн частинами і т.д.

Axelar починає з моделі Delegated Proof-of-Stake, у якій спільнота вибирає набір валідаторів для досягнення консенсусу. Зверніть увагу, що стандартні порогові схеми відносяться до кожного гравця однаково і не мають поняття «вага» у консенсусі. Відтак мережа повинна адаптувати їх, щоб враховувати вагу валідаторів. Простий підхід полягає у призначенні декількох порогових часток більшим валідаторам. Нижче описано три основні функції, які колективно виконують валідатори.

- *Генерація граничного ключа.* Існуючі алгоритми генерації порогового ключа для стандартних схем підпису блокчейна (ECDSA, Ed25519) є інтерактивними протоколами між декількома учасниками (див. Розділ 4). Спеціальна транзакція в мережі Axelar дає вказівку валідаторам розпочати виконання цього протоколу з відстеження стану. Кожен валідатор запускає процес порогового демона, який відповідає за безпечне збереження секретного стану. Для кожної фази протоколу:

    1. Валідатор зберігає стан протоколу у своїй локальній пам'яті.

    2. Він викликає секретний демон для генерації повідомлень відповідно до опису протоколу для інших валідаторів.

    3. Він поширює повідомлення або через широкомовне розсилання, або через приватні канали іншим валідаторам.

    4. Кожен валідатор виконує функції переходу стану, щоб оновити свій стан, перейти до

наступного етапу протоколу та повторити описані вище кроки.

В Наприкінці протоколу в ланцюжку Axelar генерується пороговий відкритий ключ, і його можна відобразити назад користувачеві (наприклад, для депозитів) або додатку, що згенерував початковий запит.

- *Підписання порога.* Запити на підпис у мережі Axelar обробляються аналогічно до запитів на генерацію ключів. Вони викликаються, наприклад, коли користувач хоче вивести актив із одного з протоколів. Це ланцюги. інтерактивні протоколи, і перехід стану між раундами запускається як функція повідомлень, що розповсюджуються через подання блокчейна Axelar та локальну пам'ять кожного валідатора.

- *Обробка змін членства у Валідаторі.* Набір валідаторів необхідно періодично змінювати, щоб нові зацікавлені сторони могли приєднатися до набору. Після оновлення набору валідаторів нам потрібно оновити пороговий ключ, який буде використовуватись у новому наборі. Таким чином, якби ми дозволили будь-кому приєднуватися будь-коли, нам довелося б дуже часто оновлювати пороговий ключ. Щоб запобігти цьому, ми змінюємо валідаторів кожні блоки. В інтервалах патрони, набір B p і пороговий ключ фіксовані. У кожному раунді, який є цілим кратним параметром, ми оновлюємо набір валідаторів наступним чином:

  1. У будь-якому раунді R стан Axelar відстежує поточний набір валідаторів VR. VR+1 = VR поки що не R + 1 кратно T .

  2. Під час раундів ((i − 1)T, iT) користувачі публікують зв'язуючі/роз'єднуючі повідомлення.

  3. Наприкінці раунду iT ці повідомлення застосовуються до ViT-1, щоб отримати ViT .

- *Генерація порогового ключа та підпис за наявності мінливих валідаторів.* Блокчейн Axelar може видати запит на новий ключ або граничний підпис у раунді p. Процес підписання займає більше одного раунду, і ми не хочемо уповільнювати досягнення консенсусу, тому просимо підписати до раунду R + 10 пусків. Зокрема, валідатори починають раунд R + 10 лише після перегляду сертифіката на раунд R + 9 та підпис для кожного запиту кейгена/підпису, виданого на раунді R. Підсумки всіх раундів R запити мають бути включені до блоку R + 11. Іншими словами, раунд R блокує пропозицію, яка не містить результатів раунду R - 11 вважається недійсною, і валідатори за нею не голосують. Щоб переконатися, що всі порогові повідомлення підписані до оновлення набору валідаторів, Axelar не видає жодних порогових запитів протягом раунду, що дорівнює -1,-2, . . . ,-9 За модулем T .

## 5.2 Безпека мережі

Безпека систем блокчейна залежить від різних криптографічних та ігрових протоколів, а також від децентралізації мережі. Наприклад, у блокчейнах з доказом частки без належних стимулів валідатори можуть вступити в змову та переписати історію, викрадаючи у процесі засоби інших користувачів. У мережах з доказом роботи без достатньої децентралізації досить легко створювати довгі форки та подвійні витрати, як довели численні атаки на Bitcoin Gold та Ethereum Classic.

