Desarrollo de software para blockchain 2024

Clase 07

Conceptos

- Trilema de blockchain
- L2
- Bridge
- Multichain
- Polkadot
- Substrate
- Ink!
- Ejemplo práctico

Trilema blockchain

término popularizado por Vitalik Buterin, cofundador de Ethereum, describe el dilema enfrentado por las blockchains en términos de navegación entre tres propiedades principales: *seguridad, escalabilidad y descentralización*. Según el trilema, una blockchain no puede maximizar simultáneamente estas tres propiedades; siempre debe comprometer al menos una de ellas para fortalecer las otras dos.

L2: Layer 2

se refiere a una capa secundaria o sobrepuesta de solución que se construye "encima" de una blockchain existente (la capa base o Layer 1, L1), con el objetivo de escalar la red y permitir que se procesen más transacciones de manera más eficiente

Las blockchains, particularmente Ethereum, enfrentan problemas de escalabilidad, donde la red se vuelve más lenta y las tarifas de transacción aumentan cuando hay una alta demanda. L2 busca solucionar estos problemas sin cambiar la blockchain subyacente (L1), proporcionando soluciones que permiten la creación y ejecución de aplicaciones y contratos inteligentes de manera más eficiente.

L2: Categorías

1. Canales de Estado: Lightning Network en Bitcoin.

Los canales de estado permiten a los usuarios realizar transacciones entre ellos fuera de la cadena principal (off-chain) y luego finalmente asentar el estado final de todas las transacciones en la cadena principal.

2. Rollups: Optimistic Rollups, zk-Rollups.

Los rollups ejecutan transacciones fuera de la cadena y suben a la blockchain principal un resumen criptográfico de esas transacciones, junto con una prueba que puede ser verificada por otros participantes en la red.

> *Optimistic Rollups:* Las transacciones se ejecutan primero en L2 y las disputas sobre la validez de las transacciones se resuelven en L1.
>
> *zk-Rollups:* Todas las transacciones son válidas por defecto en L2 gracias al uso de pruebas de conocimiento cero (zk-SNARKs) que se verifican en L1.

L2: Categorías

3. Sidechains: xDai, Polygon (antes Matic).

Las sidechains son blockchains separadas que están pegadas a la blockchain principal. Los tokens y otros activos pueden ser movidos entre la cadena principal y la sidechain, pero la sidechain opera con un consenso independiente y, a menudo, con reglas y capacidades diferentes.

4. Plasma:

Plasma implica la creación de blockchains hijos que se derivan de la cadena principal. Los blockchains hijos (Plasma Chains) pueden procesar transacciones y contratos inteligentes en su propia red y luego publican el estado de esas transacciones en la cadena principal.

Bridge

Un bridge (puente) en el contexto de la tecnología blockchain se refiere a un protocolo que permite la transferencia de tokens y/o datos entre dos blockchains independientes. Los puentes buscan mejorar la **interoperabilidad** entre diferentes redes blockchain, permitiendo que los activos y la información se muevan de manera fluida entre diferentes plataformas.

Tipos de bridges

1- Token Bridges: Permiten la transferencia de tokens entre dos blockchains.

Los tokens pueden ser representaciones de activos existentes (por ejemplo, tokens estables, tokens de criptomonedas, tokens de activos reales) y se bloquean en una cadena mientras están activos en la otra.

2- Data Bridges: Facilitan la transferencia de datos e información entre blockchains.

Esto es crucial para la operación de oráculos descentralizados y para que los smart contracts en diferentes cadenas interactúen entre sí.

3- NFT Bridges: Permiten la transferencia de tokens no fungibles (NFTs) entre diferentes blockchains.

Los NFTs se bloquean en una cadena y se representan en otra para preservar la unicidad y la propiedad del activo.

Bridges: Mecanismos Comunes de Funcionamiento

- Locking and Minting: Un activo se bloquea en una blockchain y se crea (mintea) una representación de ese activo en la cadena de destino.

- Multi-Signature Wallets: Las transacciones a menudo requieren firmas de múltiples partes para validar la transferencia de fondos entre cadenas.

- Relayers: Los relayers son nodos que facilitan la comunicación entre las blockchains involucradas en el puente, a menudo retransmitiendo pruebas o eventos de una cadena a otra.

