Aleo | Workshop

Compliant Private Tokens

Agenda

1 Logistics

2 What is a Token?

3 Privacy & Compliance

4 What is Aleo?

5 Tokens on Aleo

6 Compliance on Aleo

7 Q&A

8 Hands-on Challenge

What You’ll Need

Leo Playground: https://**play.leo-lang.org/**

OR

IDEs: VSCode / Sublime Text / IntelliJ

- Install Rust
- Install Leo
- Install Leo Extension for your IDE

Workshop: https://github.com/**alex-aleo/private-token-workshop**

Other Resources

Aleo Developer Docs: https://**developer.aleo.org**

Leo Docs: https://**docs.leo-lang.org**

Discord: https://discord.gg/**aleo**

Devs Telegram:

What is a Token?

- Digital assets representing value or utility onchain
- **Fungible Tokens**:
  - Units are interchangeable with each other
  - **Stablecoins**, ERC-20 tokens, Bitcoin
- **Non-Fungible Tokens (NFTs)**:
  - Units are unique and possess distinct characteristics or value
  - Digital art, collectibles, virtual real estate, event tickets

What is a Token?

- **Tokenomics:**
    - The economics of a token, governing its creation, distribution, and usage
    - **Minting:**
        - Process of creating new tokens
    - **Burning:**
        - Removing tokens from circulation, often to reduce supply
    - **Transferring:**
        - Sending a token to another user or service

Privacy

- **Major problem:**
  - Blockchains are fully public ledgers
  - All holdings and transactions are publicly visible
    - Surveillance in **perpetuity**
- Example:
  - Etherscan

Privacy

- **Why is privacy important?**
  - Blockchain gives true ownership and control over assets
    - **Does not guarantee privacy**
  - Private ≠ Illegal
    - Provides stronger security and efficiency
    - Reduces need to store data all in one place
      - Lowers risk of single point of failure.

Privacy

- **Why is privacy important?**
    - Payments with stablecoins necessitate privacy
        - Users shouldn’t have to share everything they pay for
        - Business shouldn’t be able to see their competitors’ expenditures
- **Caveat:**
    - Any payments must also maintain regulatory compliance

Compliance

- **KYC / AML**
    - KYC = Know Your Customer
    - AML = Anti-Money Laundering
    - Set of regulations designed to prevent illicit activities
        - Fraud, money laundering, terrorist financing, etc.
    - Affected businesses must verify the identity of their clients
        - ex) Coinbase requires ID upload to open an account

