



Вступ до Blockchain та Algorand

ЛЕКЦІЯ 1

План

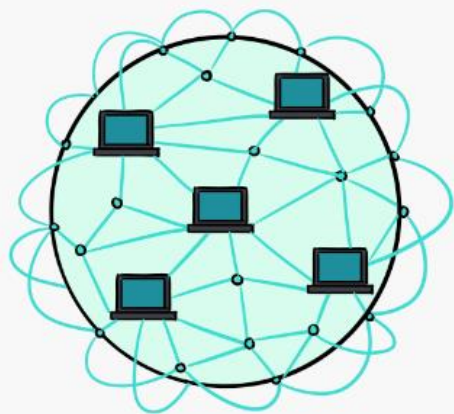
- Що таке блокчейн. Види блокчейнів (private/public).
- Різниця між блокчейном та БД.
- Використання блокчейну в різних галузях.
- Блокчейн Algorand. Pure Proof of Stake (PPoS).
- AVM.
- Що таке Dapps та смарт-контракти. Роль смарт-контрактів в екосистемах блокчейну.



Блокчейн — це загальний незмінний реєстр, що полегшує процес запису транзакцій та відстеження активів у бізнес-мережі. Актив може бути матеріальним (будинок, машина, гроші, земля) чи нематеріальним (інтелектуальна власність, патенти, авторські права, брендинг). Практично все, що має цінність, можна відстежувати та продавати в мережі блокчейн, що знижує ризики та витрати для всіх учасників.

- IBM Блокчейн

•



Blockchain Ledger



Traditional Ledger

Блокчейн - це публічний реєстр (або файл) транзакційних даних, розподілений по безлічі комп'ютерів (вузлів) в мережі. Всі ці вузли працюють разом, використовуючи один і той же набір програмного забезпечення та правил для перевірки транзакцій для додавання до остаточного реєстру.

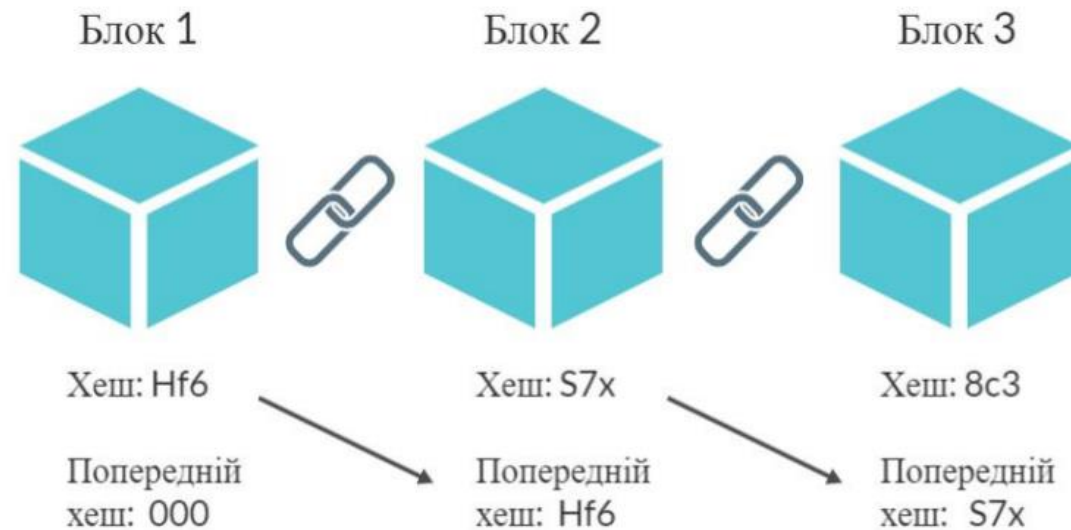
- *Algorand blockchain*

Коротка історія блокчейну



Блокчейн отримав назву як ланцюг блоків.

Що робить його особливим, так це те, що кожен блок зберігає знання про попередній. Кожен блок у ланцюжку має хеш, який подібний до унікального цифрового відбитка пальця, що представляє певну частину інформації, яка пов'язує його з попереднім блоком, створюючи ланцюжок блоків, практично захищений від несанкціонованого доступу.