- Smart Contracts: Los puentes utilizan contratos inteligentes para administrar el bloqueo, minteo y la quema de tokens, y para validar las pruebas y datos de la cadena opuesta.

Concepto de Multichain

se refiere a la interoperabilidad y la interacción entre múltiples blockchains.

- Interoperabilidad: La capacidad de interactuar e intercambiar datos y valor entre diferentes blockchains.

- Escalabilidad: Utiliza múltiples cadenas de bloques para distribuir la carga y mejorar la escalabilidad del sistema.

- Optimización: Permite la utilización de diferentes blockchains para diferentes usos, cada una optimizada para un caso de uso particular. Gaming, DeFi, etc

Polkadot

es una plataforma de blockchain diseñada para permitir la interoperabilidad entre múltiples blockchains. Fue desarrollada por el Dr. Gavin Wood, uno de los co-fundadores de Ethereum, y está siendo construida por la Web3 Foundation y Parity Technologies.

whitepaper

Polkadot: arquitectura

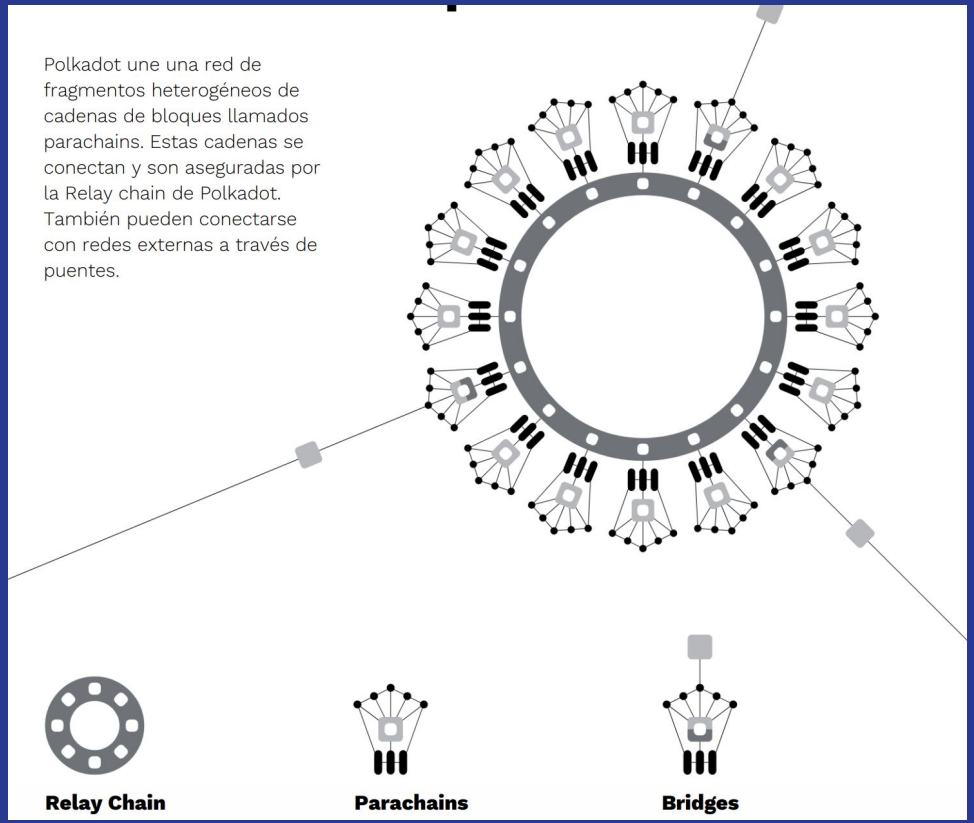
1. Relay Chain: La blockchain principal de Polkadot, que garantiza la seguridad y la interoperabilidad entre las blockchains conectadas a la red.

2. Parachains: blockchains individuales que se ejecutan en paralelo, permitiendo múltiples transacciones y operaciones en diferentes cadenas simultáneamente, proporcionando escalabilidad.

3. Bridges: Permiten la conexión e interacción con blockchains externas (como Ethereum y Bitcoin), posibilitando la transferencia de mensajes y valor de una manera segura y fiable a través de cadenas.