Більшість досліджень безпеки блокчейна була зосереджена на суверенних ланцюгах. Але як тільки ланцюжки взаємодіють, необхідно враховувати нові вектори атак. Наприклад, припустимо, що Ethereum взаємодіє з невеликим ланцюжком блоків X через прямий міст, контрольований двома смарт-контрактами, одним на Ethereum і одним на X. Крім інженерних проблем, які ми підсумовували у розділі1.1, Необхідно вирішити, що відбувається, коли припущення про довіру X порушуються. У цьому випадку, якщо ETH перемістився в X, валідатори X можуть вступити в змову, щоб підробити історію X, в якій вони тримають усі ETH, опублікувати підроблені докази консенсусу Ethereum і вкрасти ETH. Ситуація ще гірша, коли X з'єднаний з декількома іншими ланцюгами через прямі мости, де, якщо X розгалужується, ефекти поширюються через мости. Налаштування керівних принципів управління відновленням для кожного парного мосту є непосильним завданням для будь-якого окремого проекту.

Мережа Axelar вирішує проблеми безпеки, використовуючи такі механізми:

- *Максимальна безпека.* Axelar встановлює поріг безпеки на рівні 90%, а це означає, що майже всі валідатори повинні будуть змовитися, щоб зняти будь-які засоби, які заблоковані його мережею., або підробити докази стану.[1]. На практиці було відмічено, що валідатори PoS мають дуже великий час безвідмовної роботи (близько 100%), якщо вони належним чином мотивовані. Отже, мережа Axelar вироблятиме блоки навіть незважаючи на цей високий поріг. Проте в окремих випадках, коли щось піде не так і мережа зупиниться, їй потрібні надійні резервні механізми для перезавантаження системи, описаної нижче.

- *Максимальна децентралізація.* Оскільки в мережі використовуються порогові схеми підпису, кількість валідаторів може бути максимально великою. Мережа не обмежена кількістю валідаторів, які ми можемо підтримувати, лімітами транзакцій або зборами, які можуть виникнути, наприклад, при використанні мультипідписів у різних ланцюжках, де складність (і збори) збільшуються лінійно з кількістю валідаторів.[2]

- *Надійні резервні механізми* Перше питання, яке необхідно вирішити в мережі з високими порогами безпеки, як зазначено вище, це те, що відбувається, коли сама мережа зупиняється. Допустимо, сама мережа Axelar зависає. Чи можемо ми мати запасний механізм, який дозволив би користувачам відновити свої кошти? Щоб усунути будь-яку потенційну зупинку мережі Axelar, кожен обліковий запис порогового мосту в блокчейні X, яку колективно контролюють валідатори Axelar, має «ключ аварійного розблокування». Цим ключем можна поділитись

*1 Остаточний параметр, який буде вибрано для розгортання мережі, може бути скориговано.

*2 Для деяких блокчейнів мультипідписи пропонують розумну альтернативу, коли кількість газу невелика, а формати повідомлень, що підтримуються, підходять. Але вони не підходять для двох найбільших платформ, таких як Біткойн та Ефіріум.

між тисячами сторін і може навіть бути налаштованим ключем для блокчейна X, який використовується в спільноті цього ланцюжка. Отже, якщо мережа Axelar зупиниться, цей ключ діятиме як запасний варіант і дозволить відновити активи (докладніше див. нижче).

- *Максимальна децентралізація резервних механізмів.* Цей резервний механізм включає вторинний набір відновлення користувачів, у якому кожен може брати участь без будь-яких витрат. Цим користувачам не потрібно бути в мережі, запускати вузли або координувати свої дії. Їх «дежурять» тільки в тому випадку, якщо мережа Axelar зависає і не може відновитись. Безпека мережі підвищується за рахунок дуже високого порога для основного набору валідаторів та максимально децентралізованого вторинного набору для відновлення.

- *Спільне керування.* Загальний протокол керує мережею Axelar. У сукупності користувачі можуть голосувати за те, який ланцюжок повинен підтримуватися через мережу. Мережа також виділятиме пул коштів, які можна використовувати для відшкодування користувачам витрат у разі непередбачених надзвичайних ситуацій, які також контролюються за допомогою протоколів управління.