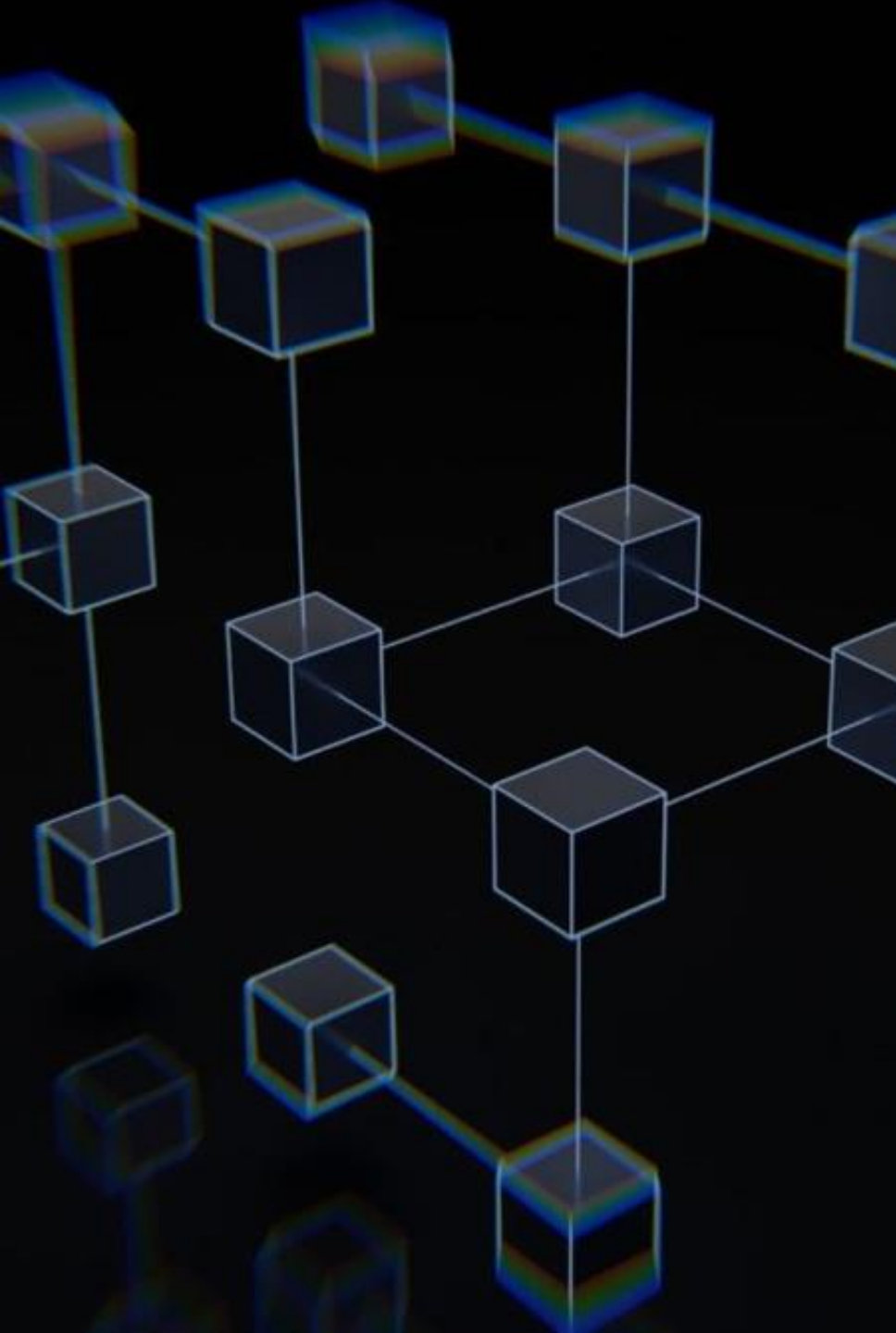
Види блокчейнів.

Public vs Private

- Публічний блокчейн - це тип блокчейну, який відкритий для всіх і може бути доступний та перевірений будь-яким учасником мережі. Його ключові особливості та переваги включають прозорість, безпеку та децентралізацію. Прикладами публічних блокчейнів є Ethereum, Bitcoin, Algorand.
- Приватний блокчейн — це тип блокчейну, який обмежений групою учасників, яким було надано дозвіл на доступ до мережі. Його ключові особливості включають конфіденційність та контроль. Однак недоліком є те, що вони вразливі для загроз безпеці, непрозорі, можуть бути дорогими в обслуговуванні та централізовані.

Різниця між публічними та приватними блокчейнами

	Public	Private
Доступність	Будь-хто може вільно приєднатися та брати участь в основних видах діяльності мережі блокчейн, включаючи читання, запис, додавання блоків та аудит діяльності мережі.	До мережі можуть приєднатися лише вибрані та перевірені учасники.
Прозорість	Прозорий, оскільки всі транзакції видно будь-кому в мережі.	Приватний, оскільки лише авторизовані користувачі можуть переглядати дані та транзакції у мережі.
Контроль	Децентралізований та керується спільнотою користувачів без жодної точки контролю. Після перевірки блоків запису не можна буде редагувати або видаляти.	Централізовані та контролюються одним суб'єктом або організацією. Оператор має право перевизначати, редагувати чи видаляти записи у блокчейні.



Різниця між базами даних та блокчейном

- Звичайна база даних просто зберігає та отримує дані.
- Блокчейн зберігає дані, отримує дані, підключається до однорангових пристроїв, перевіряє нові дані на відповідність уже існуючим правилам і передає цю інформацію по мережі, і робить це постійно.
- База даних зазвичай структурує свої дані у таблиці, тоді як блокчейн, як впливає з його назви, структурує свої дані у шматки (блоки), які пов'язані один з одним.

Різниця між базами даних та блокчейном

	Бази даних	Блокчейн
Централізація та децентралізація	Зазвичай база даних є централізованою, тобто всі дані зберігаються та керуються однією централізованою організацією або сутністю.	Блокчейн є децентралізованою технологією, де дані розподіляються та керуються різними вузлами або учасниками мережі без центрального контролю.
Механізм консенсусу	Контроль за даними та їхню цілісність зазвичай здійснюється централізованою системою або адміністратором бази даних.	У блокчейні механізм консенсусу дозволяє учасникам мережі досягати узгодженості щодо стану мережі та транзакцій, що забезпечує цілісність даних.
Цілісність даних	У базі даних дані можуть бути змінені або видалені централізованою владою відповідно до прав доступу та політик безпеки.	Дані в блокчейні незмінні. Після того, як дані додані до блокчейну, їх вже не можна змінити або видалити без згоди більшості учасників мережі, що забезпечує цілісність даних.