4. Parathreads: Similares a las parachains pero con un modelo de pago-por-uso, lo que permite una mayor flexibilidad para los desarrolladores que no necesitan un slot continuo en la red.

Polkadot: arquitectura



Polkadot: algoritmo de consenso

Utiliza un mecanismo de consenso híbrido que combina elementos de consenso de Proof of Stake (PoS) y Byzantine Fault Tolerance (BFT). Específicamente, utiliza dos sistemas principales de consenso:

1- *Nominated Proof of Stake (NPoS):*

-Validators: Son responsables de la producción de bloques y la confirmación de las transacciones.

-Nominators: Ayudan a asegurar la red al nominar validadores, optando por aquellos en los que confían para actuar honestamente.

Los validadores y los nominadores son recompensados con tokens DOT por su papel en la seguridad de la red y son penalizados por el mal comportamiento o las fallas de seguridad.

Polkadot: algoritmo de consenso

2- *GRANDPA (GHOST-based Recursive Ancestor Deriving Prefix Agreement)*:

GRANDPA es el algoritmo de finalización de BFT utilizado en Polkadot. Asegura que las transacciones son irreversibles una vez que son confirmadas.

Proporciona una finalización eficiente y puede finalizar múltiples bloques a la vez si se produce un particionamiento de la red.

GRANDPA

Polkadot: curva elíptica

En cuanto a la criptografía de clave pública, Polkadot y su blockchain hermana, Kusama, permiten tres tipos de criptografía de clave pública para las direcciones de cuenta:

1- *Edwards-curve Digital Signature Algorithm (EdDSA) con la curva Ed25519*: Es conocida por ofrecer un buen balance entre seguridad y rendimiento.Utiliza una longitud de clave de 256 bits.

2- *Schnorrkel (Sr25519):* Una variante de Schnorr en la curva Twisted Edwards (Ed25519). Ofrece robustez y eficiencia en términos de desempeño y seguridad. Schnorrkel utiliza también la curva Ed25519 y añade la capacidad de utilizar firmas múltiples y de umbral.

Polkadot: Substrate

Es un framework de desarrollo de blockchain creado por Parity Technologies, diseñado para facilitar la creación de blockchains personalizadas. Substrate proporciona las bases sobre las que se construye Polkadot, y ha sido diseñado para ser extremadamente flexible, permitiendo a los desarrolladores construir una amplia variedad de blockchains y aplicaciones descentralizadas (dApps). Aunque Substrate y Polkadot están estrechamente relacionados, Substrate se puede usar para crear blockchains independientes que no estén en la red Polkadot.

Substrate

Polkadot: Substrate, algunas características

1. Modularidad:

Runtime Module Library (FRAME): Los desarrolladores pueden utilizar módulos preexistentes (**pallets**) para funciones como balances, prueba de participación (PoS), y contratos inteligentes, o pueden crear sus propios módulos.

2. Flexibilidad en el Consenso:

Consensos Integrados: Ofrece varios algoritmos de consenso listos para usar, como Aura, BABE, y GRANDPA.

Consenso Personalizable: También se pueden implementar mecanismos de consenso propios.

Polkadot: Substrate, algunas características

3. Interoperabilidad:

Conectividad con Polkadot: Si se desea, una blockchain basada en Substrate puede conectarse a Polkadot y beneficiarse de la seguridad compartida y la interoperabilidad entre cadenas.

Compatibilidad entre Blockchains: A través de la utilización de puentes, las blockchains creadas con Substrate pueden interactuar con otras redes.

4. Desarrollo Eficiente:

Innovación de "Hot-Upgrade": Permite actualizar el protocolo blockchain sin realizar un hard fork.

Soporte para WASM: WebAssembly permite la ejecución de nuevas versiones del blockchain y simplifica el proceso de actualización.

Polkadot: Substrate, algunas características

5. Soporte para Contratos Inteligentes:

Ink!: Es un eDSL (embedded Domain Specific Language) para el desarrollo de contratos inteligentes en Rust, optimizado para Substrate.