Нижче обговорюються різні механізми безпеки.

**Механізми повернення.**Коли Axelar зупиняється через високий поріг, «ключ аварійного розблокування» перебирає управління мережею. Існує кілька способів створення екземпляра цього ключа розблокування, і деякі ланцюжки/програми можуть використовувати інший варіант для «набору відновлення» або повністю відмовитися:[3]

- *Варіант а.* Поділіться ключем між фондами блокчейн-проектів та шановними людьми у спільноті.

- *Варіант б.* Розділіть вибір між сторонами, які вибрали через делегований механізм PoS.

- *Варіант ст.*Для облікових записів, що управляють активами та інформацією для ланцюжка/додатка X, надайте загальний

  ключ зацікавленим сторонам/валідаторам X. Припускаючи, що X має механізми управління, ті ж механізми управління можуть застосовуватися для визначення курсу дій, якщо Axelar зупиниться.

Тепер, зважаючи на ідентифікатори користувачів для відновлення та їх відкриті ключі, простий протокол створює спільний доступ до ключа відновлення, про який ніхто не знає. Крім того, користувачам набору для відновлення не потрібно перебувати в мережі, доки не будуть викликані для відновлення через механізми керування. Наслідуючи стандартні розподілені протоколи генерації ключів, кожен валідатор Axelar використовує випадкове значення. Секретний ключ відновлення генерується шляхом підсумовування цих значень. Замість того, щоб виконувати підсумовування у відкритому вигляді, всі загальні ресурси шифруються відкритими ключами користувачів відновлення, а потім складаються гомоморфно (це передбачає адитивне гомоморфне шифрування і додатковий рівень нульового розголошення, обидва з яких легко доступні). Результатом цього протоколу є відкритий ключ відновлення. RPK і потенційно тисячі шифрів (під відкритими ключами користувачів відновлення) часток відповідного секретного ключа Enci(si), які розповсюджуються серед їхніх власників (наприклад, розміщуються у ланцюжку). Бридж-контракти Axelar включають можливість відшкодування коштів за допомогою RPK за певних умов. Нарешті, також можна оновити цей ключ відновлення і навіть змінити набір користувачів, які володіють акціями, без будь-яких дій з боку акціонерів. Бридж-контракти Axelar включають можливість відшкодування коштів за допомогою RPK за певних умов. Нарешті, також можна оновити цей ключ відновлення і навіть змінити набір користувачів, які володіють акціями, без будь-яких дій з боку акціонерів. Бридж-контракти Axelar включають можливість відшкодування коштів за допомогою RPK за певних умов. Нарешті, також можна оновити цей ключ відновлення і навіть змінити набір користувачів, які володіють акціями, без будь-яких дій з боку акціонерів.

Якщо ланцюжок X, пов'язаний з Axelar, розірветься, є кілька варіантів:

- Встановіть обмеження на вартість активів у доларах США, які можуть бути переміщені у/з X у будь-який день. Таким чином, шкідливий ланцюг X може вкрасти лише невелику частину всіх активів, які пов'язані з ним, перш ніж валідатори Axelar виявлять це і спрацюють механізми управління з наступних куль.

- Модуль керування Axelar можна використовувати для голосування з приводу того, що відбувається у таких ситуаціях. Наприклад, якщо виникає несерйозна помилка і спільнота перезапускає X, керування Axelar може прийняти рішення про перезапуск з'єднання з місця, де воно було перервано.

- Якщо ETH перемістився в X, власний ключ відновлення Ethereum може визначити, що станеться з активами

  ETH.

---

*3 Остаточне розгортання мережі Axelar буде завершено ближче до запуску мережі.

# 6 Протокол міжмережевого шлюзу (CGP)

В У цьому розділі ми пояснюємо протокол міжмережевого шлюзу та механізми маршрутизації на двох основних прикладах, загальних для багатьох додатків:

Синхронізація станів (розділ 6.2). Розміщувати інформацію про стан вихідного блокчейну S в стан цільового блокчейну D.
*(Наприклад, опублікуйте заголовок блоку біткойнів у блокчейні Ethereum.)*

Передача активів (Розділ 6.3). Перенести цифровий актив із S до D і назад.
*(Наприклад, перенесіть біткойни з блокчейна Біткойн в блокчейн Ефіріуму, а потім назад в блокчейн Біткойн.)*

Для простоти вважатимемо, що ланцюжок D має хоча б мінімальну підтримку смарт-контрактів, але S може бути будь-яким блокчейном.