Блокчейн

- Публічний реєстр транзакційних даних, представлений у вигляді ланцюжка блоків, розподілених через систему з багатьох вузлів.
- Набір правил для узгодження наступного блоку: консенсус. Мережа вибирає блок-пропонента, який потім поширює блок до мережевих вузлів. Ноди перевіряють блок на правильність транзакції та право пропозиції. Якщо правильно, блокчейн додається.
- Публічно перевіряється (прозорий), не потребує дозволу та захищений від несанкціонованого доступу.

Трилема блокчейну

- Трилема блокчейна відноситься до проблеми досягнення трьох найважливіших аспектів технології блокчейн: безпеки, масштабованості та децентралізації.
- Трилема передбачає, що оптимізація одного аспекту часто ставить під загрозу інші, що ускладнює досягнення всіх трьох одночасно.



Основною будь-якого блокчейна є здатність не контролюватись якимось одним вузлом. Це одна з найбільш важливих та одночасно найбільш затребуваних функцій, оскільки вона дозволяє існувати новим видам бізнесу.

Жодного державного контролю.



Галузі застосування блокчейну

Фінанси

- Криптовалюти та цифрові активи.
- Міжнародні перекази та розрахунки.
- Системи електронних платежів.

Логістика та поставки

- Управління ланцюгами постачання.
- Покращення транспортних послуг.
- Забезпечення легкого зворотного руху грошей під час платежів та документування фінансових транзакцій.

Інтернет речей (IoT)

- Безпечний обмін даними між пристроями.
- Відстеження та автоматизація процесів.
- Створення децентралізованих систем управління та моніторингу.

Галузі застосування блокчейну

Медицина

- Електронні медичні записи та ідентифікація пацієнтів.
- Відстеження лікарських засобів та медичного обладнання.
- Управління медичними даними та дослідженнями.

Нерухомість

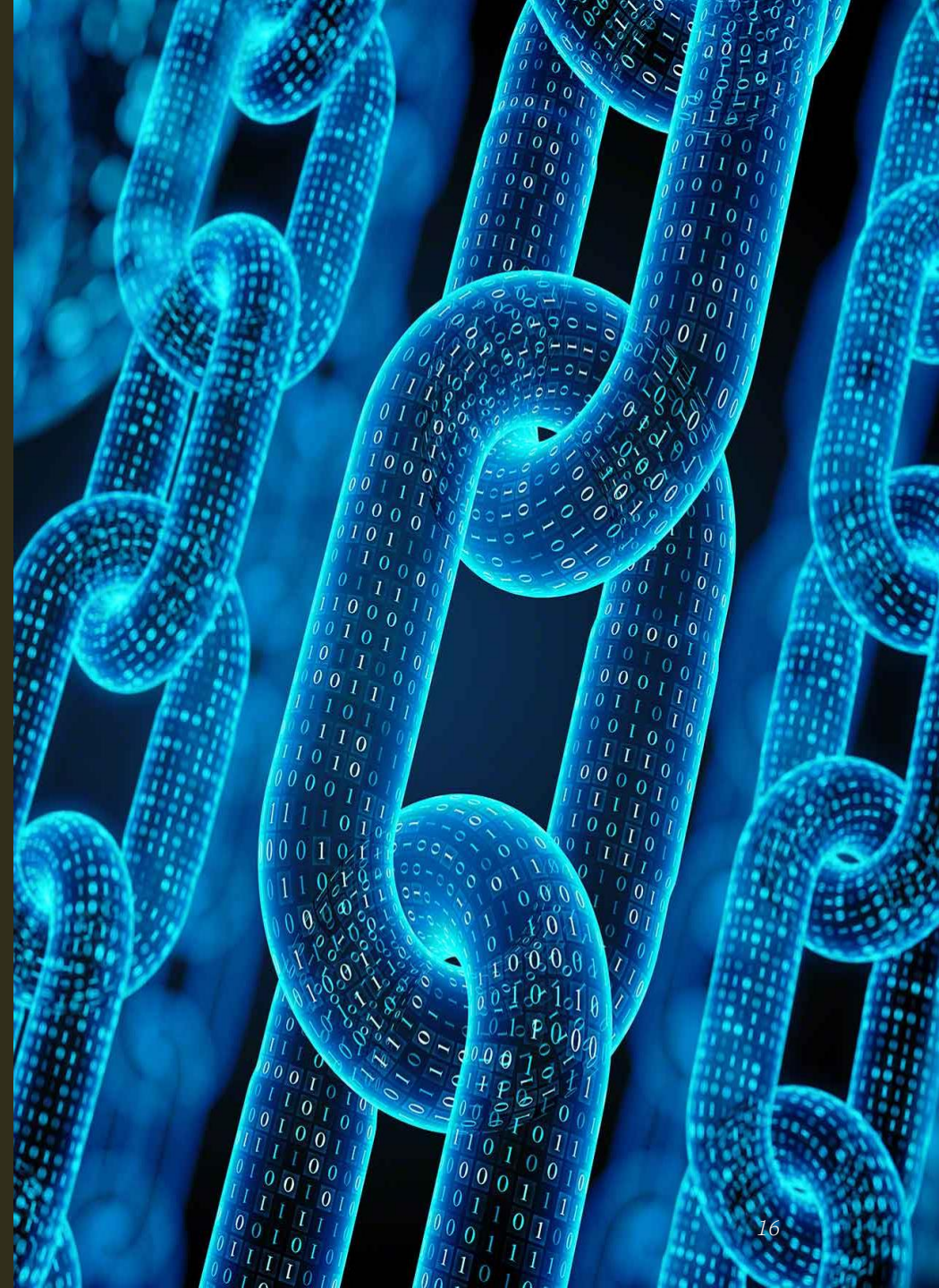
- Реєстрація власності та управління правами власності.