Compliance

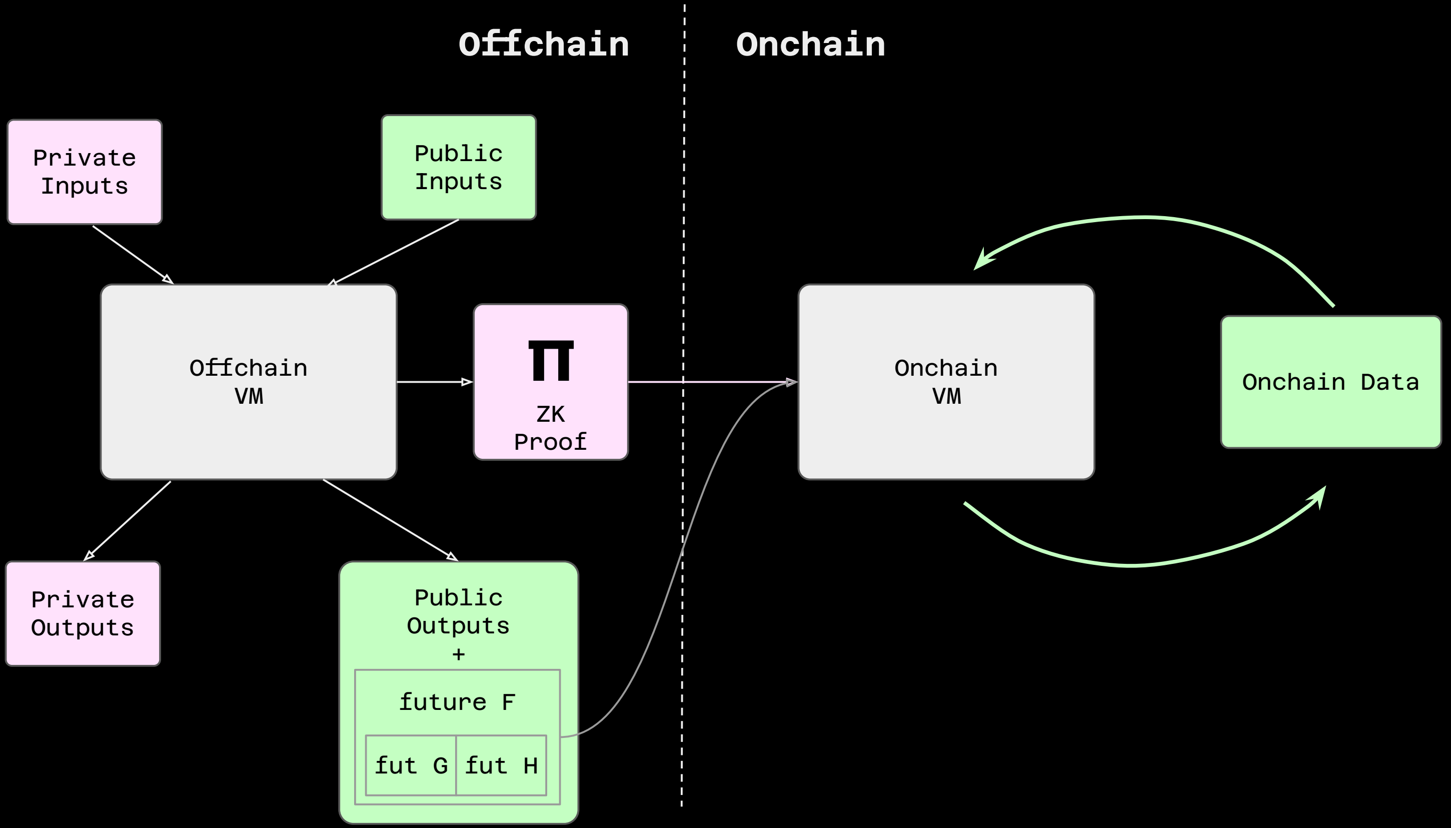
- **Sanctioned Address List**
  - Maintained by the Office of Foreign Assets Control (OFAC)
    - U.S. Treasury Department
  - List of cryptocurrency addresses associated with sanctioned entities

How can we maximize privacy while maintaining regulatory compliance?

Aleo

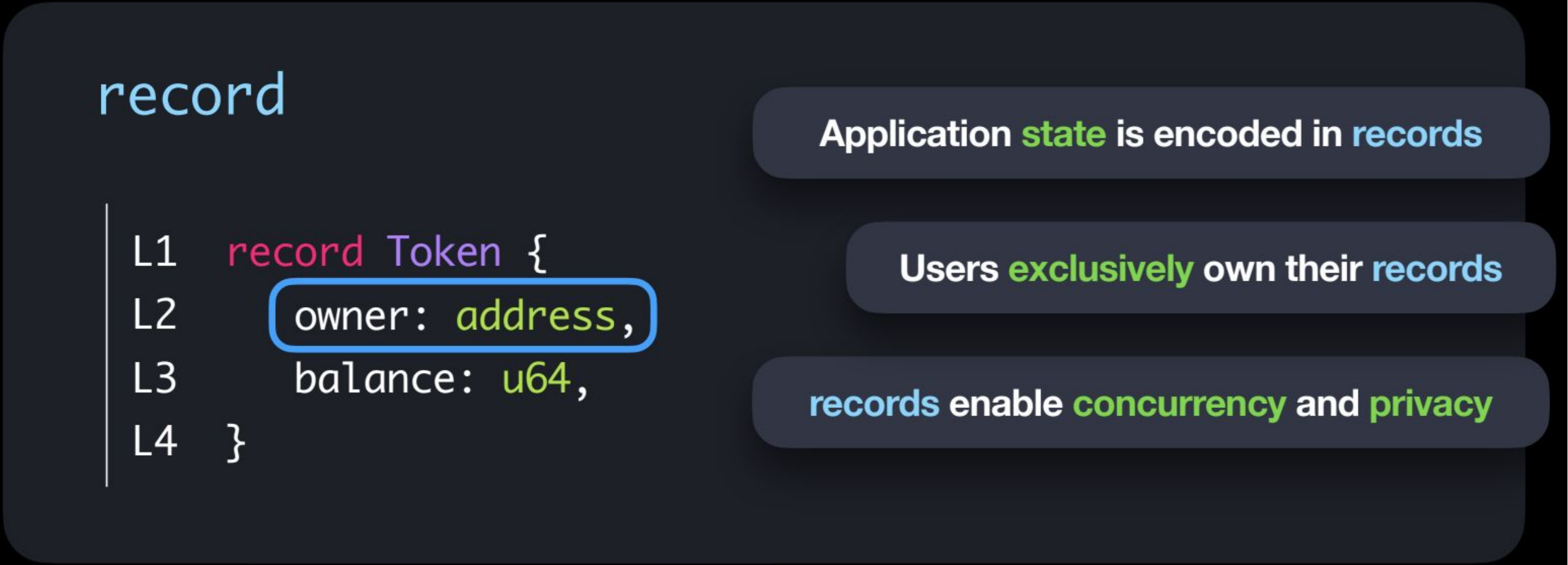
- Layer 1 blockchain with privacy as a first-class citizen
  - Powered by zero-knowledge proofs
- Privacy is **programmable**, so developers can choose what gets revealed
  - Bake-in compliance without sacrificing user privacy

