Лабораторна робота №6: Корпоративні блокчейни та перспективи технології

Мета роботи: Ознайомитися з архітектурою корпоративних блокчейн-систем (на прикладі Hyperledger Fabric), вивчити механізми масштабування Ethereum (L2) та освоїти концепції приватності за допомогою Zero-Knowledge Proofs (ZKP).

1. Теоретична частина

1.1. Enterprise-рішення: Публічні vs Приватні мережі

Публічні блокчейни (Ethereum, Bitcoin) працюють за принципом Permissionless — будь-хто може стати вузлом або відправляти транзакції. Проте для бізнесу це створює ряд проблем:

1. Відсутність приватності: Всі дані транзакцій доступні всьому світу.
2. Продуктивність: Потреба в глобальному консенсусі обмежує кількість транзакцій на секунду (TPS).
3. Комплаєнс: Бізнесу потрібно знати, з ким він взаємодіє (KYC/AML), що неможливо в анонімних мережах.

Permissioned Blockchain (приватна мережа) дозволяє обмежити коло учасників. Доступ надається лише ідентифікованим контрагентам, що робить систему контрольованою та швидкою.

1.2. Архітектура Hyperledger Fabric

Hyperledger Fabric — це модульна платформа для корпоративних рішень.

- Канали (Channels): Механізм ізоляції. Учасники можуть створювати окремі канали для конфіденційних угод. Дані в каналі абсолютно невидимі для учасників каналу .
- MSP (Membership Service Provider): Компонент, що керує ідентифікаторами (сертифікати X.509). Він визначає, хто має право підключатися до мережі та які ролі виконувати.
- Ordering Service (Вузли-ордератори): На відміну від Ethereum, де майнери збирають блоки, у Fabric є спеціальні вузли, які лише впорядковують транзакції у блоки та розсилають їх іншим вузлам. Це забезпечує Deterministic Finality (транзакція вважається остаточною одразу після включення в блок).

- Консенсус: Fabric не використовує енергозатратний PoW. Замість цього застосовуються алгоритми типу Raft або PBFT, що базуються на довірі між ідентифікованими вузлами.

1.3. Рішення другого рівня (L2) та Rollups

Проблема масштабованості (Scalability Trilemma) змушує виносити обчислення за межі основної мережі (Mainnet).

- Optimistic Rollups (напр. Arbitrum, Optimism): Припускають, що всі транзакції валідні ("оптимістично"). Якщо хтось помітить помилку, він може подати "доказ шахрайства" (Fraud Proof) протягом певного часу.
- ZK-Rollups (напр. zkSync, Starknet): Використовують математичні докази для підтвердження валідності кожної пачки транзакцій. Вони швидші у фіналізації, оскільки не потребують періоду очікування на оскарження.

1.4. Zero-Knowledge Proofs (ZKP)

Доказ із нульовим розголошенням — це криптографічний метод, який дозволяє одній стороні (Доводжувачу) переконати іншу сторону (Верифікатора) у тому, що вона знає певне значення , не розкриваючи при цьому саме значення .

Простий приклад: Ви можете довести, що знаєте пароль від сейфа, відкривши його перед свідком, але не показуючи самому свідку комбінацію цифр. У Web3 це дозволяє створювати Private Transactions та системи ідентифікації, де ви підтверджуєте право на дію, не розкриваючи персональних даних.

1.5. Концепція Web 3.0

Web 3.0 — це ідея "Інтернету цінності".

- Web 1.0: Read (Читання).
- Web 2.0: Read-Write (Платформи збирають ваші дані).
- Web 3.0: Read-Write-Own (Ви володієте своїми даними через цифрові активи та децентралізовані ID).

2. Практична частина

Завдання 1: Архітектурне моделювання

Спроектуйте логічну схему блокчейн-мережі для ланцюга поставок кави. Учасники: Фермер, Логістична компанія, Митниця, Обсмажувальник, Ритейлер.

Вимоги до схеми:

1. Створіть Канал 1 (Фермер - Логіст - Обсмажувальник) для відстеження походження.

2. Створіть Канал 2 (Обсмажувальник - Ритейлер) для фінансових розрахунків.
3. Поясніть, чому Митниця повинна мати доступ лише до певних даних (використання Private Data Collections).

Завдання 2: Знайомство з ZKP (ZoKrates)

Використовуючи мову програмування для ZKP — ZoKrates, ми створимо доказ того, що користувачу більше 18 років, не розкриваючи його точний вік.

Скрипт (мова ZoKrates):

```
def main(private field age, field threshold) {
    assert(age >= threshold);
    return;
}
```

Інструкція: 1. Перейдіть у ZoKrates Online IDE. 2. Вставте код. 3. Виконайте Compile, потім Setup. 4. Введіть age = 20, threshold = 18 у вкладці Compute. 5. Згенеруйте Proof. Тепер цей файл "proof.json" можна відправити будь-кому, і вони математично переконаються, що вам >18, не знаючи, чи вам 20, чи 100.

Завдання 3: Аналіз L2 мереж

Відкрийте два експлорери:

1. Etherscan (Mainnet).
2. Arbiscan (Arbitrum L2).

Порівняйте:

- Середню вартість транзакції (Gas Price).
- Час створення блоку.
- Візуально порівняйте кількість транзакцій у блоці.

3. Контрольні запитання

1. У чому головна архітектурна відмінність Hyperledger Fabric (Execute-Order-Validate) від Ethereum (Order-Execute)?
2. Як канали у Fabric допомагають бізнесу дотримуватися банківської таємниці?
3. Чому ZK-Rollups вважаються більш технологічно досконалими за Optimistic Rollups у довгостроковій перспективі?

4. Наведіть приклад використання ZKP у повсякденному житті (крім блокчейну).
5. Що таке "Self-Sovereign Identity" (SSI) у контексті Web 3.0?