Голосування

- Електронні голосування.
- Підвищення прозорості та безпеки виборчих процесів.

Галузі застосування блокчейну

- Захист авторських прав
- Краудфандінг
- Реєстрація права власності на землю
- Страхування
- Туризм
- Освіта
- ЗМІ та розваги





Algorand - це децентралізована мережа, створена для вирішення трилеми блокчейна: одночасного досягнення швидкості, безпеки та децентралізації. Algorand, був запущений у червні 2019 року вченим-комп'ютерником і професором Массачусетського технологічного інституту Сільвіо Мікалі. Алгоранд є мережею блокчейнів з відкритим вихідним кодом, на якій може будувати кожен.

Algorand використовує механізм консенсусу Proof-of-Stake (PoS) та розподіляє винагороди валідаторів усім власникам своєї власної криптовалюти ALGO.

Особливості Algorand

1. **Протокол Pure Proof of Stake (PPoS):** Algorand використовує консенсусний механізм PPoS, який дозволяє досягти високої швидкості транзакцій та високої ступені безпеки. Кожен власник tokenів Algorand може брати участь у процесі прийняття рішень, використовуючи свої токени як голоси.
2. **Масштабованість:** Algorand розроблений з урахуванням масштабованості. Протокол може обробляти тисячі транзакцій за секунду (TPS), що робить його одним з найшвидших блокчейнів.
3. **Безпека:** алгоритм консенсусу Algorand гарантує безпеку мережі, навіть при участі великої кількості учасників. Це досягається за допомогою випадкового вибору учасників для створення нових блоків та перевірки транзакцій.
4. **Децентралізація:** Algorand прагне забезпечити максимальний рівень децентралізації, уникнувши проблем, пов'язаних з централізованими учасниками чи групами. Це робить мережу стійкою до цензури і маніпуляцій.
5. **Відкритість:** Algorand повністю відкритий і не вимагає дозволів. Будь-яка людина в будь-якій точці світу, яка володіє Algos, може брати участь у консенсусі.

Особливості Algorand

1. **Algo:** як і кожен блокчейн, Алгоранд має свою валюту - Algo, яка відіграє вирішальну роль у стимулюванні належної поведінки мережі. Якщо ви володієте Algos, ви можете зареєструватися для участі в консенсусі, що означає, що ви братимете участь у процесі пропозиції та голосування за нові блоки. Algo також використовуються для сплати комісії за транзакції в мережі.
2. **Нізка вартість транзакції:** мінімальна комісія за транзакцію становить лише 1000 microAlgos або 0,001 Algos. У мережі Algorand немає поняття плати за газ, як в Ethereum.
3. **Інструмент для розробки:** Розробники можуть писати смарт-контракти на Python і використовувати один із чотирьох пакетів SDK (Python, JavaScript, Golang, Java) для підключення до мережевих ресурсів або програм.