Compatibilidad con EVM: Substrate puede configurarse para ser compatible con la Ethereum Virtual Machine (EVM), lo que permite ejecutar contratos inteligentes escritos para Ethereum.

6. Escalabilidad:

Parachains: Al conectarse a Polkadot como una parachain, una blockchain Substrate puede operar en paralelo con otras blockchains, mejorando la escalabilidad.

Polkadot: Substrate, algunas características

7. Seguridad:

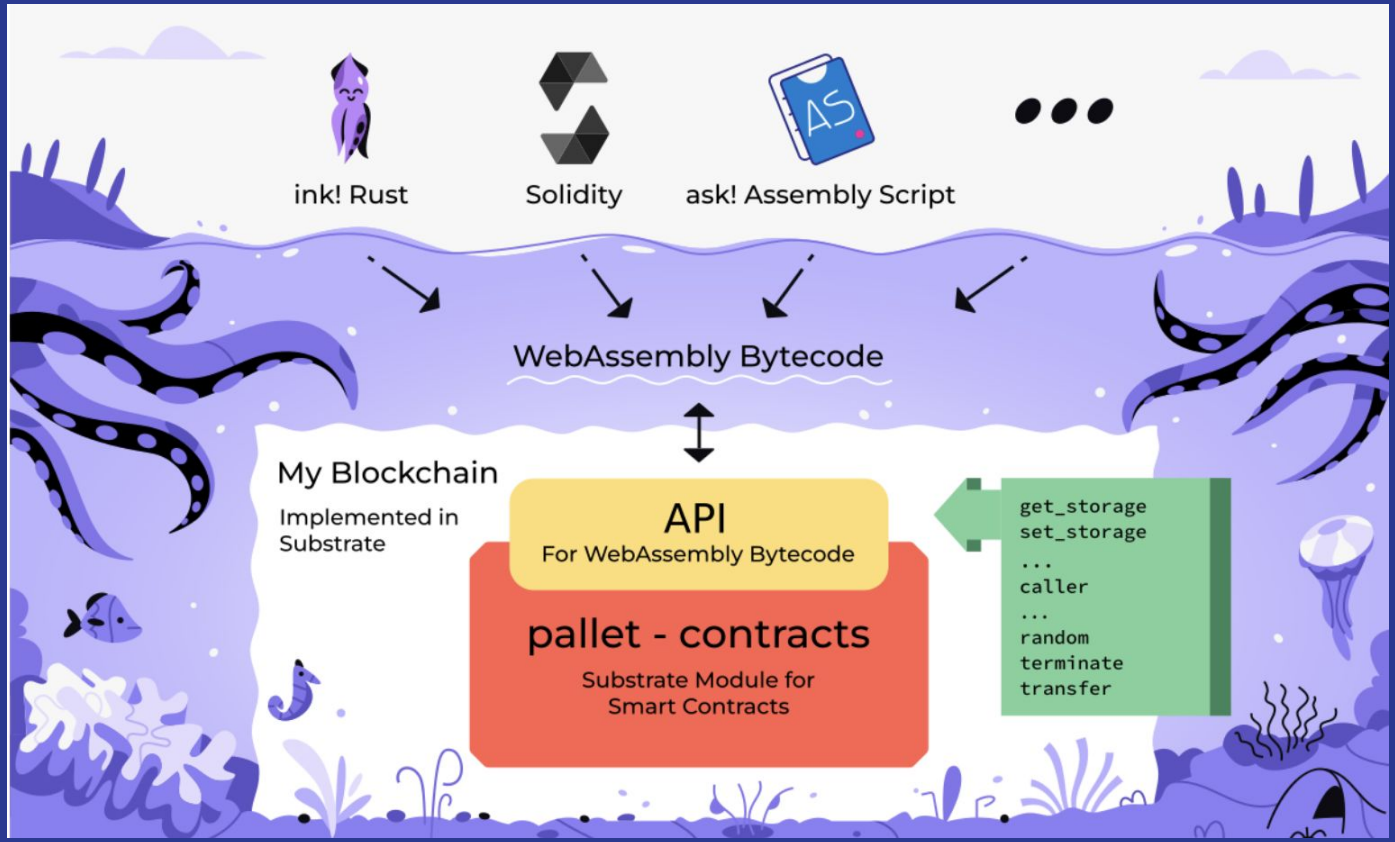
Seguridad en la Red Polkadot: Al unirse a Polkadot, una cadena Substrate puede aprovechar la seguridad proporcionada por los validadores de la Relay Chain de Polkadot.