## 6.1 Аккаунти в інших мережах

Щоб з'єднати різні ланцюжки, у кожному ланцюжку створюються граничні облікові записи, які контролюють потік цінності та інформації через них. Для ланцюга Chain, позначимо рахунок через $Chain_{Axelar}$.

Біткойн-рахунок. Для біткойнів та інших ланцюжків смарт-контрактів валідатори Axelar створюють пороговий ключ ECDSA відповідно до розділу 5.1. Цей ключ управляє обліковим записом ECDSA в біткойнах і є адресою призначення, куди користувачі надсилають депозити. Персоналізовані порогові ключі можуть бути створені на запит користувача. Ключ може періодично оновлюватися, а останній ключ та персоналізовані ключі можна знайти, запросивши вузол Axelar.

Обліковий запис порогового мосту в ланцюжках зі смарт-контрактами. Позначимо ланцюг через SC. валідатори створюють пороговий ключ ECDSA або ED25519 згідно з розділом5.1, залежно від, який тип ключа підтримує ланцюжок. Позначимо цей ключ через PKAxelar, коли немає жодної двозначності щодо того, про який ланцюжок йдеться. Цей ключ керує обліковим записом смарт-контракту на SC, що позначається $SC_{Axelar}$, і будь-яка програма на SC може запитувати $SK_{Axelar}$ щоб дізнатися PK-адресу цього ключа. Таким чином, будь-яка програма SC може розпізнавати повідомлення, підписані $SC_{Axelar}$. Протокол також має враховувати ротацію значень $PK_{Axelar}$. Це відбувається так:

1. Ініціалізувати $SC_{Axelar}$ на SK. Він зберігає $PK_{Axelar}$ як частину свого стану, який ініціалізується як значення генезису на Axelar. $SC_{Axelar}$ також включає правила оновлення PK.

2. Оновити $PK_{Axelar}$, транзакція формату (оновлення, $PK_{new}$) має бути представлена з підписом чинного $SC_{Axelar}$. Потім встановлюється договір $SC_{Axelar} = PK_{new}$.

3. Щоразу, коли валідатори оновлюють пороговий ключ для SC з $PK^i$ до $PK^{i+1}$, Axelar вимагає, щоб валідатори використовували $SK^i$ підписи (оновлення, $PK^{i+1}$). Згодом цей підпис розміщується на $SC_{Axelar}$. Які оновлюють $PK_{Axelar}$.

## 6.2 Синхронізація станів

Давайте qs позначимо як довільне питання про стан ланцюжка S. Приклади таких питань включають:

- *"У якому раунді блоку, якщо будь-який, з'явилася транзакція tx?"*
- *"Яке значення певного поля даних?"*
- *Що таке кореневий хеш Меркла входу стану S в блок-раунді 314159?*

Давайте позначимо as як правильну відповідь на qs і припустимо, що кінцевий користувач або програму вимагають, щоб as бути розміщеним у ланцюжку D. Мережа Axelar відповідає цій вимогі таким чином:

1. Користувач надсилає запит qs на одному з бридж-акаунтів (які згодом підхоплюються валідаторами) або безпосередньо в блокчейн Axelar.

2. В рамках консенсусу Axelar кожен валідатор має запускати програмне забезпечення вузла для ланцюжків. S, D. Валідатори Axelar запитують API свого ланцюжка S програмне забезпечення вузла для відповіді as та повідомити відповідь у ланцюжок Axelar.

3. Один раз > F зважені валідатори повідомляють про одну й ту саму відповідь у раунді R, Axelar просить валідаторів підписати as.

4. Використовуючи граничну криптографію, валідатори підписують as. Підпис включений у блок R + 11.

5. Будь-який може взяти значення зі знаком as з блоку R + 11 і опублікувати D.

6. Заявка обслуговується. Будь-яка програма на D тепер може приймати значення зі знаком as, запит $D_{Axelar}$ для останніх $PK_{Axelar}$, і переконайтеся, що підпис as відповідає $PK_{Axelar}$. Валідатори також публікують as на бридж аккаунт у ланцюжку D, які програми можуть отримати.