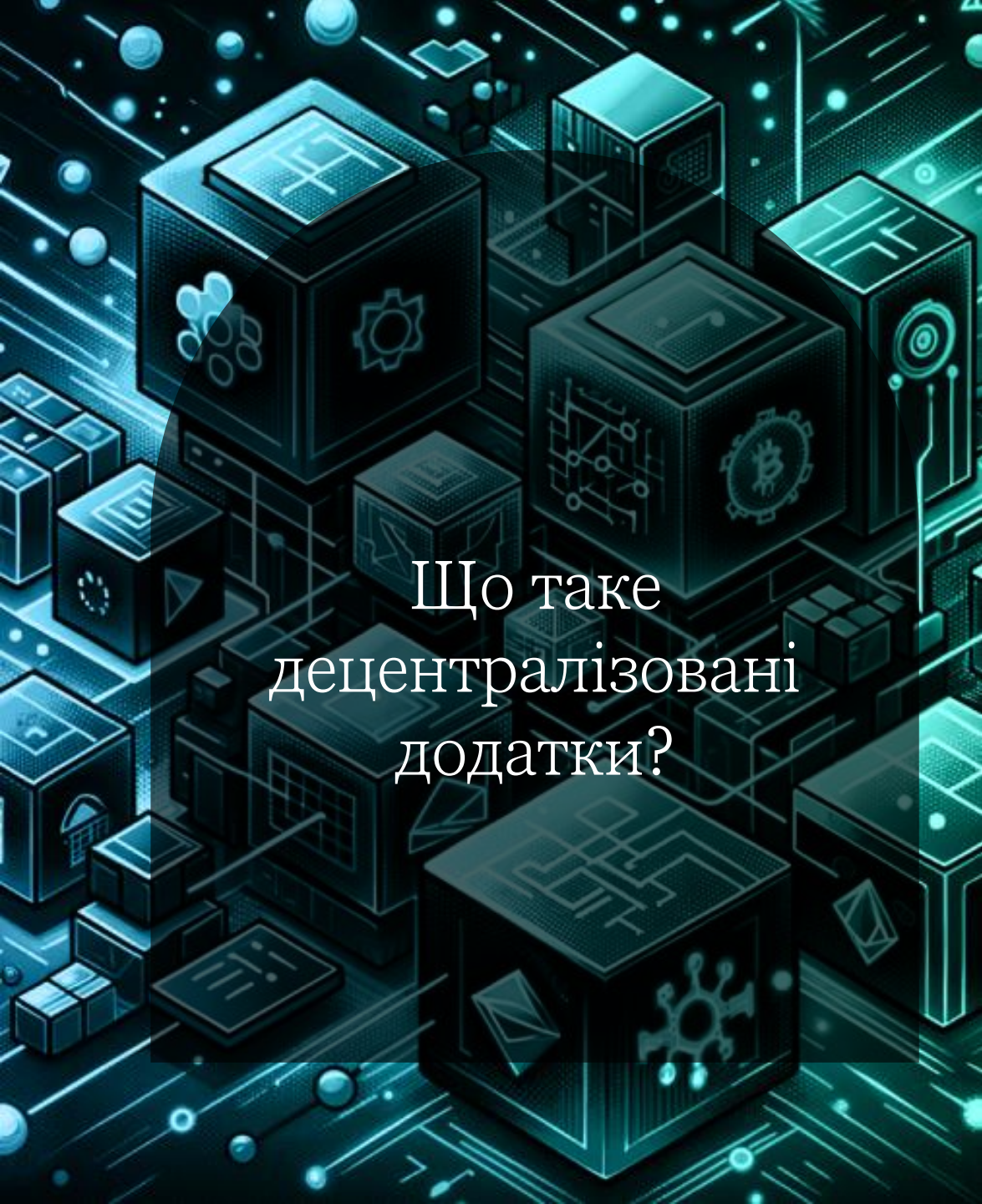
Протокол та механізм консенсусу

Протокол консенсусу - це набір правил і процедур, які визначають, як в узгоджений спосіб досягається згода між учасниками мережі щодо поточного стану системи та правильності виконаних транзакцій. В основі протоколу консенсусу лежить ідея забезпечення єдності даних та вирішення конфліктів у децентралізованих системах.

Механізм консенсусу - це конкретна реалізація протоколу консенсусу, яка використовується у певній системі або мережі для досягнення згоди між її учасниками. Існують різні механізми консенсусу, які можуть використовувати різні алгоритми, техніки та підходи. Прикладами механізму консенсусу є "Proof of Work", "Proof of Stake" тощо.



Pure Proof of Stake

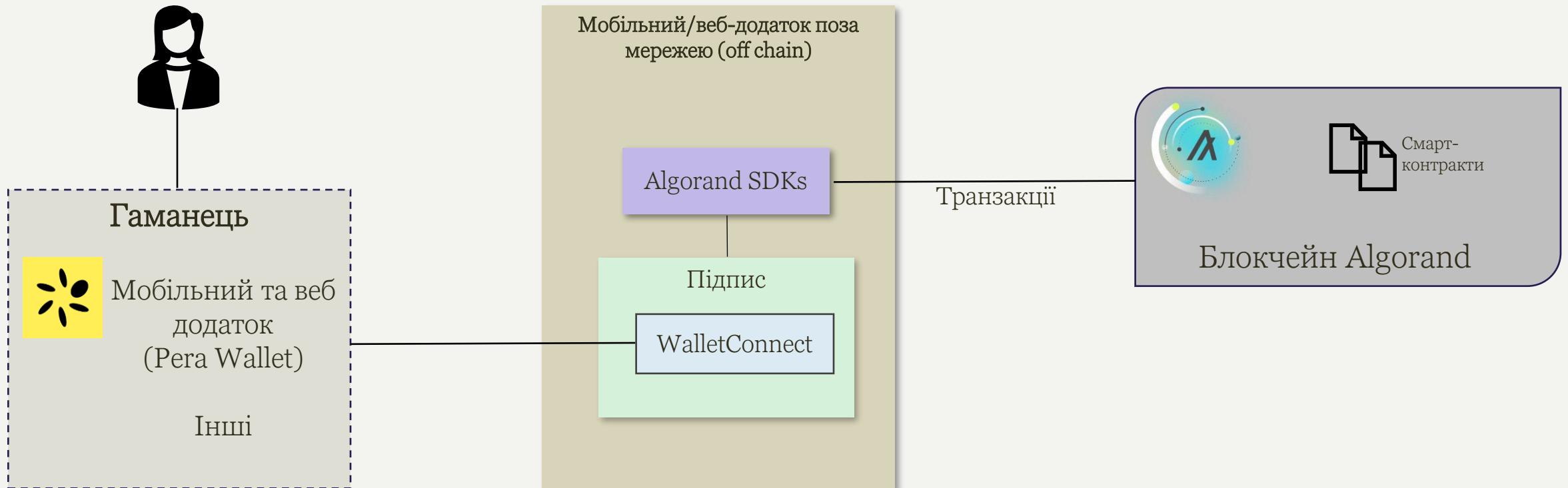


Що таке децентралізовані додатки?