Polkadot: Substrate, Ink!

Ink! es un eDSL (embedded Domain Specific Language) desarrollado por Parity Technologies para la creación de contratos inteligentes en la plataforma blockchain de Substrate. Está escrito en el lenguaje de programación Rust y es utilizado para desarrollar contratos inteligentes que se ejecutan en blockchains construidas con el framework Substrate.

Ink!

Polkadot: Substrate, Ink!



Ink! Algunas características

1. Escrito en Rust:

ink! permite a los desarrolladores escribir contratos inteligentes utilizando la sintaxis y las capacidades del lenguaje Rust, que es conocido por su seguridad y rendimiento.

2. Optimizado para Contratos Inteligentes:

Aunque los contratos se escriben en Rust, ink! proporciona macros y herramientas que facilitan el desarrollo específico de contratos inteligentes y optimizan el código para ejecutarse en un entorno de blockchain.

Ink! Algunas características

3. Interacción con Substrate:

Los contratos inteligentes escritos con ink! pueden ser desplegados y ejecutados en cualquier blockchain desarrollada con Substrate, aprovechando su interoperabilidad y características de seguridad.

4. WASM:

ink! compila los contratos inteligentes a WebAssembly (WASM), permitiendo que se ejecuten en Substrate y sean interoperables con otras plataformas que soportan WASM.

Ink! Algunas características

5. Seguridad: Rust e ink! poseen características que favorecen la escritura de código seguro y robusto, reduciendo la posibilidad de errores y vulnerabilidades en los contratos inteligentes.

6. Manejo Eficiente de Recursos: Los contratos escritos en ink! pueden gestionar los recursos de manera eficiente gracias a las capacidades de bajo nivel y manejo de memoria de Rust.

7. Testeo de Contratos: también proporciona herramientas para realizar unit testing en los contratos inteligentes, facilitando el desarrollo y la validación de la lógica del contrato antes de su despliegue en la cadena de bloques.

Ink! Instalación

https://github.com/paritytech/cargo-contract

1: `rustup component add rust-src`

2: `cargo install --force --locked cargo-contract`

3: (Optional) Install dylint for linting.

(MacOS) brew install openssl

`cargo install cargo-dylint dylint-link`

Ink! comandos

```
$ cargo contract
Utilities to develop Wasm smart contracts

Usage: cargo contract <COMMAND>

Commands:
  new          Setup and create a new smart contract project
  build        Compiles the contract, generates metadata, bundles both together in a
`<name>.contract` file
  check        Check that the code builds as Wasm; does not output any `<name>.contract` artifact
to the `target/` directory
  test         Test the smart contract off-chain
  upload       Upload contract code
  instantiate  Instantiate a contract
  call         Call a contract
  decode       Decodes a contracts input or output data (supplied in hex-encoding)
  help         Print this message or the help of the given subcommand(s)

Options:
  -h, --help     Print help information
  -V, --version  Print version information
```

Ink! Ejemplo práctico

Links y comandos útiles

comandos:

cargo contract build

en caso de error de opcode 192, instalar nightly-2023-02-07 => https://stackoverflow.com/questions/58226545/how-to-switch-between-rust-toolchains

cargo +nightly-2023-02-07 contract build

cargo test

cargo tarpaulin --target-dir src/coverage --skip-clean --exclude-files=target/debug/* --out html

cargo test --features e2e-tests

verificar contrato: https://medium.com/chainlens/how-to-verify-ink-smart-contracts-83fec5de81aa

wallet: https://polkadot.js.org/extension/

testnet: https://substrate.io/developers/rococo-network/

ui deploy contracts: https://contracts-ui.substrate.io/

faucet: https://app.element.io/#/room/#rococo-faucet:matrix.org => escribir !drip PONER_TU_ADDRESS:1002 en el chat

explorer: https://polkadot.js.org/apps/?rpc=wss%3A%2F%2Frococo-contracts-rpc.polkadot.io