INPUT | OUTPUT

On-chain VS Off-chain

Descripción general de los roles y características de código on-chain y off-chain de smart contracts en Cardano

Qué quiere decir on- y off-chain?

Roles de código on-chain y off-chain en un smart contract

- **On-chain:** Código que corre en el nodo durante la adición de nuevas transacciones e información que se guarda en la blockchain.
- **Off-chain:** Código que corre en el dispositivo del usuario (o proveedor de servicio) para consultar la blockchain y construir y enviar transacciones.

¿Por qué es necesario descomponer nuestro código?

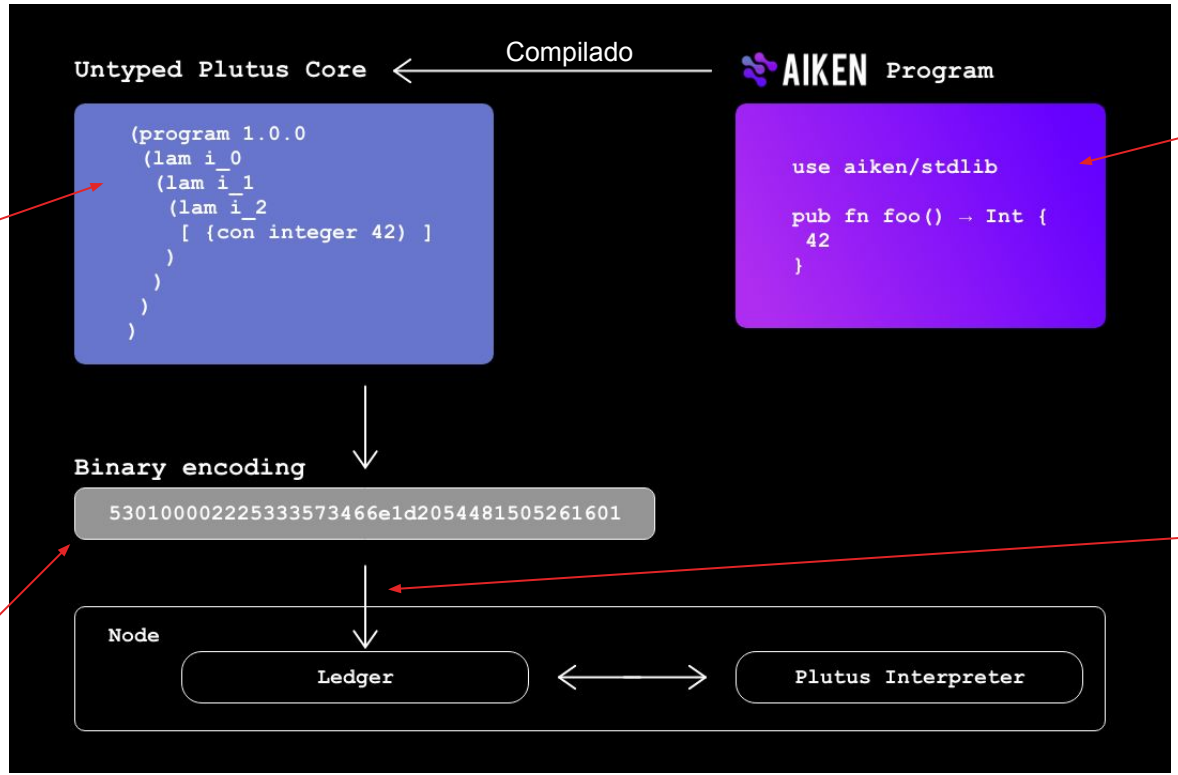
Porque tanto el código on-chain como off-chain tienen ventajas y limitaciones inherentes

- **On-chain:** Asegura integridad, pero es caro.
- **Off-chain:** No garantiza la integridad, pero tiene acceso a la billetera y los recursos del usuario.

Compilar código on-chain

UPLC es el lenguaje que los nodos saben interpretar y, por lo tanto, el que se ejecuta en la blockchain

Se hace el encoding de UPLC a binario antes de enviar.



PlutusTx
Plutarch
Plu-ts
Scalus
Opshin

Código Off-chain

+ OFF-CHAIN _

Introducción a modelo UTxO

```
cardano-cli query tip
                  utxo
                  <otro>
```

Consultando la blockchain

Obteniendo todo lo que necesitamos

- Al ser determinística, tenemos que determinar todos los detalles de la transacción de antemano:
  - El contexto de la transacción.
  - Todas sus entradas y salidas (con sus respectivos Valores, Datums, Redeemers, y Scripts).
  - Etc.

Hay varias maneras de obtener esta información

- Usando un Nodo local
- Usando una base de datos local sincronizada con un nodo que puede ser local o remoto con un blockchain indexer (e.g., db-sync, Kupo, Scrolls, Carp,...)
- Usando herramientas de terceros que ejecutan sus propios nodos y bases de datos. (e.g., Maestro)

```
cardano-cli address build


cardano-cli transaction build
                          build-raw
```

Construyendo la Tx (“Desplegar” validador)

- Primero, necesitamos la dirección del script.
- Indicar los inputs y outputs
- Proporcionar el Datum
  - Solo el Hash de Datum en la salida
  - Hash de Datum en salida + Datum en cuerpo de Tx
  - Inline Datum (el datum en el UTxO)
- Proporcionar el script
  - Sólo el hash del script
  - El script (que se utilizará como script de referencia)
- Selección de monedas, cálculo de fees y balanceo de transacción

```
cardano-cli transaction build
                        build-raw
```

Construyendo la Tx (“Usando” el validator)

- Indicar los inputs, inputs de referencia, y los outputs.
- Proporcionar Datum
  - UTxO tiene el hash del dato -> Proporcionar el Datum en Tx
  - UTxO tiene datos en el cuerpo de Tx -> Encontrarlo de la blockchain y proporcionalo en la Tx
  - UTxO tiene un inline Datum -> Se indica que es un inline Datum (no es necesario proporcionarlo)
- Proporcionar Script
  - Ningún UTxO tiene el script adjunto -> Proporcionamos el script en la Tx
  - El script fue adjuntado a un UTxO -> Proporcionar referencia del UTxO.
- Proporcionar Redeemer
- Proporcionar colateral

```
cardano-cli transaction sign
```

Off-chain: Firmando la Tx

Tener en cuenta

- Todas las transacciones deben estar firmadas con al menos una llave.
- Este es el único paso que DEBE realizarse en el lado del cliente (billetera del usuario)
- Las credenciales deben almacenarse en una billetera criptográfica (dispositivo del usuario)
- Hay que proporcionar una forma de conectar su código fuera de la cadena con billeteras

```
cardano-cli transaction submit
```

Off-chain: Enviando la Tx

Fase 1

- Comprueba si la Tx se creó correctamente y se puede agregar al ledger.
- Si falla, se rechaza Tx (no se cobran fees ni colateral)

Fase 2

- Tenemos todo lo que necesitamos. ¡Ejecutemos los scripts!
- Si tiene éxito, la Tx se agrega al ledger y las fees se utilizan para pagar al nodo.
- Si falla, se ignora la Tx y se utiliza el colateral para pagar al nodo.

Determinismo

- Efectos predecibles -> Verificar fases off-chain


Fin
Preguntas?
INPUT | OUTPUT