Децентралізовані програми, або dApps, — це програми, які запускаються в децентралізованій обчислювальній системі, як-от блокчейн. Вони здебільшого або повністю децентралізовані.

Для більшості програм смарт-контракти будуть лише частиною архітектури dApp. Зазвичай розробники створюють у dApp функціональність, яка знаходиться на блокчейні, та деякий зовнішній інтерфейс для взаємодії зі смарт-контрактами.

Архітектура децентралізованого додатку в Algorand



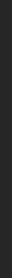


Що таке смарт-контракти?

Смарт-контракти - це програмні коди, що зберігаються на блокчейні і мають можливість автоматично виконувати угоди, зафіксовані в них. Вони дозволяють сторонам угоди автоматизувати та безпечно виконувати умови без необхідності посередників.

Розумні контракти не містять юридичних формулювань, умов або угод — лише код, який виконує дії, коли виконуються визначені умови.

ЖИТТЄВИЙ ЦИКЛ СМАРТ- КОНТРАКТУ





Життєвий цикл смарт-контракту відноситься до послідовності етапів, які проходить смарт-контракт від його створення до його припинення або завершення.

Цей життєвий цикл зазвичай включає такі етапи, як

- проектування,
- розробка,
- тестування,
- розгортання,
- виконання,
- аудит,
- модифікації
- завершення.

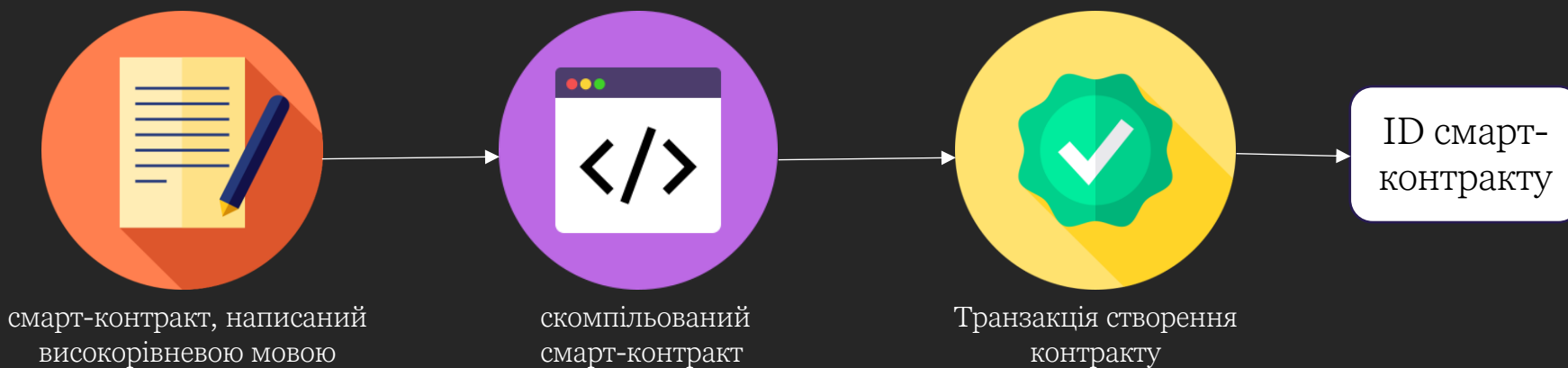
Кожен із цих етапів передбачає певні дії та вимоги, адаптовані до потреб і умов смарт-контракту.

Розумні контракти зазвичай пишуться мовою високого рівня, тип якої характеризується відповідною мережею блокчейнів. Щоб смарт-контракт був розгорнутий, він має бути скомпільований у низькорівневий код, який може зрозуміти віртуальна машина блокчейну.

Блокчейн	Мова
Ethereum	Solidity/Vyper
Algorand	PyTeal/Reach
Hyperledger Fabric	JavaScript/Java/Go

Після компіляції з коду, написаного мовою високого рівня, смарт-контракт можна розгорнути на платформі блокчейну за допомогою спеціальної транзакції для створення контракту.

З цієї транзакції буде повернуто унікальний ідентифікатор. Його можна використовувати для взаємодії з розгорнутим контрактом.



Смарт-контракт:

- запускається лише тоді, коли він викликається транзакцією.
- ніколи не працює у фоновому режимі сам по собі.
- може викликати інший контракт, який може викликати інший контракт.

Але перший виклик смарт-контракту виконується транзакцією від зовнішнього об'єкта (користувача).