## 6.3 Міжмережева передача активів

Мережа дозволяє передавати цифрові активи між ланцюжками, розширюючи робочий процес синхронізації станів розділу. 6.2.

Достатній запас pegged-S токенів друкуються та контролюються $D_{Axelar}$ у його ініціалізації. Припустимо, користувач вимагає обміну x кількість токенів у вихідному ланцюжку S за x кількість pegged-S токенів у ланцюжку призначення D, що підлягає депонування в D-адресу wd на вибір користувача. Ми представляємо повністю загальний робочий процес, який підтримує довільні вихідні ланцюжки S - навіть такі мережі, як Біткойн, які не підтримують смарт-контракти:

1. Користувач (або програма, що діє від імені користувача) надсилає запит на перенесення (x, wd) на обліковий запис порогового мосту, який згодом надсилається до мережі Axelar.

2. Валідатори Axelar використовують граничну криптографію для колективного створення нової адреси депозиту. ds за S. Вони публікують ds до блокчейну Axelar.

3. Користувач (або програма, що діє від імені користувача) дізнається ds шляхом моніторингу блокчейна Axelar. Користувач відправляє x кількість S-токенів на адресу ds через звичайну S-транзакцію txs використовуючи її улюблене програмне забезпечення для ланцюжка S. *(Через порогову властивість* ds, токени не можна витратити з ds якщо гранична кількість

валідаторів не погоджено це зробити.)

4. $tx_s$ розміщено на Axelar. Валідатори запитують API свого ланцюжка S програмне забезпечення вузла для існування $tx_s$ і, якщо відповідь «вірна», повідомляє відповідь у ланцюжок Axelar.

5. Як тільки > F виважені валідатори повідомляють «true» для $tx_s$ в раунді R, Axelar просить валідаторів підписати транзакцію ad який посилає x кількість pegged-S токенів від $D_{Axelar}$ до $w_d$
6. Використовуючи граничну криптографію, валідатори підписують ad. Підпис включений у блок R + 11.

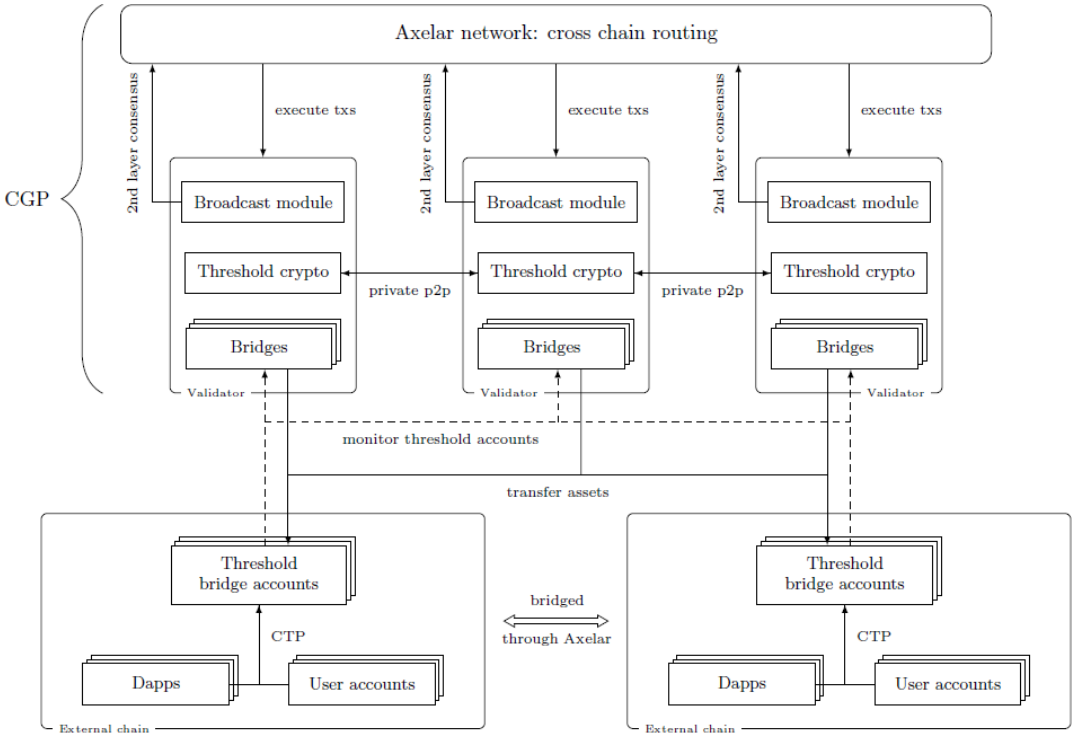
7. Будь-який може взяти значення зі знаком ad з блоку R+11 і опублікувати D.