Aleo Model


Offchain
Onchain
Private Inputs
Public Inputs
Offchain VM
π
ZK Proof
Onchain VM
Onchain Data
Private Outputs
Public Outputs +
future F
fut G
fut H

Tokens on Aleo

- Offchain State



record

```
record Token {
    owner: address,
    balance: u64,
}
```

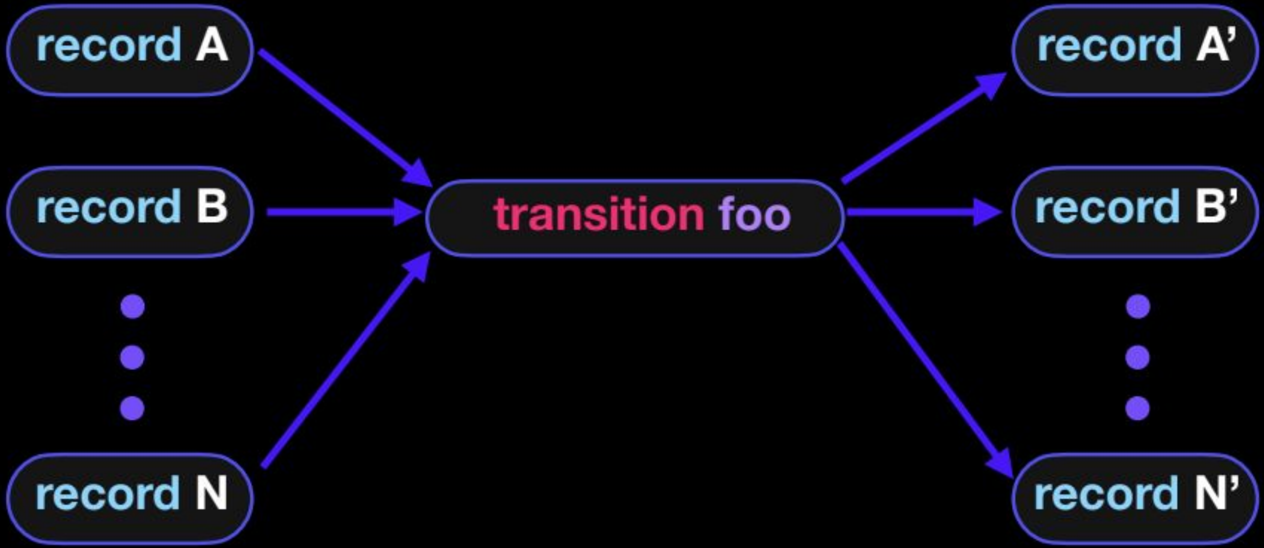