Створення смарт-контрактів для блокчейну Algorand. Середовище розробки



Налаштування середовища

Під час налаштування середовища розробки ви можете вибрати один із трьох варіантів:

- **Algorand Sandbox:** найбільш використовуваним варіантом є налаштування пісочниці Algorand. Пісочниця дозволяє розробникам створювати локальні приватні мережі. Крім того, ви можете швидко видалити мережу, скинути її стан або створити нову мережу.
- **Сторонні API-сервіси:** можливо використовувати сторонні API-сервіси для доступу до власних Algorand REST API для основної мережі, тестової мережі та бета-мережі. Це чудовий вибір, якщо ви не хочете налаштовувати локальну мережу за допомогою Docker, а просто хочете спочатку поекспериментувати з розробкою Algorand.
- **Власний вузол:** можливо запустити власний вузол Algorand, який містить повну реалізацію програмного забезпечення Algorand. Це рішення є більш складним для налаштування та менш гнучким. На відміну від Algorand Sandbox, ви не можете викинути мережу та налаштувати нову, коли захочете. Налаштування вузла Algorand займає набагато більше часу, ніж налаштування локальної приватної мережі за допомогою інструментів Sandbox.



Algorand мережі

Algorand має три публічні мережі: MainNet, TestNet і BetaNet.

- **Mainnet:** основна мережа, яка використовується для створення реальних транзакцій і розгортання смарт-контрактів. Усі транзакції в мережі Mainnet є остаточними та мають реальну економічну цінність.
- **Testnet:** тестова мережа призначена для тестування та розробки. Вона імітує середовище Mainnet, але без реальної економічної цінності транзакцій. Розробники можуть тестувати свої додатки, не ризикуючи реальними активами.
- **Betanet:** служить раннім випробувальним майданчиком для нових функцій і оновлень, які ще не готові для Mainnet або Testnet. Його використовують переважно розробники, які хочуть експериментувати з останніми протоколами та оновленнями Algorand.



Algorand Sandbox

Algorand Sandbox - це інструмент, призначений для забезпечення спрощеного та простого у використанні середовища для розробки та тестування на блокчейні Algorand. Він використовує Docker для створення ізольованих контейнерів, що імітують мережу Algorand, що дозволяє розробникам експериментувати та створювати програми без необхідності налаштовувати повний вузол Algorand або підключатися до діючої мережі.

Основні характеристики Algorand Sandbox

- **Для навчання:** пісочниця призначена для навчання, а не для створення реальних додатків.
- **Ізольоване середовище:** за допомогою контейнерів Docker пісочниця забезпечує ізольоване середовище, де ви можете розробляти та тестувати свої програми, не впливаючи на активну мережу (Mainnet, Testnet..).
- **Попередньо налаштовані інструменти:** sandbox поставляється з попередньо налаштованими інструментами, такими як 'goal' (інструмент командного рядка Algorand) та іншими утилітами, необхідними для розробки.



Налаштування середовища

Необхідне програмне забезпечення:

Visual Studio Code, Docker (Docker Desktop for Windows), Linux Distribution for Windows

Додатково: Windows Terminal

Налаштування Algorand Sandbox

1. Відкрити термінал (термінал Ubuntu на Windows)
2. Відкрити потрібний локальний каталог та виконати наступні команди:

```
mkdir yourProjectName
```

```
cd yourProjectName
```

```
git clone https://github.com/algorand/sandbox.git
```

```
cd sandbox
```

```
./sandbox up
```

Для [Ubuntu and macOS](#)

Для [Windows](#)

ОСНОВНІ КОМАНДИ

Команди пісочниці:

up [config] -> запустити середовище пісочниці;

down -> знести середовище пісочниці;

reset -> скинути контейнери у вихідний стан;

enter [algod|conduit|indexer|indexer-db] -> увійти до контейнера пісочниці;

copyTo <file> -> скопіювати <file> в algod. Корисно для офлайн-транзакцій, офлайн роботи LogicSigs і TEAL.

copyFrom <file> -> копіювати <file> з algod. Корисно для офлайн-транзакцій, офлайн роботи LogicSigs і TEAL.

Більше команд sandbox: [Algorand Sandbox Usage](#)



Контейнери Sandbox

Пісочниця Algorand використовує контейнери Docker для імітації різних компонентів мережі Algorand:

- *algod Container*: контейнер algod запускає Algorand daemon, який є основним програмним забезпеченням, відповідальним за підтримку блокчейну Algorand. Він керує створенням блоків, перевіркою та консенсусними протоколами. Розробники використовують контейнер algod для локальної імітації середовища mainnet або testnet.
- *indexer Container*: контейнер indexer запускає службу Algorand Indexer, яка надає розширені можливості запитів для зафіксованих даних у блокчейні. Indexer індексує дані блокчейну, полегшуючи запит такої інформації, як історія транзакцій, баланси активів та інші дані про стан.

Приклади команд

- Запустити вузол із стандартною конфігурацією: `./sandbox up`
- Увійти в контейнер Algod (можна використовувати ту ж команду для входу до інших контейнерів, таких як контейнери `indexer` або `indexer-db`): `./sandbox enter algod`
- Зупинити пісочницю: `./sandbox down`
- Скопіювати файл у пісочницю: `./sandbox copyTo "example.teal"`

*якщо ви помістили файл `example.teal` в іншу папку, обов'язково передайте абсолютний шлях до цього файлу.