8. Запит був обслужений, як тільки ad розміщений на D передача обробляється.

Тепер припустимо, що користувач вимагає викупити x' кількість обгорнутих-S токенів з ланцюжка D повернутися до ланцюжка S, що підлягає депонування в S-адресу ws на вибір користувача. Робочий процес виглядає так:

1. Користувач ініціює запит на передачу ($x'$, $w_s$) шляхом внесення x' кількість wrapped-S токенів у Cd через звичайну D-транзакцію з використанням її улюбленого програмного забезпечення для мережі D

2. ($x'$, $w_s$) опубліковано на Axelar. Валідатори запитують API свого вузла ноди ланцюга D про існування ($x'$, $w_s$) і, якщо відповідь «вірна», повідомляється відповідь у ланцюжок Axelar




Малюнок 1: Схема компонентів

3. Як тільки > F виважені валідатори повідомляють «true» для ($x'$, $w_s$) у раунді R, Axelar просить валідаторів підписати транзакцію as який посилає x' кількість S токенів від $S_{Axelar}$ до $w_s$.

4. Використовуючи граничну криптографію, валідатори підписують as. Підпис включений у блок R + 11.

5. Будь-який може взяти значення зі знаком as з блоку R+11 і опублікувати S.

6. Запит був обслужений, як as розміщений на S передача обробляється.

Додаткові запити, що підтримуються рівнем маршрутизації CGP, включають блокування, розблокування або передачу активів ланцюжком.

Досягнення атомарного потоку транзакцій між ланцюжками. Залежно від типу міжмережевого запиту Axelar намагається забезпечити виконання відповідних транзакцій у кількох ланцюжках чи жодній. Для цього кожен запит може знаходитися в одному з наступних станів у блокчейні Axelar: (ініціалізовано, очікується, завершено, минув час очікування). Якщо тайм-аут у стадії pending спрацьовує, запит повертає код помилки. Деякі події тайм-ауту також починають повертати гроші подію: наприклад, якщо актив з одного ланцюжка необхідно перевести в актив в іншому ланцюжку, якщо ланцюжок, що приймає, не обробив транзакцію, актив повертається вихідному користувачеві.

## 7 Протокол міжмережевої передачі (CTP)

CTP – це протокол рівня програми, який дозволяє програмам легко використовувати кроссчейн-функції. Ми пояснюємо інтеграцію, зосередивши увагу на функціях передачі активів (наприклад, що використовуються у DeFi). Ці програми зазвичай складаються з трьох основних компонентів: графічного інтерфейсу користувача, смарт-контрактів в одному ланцюжку та проміжного вузла, який публікує транзакції між інтерфейсом та смарт-контрактами. Зовнішні інтерфейси взаємодіють із гаманцями користувачів, щоб приймати депозити, обробляти зняття коштів тощо. Програми можуть використовувати кроссчейн-функції.
шляхом виклику CTP-запитів, аналогічних методам HTTP/HTTPS GET/POST. Ці запити згодом приймаються рівнем CGP для виконання, а результати повертаються користувачам.

- *CTP-запити*. Розробники програм можуть розміщувати свої програми у будь-якому ланцюжку та інтегрувати свої смарт-контракти з обліковими записами порогового мосту для виконання запитів CTP.

- *Порогові проміжні рахунки*. Припустимо, розробник програми будує свої контракти в ланцюжку A. Потім він посилатиметься на порогові мостові контракти, щоб отримати підтримку між ланцюжками. Цей контракт дозволяє додаткам:

  - Зареєструйте блокчейн, з яким він хотів би спілкуватися.

  – Зареєструйте активи у цьому ланцюжку блоків, які він хотів би використати.

  – виконувати операції над активами, такі як прийом депозитів, обробка зняття коштів та інші функції (аналогічно, скажімо, викликам контрактів ERC-20).