Тепер ця команда створить копію програми `example.teal` і помістить її в контейнер `algod`.

Приклад робочого процесу

./sandbox up (from sandbox path)

```
● olga@DESKTOP-QNI4VUN:~/pyTealStudy/sandbox$ ./sandbox up
```

```
Bringing up existing sandbox: 'release'
```

```
see sandbox.log for detailed progress, or use -v.
```

```
* docker containers started!
```

```
* waiting for services to initialize.
```

```
* services ready!
```

```
algod version
```

```
3298870427649
```

```
3.20.1.stable [rel/stable] (commit #6a6a15de)
```

```
go-algorand is licensed with AGPLv3.0
```

```
source code available at https://github.com/algorand/go-algorand
```

```
Indexer version
```

```
Dev Build compiled at 2024-01-01T19:42:34+0000 from git hash 3e9446aeb41010c514de43e8c2e1cd5e629f8773
```

```
Postgres version
```

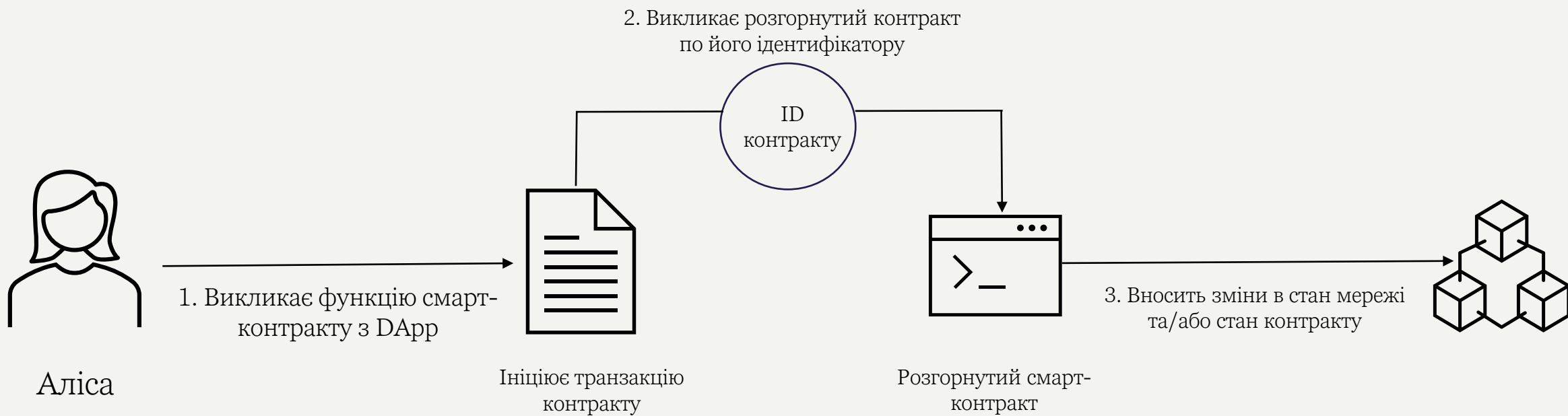
```
postgres (PostgreSQL) 13.13
```

Приклад робочого процесу

./sandbox goal account list

```
olga@DESKTOP-QNI4VUN:~/pyTealStudy/sandbox$ ./sandbox goal account list
[offline]      FXA6NGWY6MR2Y5Q7ZPFH7CC7OWHX5JAIED22235KH6RPK55GHUB5JH4054      FXA6NGWY6MR2Y5Q7ZPFH7CC7OWHX5JAIED22235KH6RPK55GH
UB5JH4054      999999999481000 microAlgos      [created app IDs: 1145, 1147, 1151, 1152, 1160, 1335, 1608, 1946]      [opted in
app IDs: 1152, 1160, 1335, 1608]
[online]      JLFS5K27S74MN2HF4C567UBK64LCJNPVE4DNKWX4W3ULGTNUVZH5QH2ML4      JLFS5K27S74MN2HF4C567UBK64LCJNPVE4DNKWX4W3ULGTNUV
ZH5QH2ML4      3999999999895000 microAlgos      [opted in app IDs: 1160, 1335, 1608]
[offline]      UW3N2WTVLZH0AZCNZCYTB0H4DSA7YSSNXYWVXZDIWGH34NII4ZDIUFZCGA      UW3N2WTVLZH0AZCNZCYTB0H4DSA7YSSNXYWVXZDIWGH34NII4
ZDIUFZCGA      4000000000000000 microAlgos      _
```

Команда повертає список всіх облікових записів, доступних у середовищі Algorand Sandbox. Зокрема, надає таку інформацію, як адреси рахунків та відповідні баланси (баланс, як правило, відображається в microAlgos (1 Algo = 1 000 000 microAlgos)).



Додаткові матеріали

- [How does a blockchain work](#) (відео)
- [What is Algorand?](#) (відео)
- [ALGORAND'S CORE TECHNOLOGY \(in a nutshell\)](#): стаття засновника Algorand Сільвіо Мікалі детальніше пояснює основну технологію Algorand та її мотивацію.
- [Introducing Sandbox: The quick way to get started on Algorand](#) (лекція)