Припустимо, відомий додаток DeFi, MapleSwap, який спочатку знаходиться в ланцюжку A, реєструється з граничним обліковим записом мосту. Валідатори Axelar колективно управляють самим контрактом у відповідному ланцюжку. Припустимо, користувач хоче внести депозит у пару між активами X і Y, які знаходяться у двох ланцюжках відповідно. Потім, коли користувач надсилає такий запит, він прямує через обліковий запис порогового моста в мережу Axelar для обробки. Форма там виконуються наступні кроки:

1. Мережа Axelar розуміє, що ця програма зареєстрована для міжмережної підтримки активів. Він генерує ключ депозиту, використовуючи граничну криптографію та консенсус для користувача у відповідних ланцюжках A та B.

2. Пов'язані відкриті ключі повертаються в програму та відображаються для користувача, який може використовувати свої улюблені гаманці для надсилання депозитів. Відповідний секретний ключ є спільним для всіх валідаторів Axelar.

3. Коли депозити підтверджені, Axelar оновлює свій міжмережевий каталог, щоб записати, що користувач у відповідних ланцюжках депонував ці активи.

4. Валідатори Axelar виконують багатосторонні протоколи для створення порогового підпису, який дозволяє оновлювати обліковий запис порогового мосту в ланцюжку A, де знаходиться програма.

5. Потім запит CTP повертається до смарт-контрактів програми DeFi, які можуть оновлювати свій стан, оновлювати формули доходності, обмінні курси або виконувати інші умови, пов'язані зі станом програми.

Протягом усього цього процесу мережа Axelar на високому рівні діє як децентралізований оракул для читання/запису між ланцюжками, CGP – це рівень маршрутизації між ланцюжками, а CTP – протокол програми.

Додаткові запити між мережами. CTP підтримує більш загальний крос-ланцюжок між додатками в блокчейнах, наприклад:

- Виконує служби імен відкритих ключів (PKNS). Це універсальний каталог для порівняння відкритих ключів із телефонними номерами/дескрипторами Twitter (кілька проектів, таких як Celo, надають ці функції на своїх платформах).

- Тригери кроссчейн-додатків. Додаток у ланцюжку A може оновити свій стан, якщо якийсь інший додаток у ланцюжку B відповідає критеріям пошуку (процентна ставка< Ікс).

- Компонування смарт-контрактів. Смарт-контракт у ланцюжку A може оновлювати свій стан залежно від стану контрактів у ланцюжку B або ініціювати дію для оновлення смарт-контракту у ланцюжку B.

На високому рівні ці запити можуть бути оброблені, оскільки в сукупності протоколи CTP, CGP і мережа Axelar можуть передавати і записувати довільну інформацію про стан через блокчейни.

# 8 Заключення

ВПротягом наступних років важливі програми та активи будуть створені на основі кількох блокчейн-екосистем. Мережа Axelar можна використовувати для включення цих блокчейнів до єдиного рівня міжмережевого зв'язку. Цей рівень забезпечують протоколи маршрутизації та рівня програм, що відповідають вимогам як розробників платформ, так і розробників додатків. Розробники додатків можуть використовувати найкращі платформи для своїх потреб та використовувати простий протокол та API для доступу до глобальної крос-чейн ліквідності, користувачів та зв'язку з іншими ланцюжками.

# використана література

[1] Алте Пеггі. https://github.com/cosmos/nerri. [Процитованона сторінці 2.]
[2] Детерміноване використання алгоритму цифрового підпису (dsa) та алгоритму цифрового підпису на основі еліптичнихкривих (ECDS). https://tools.ietf.org/html/rfc6979. [Процитовано на сторінці 5.]
[3] Алгоритм цифрового підпису на основі кривої Едвардса (Eddsa). https://tools.ietf.org/html/rfc8032. [Цитується

сторінка 5.]
[4] Технічний документ Eos.io v2. https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md. [Цитується на стор. 1.]

[5] Ethereum: безпечна децентралізована узагальнена книга

транзакцій.https:// ethereum.github.io/yellowpaper/paper.pdf. [Процитовано на сторінці 1.]
[6] Найближчий офіційний документ. https://near.org/papers/the-official-near-white-paper/. [Процитовано насторінці1.]

[7] Райдужний міст. https://github.com/near/райдужний міст. [Процитованона сторінці 2.]

[8] Рен: віртуальна машина, що зберігає конфіденційність, на якій працюють фінансові програми з нульовим розголошенням
https:
інформації. //whitepaper.io/document/419/ren-litepaper. [Процитовано насторінці 3.]
[9] tbtc: децентралізований токен erc-20, що погашається, підтримуваний btc. https://docs.keep.network/tbtc/index.
пдф. [Процитовано на сторінці 2.]

[10] Thorchain: децентралізована мережа ліквідності.https://thorchain.org/. [Процитованона сторінці 3.]

[11] Курт М. Алонсо. Нуль до монеро.https://www.getmonero.org/library/Zero-to-Monero-1-0-0.
пдф. [Процитовано на сторінці 1.]

[12] Жан-Філіп Омассон, Адріан Амелінк та Омер Шломовіц. Огляд рогового підпису ecdsa. Cryptology ePrint Archive, звіт 2020/1390, 2020.https://eprint.iacr.org/2020/1390. [Цитується сторінка 6.]

[13] Ран Канетті, Ніколаос Макріяніс і Уді Пелед. Uc неінтерактивний, проактивний, пороговий Cryptology ePrint Archive, звіт 2020/492, 2020 ecdsa. https://eprint.iacr.org/2020/492. [Цитується сторінка6.]
[14] Технічні документи cLabs. https://celo.org/papers. [Процитованона сторінці 1.]

[15] Іван Дамгард, Томас Пелле Якобсен, Йеспер Буус Нільсен, Якоб Іллеборг Пагтер та Міхаель Бексванг Остергорд. Швидкий поріг ECDSA із чесною більшістю. BSCN, том 12238 Конспект лекцій з інформатики, сторінки 382-400. Спрингер, 2020.[Процитовано на сторінці6.]
[16] Ману Дрейверс, Касра Едалатнеджад, Браян Форд, Ейке Кільц, Джуліан Лосс, Грегорі Невен та Ігор Степанов. Про безпеку двораундового мультипідпису. ВСімпозиум IEEE з безпеки та конфіденційності, сторінки
1084-1101. IIEP, 2019.[Процитовано на сторінці6.]

[17] Синтія Дворк, Ненсі Лінч та Ларрі Стокмайєр. Консенсус за наявності часткової синхронності.https://groups.csail.mit.edu/tds/papers/Lynch/jacm88.pdf. [Процитовано на сторінці 5.]

[18] Росаріо Дженнаро та Стівен Голдфедер. Один раунд порогового ecdsa з ідентифікованим перериванням. Cryptology ePrint Archive, звіт 2020/540, 2020р.https://eprint.iacr.org/2020/540. [Процитованона сторінці 6.]

[19] Йоссі Гілад, Ротем Хемо, Сільвіо Мікалі, Георгіос Влахос та Микола Зельдович. Algorand: Масштабування візантійських угод для криптовалюту. Матеріали 26 симпозіуму за принципами операційних систем, 2017 р.https://dl.acm.org/doi/pdf/10.1145/3132747.3132757. [Процитовано насторінці 1.]

[20] Еван Керейакс, До Квон, Марко Ді Маджіо та Ніколас Платіас. Гроші Terra: стабільність та прийняття.
https://terra.money/Terra_White_paper.pdf. [Процитовано на сторінці 1.]

[21] Ангелос Кіайяс, Олександр Рассел, Бернардо Давид та Роман Олійников. Ouroboros: безпечний блокчейн-протокол з доказом частки.https://eprint.iacr.org/2016/889.pdf. [Процитованона сторінці 1.]

[22] Челсі Комло та Ян Голдберг. Frost: гнучкі сигнатури порога Шнорра оптимізовані для раундів. Cryptology ePrint Archive, звіт 2020/852, 2020https://eprint.iacr.org/2020/852. [Процитованона сторінці6.]

[23] Дже Квон та Ітан Бухман. Космос: мережа розподілених реєстрів.https://cosmos.network/resources/whitepaper. [Цитується на сторінках 1 і 2.]

[24] Лавинна команда. Лавинна платформа.https://www.avalabs.org/whitepapers. [Цитується на сторінках 1 і 2.]

[25] Гевін Вуд. Polkadot: Бачення гетерогенної багатоланцюгової структури.https://polkadot.network/PolkaDotPaper.pdf. [Цитується на сторінках 1 і 2.]