Application state is encoded in records

Users exclusively own their records

records enable concurrency and privacy

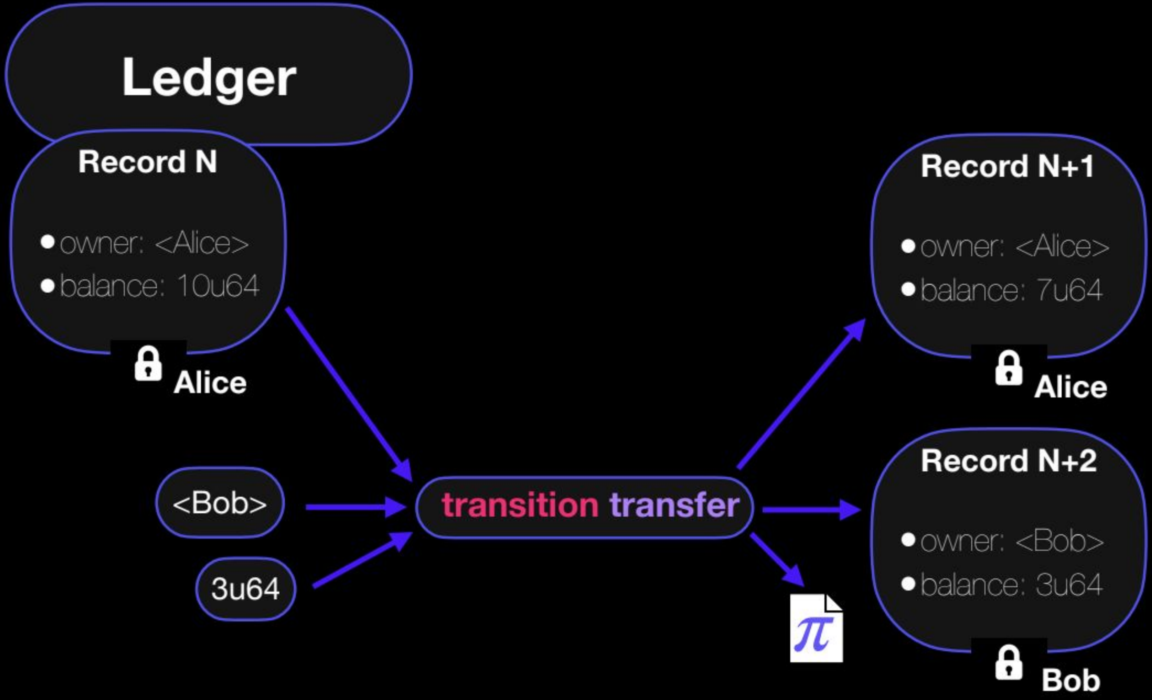
Tokens on Aleo

- Using records



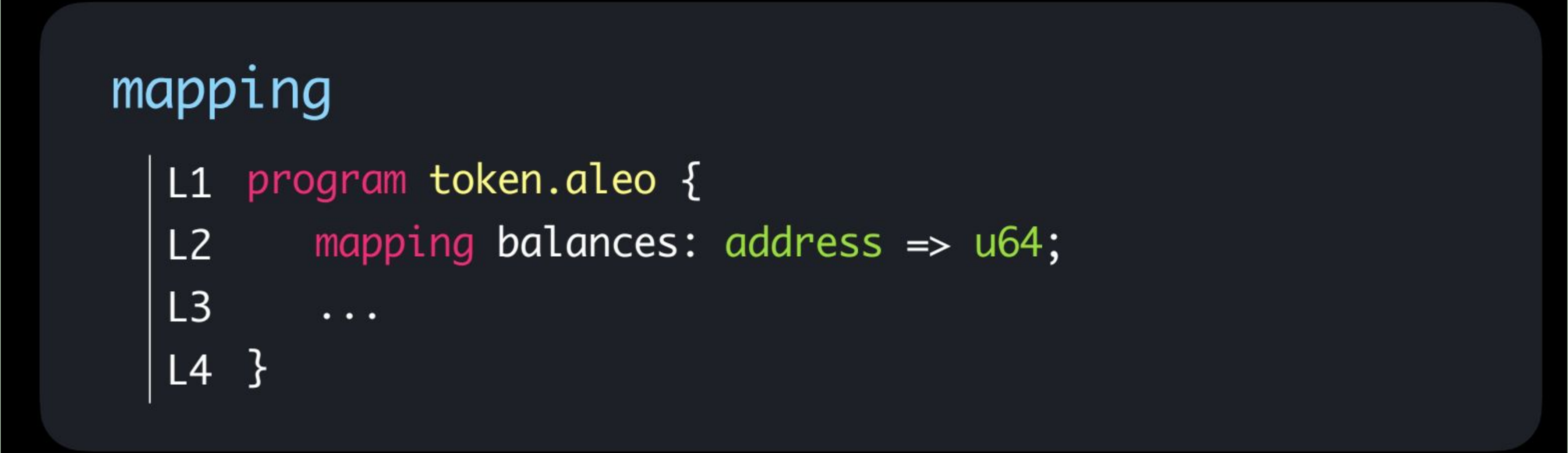
Tokens on Aleo

- Using records



Tokens on Aleo

- Onchain State

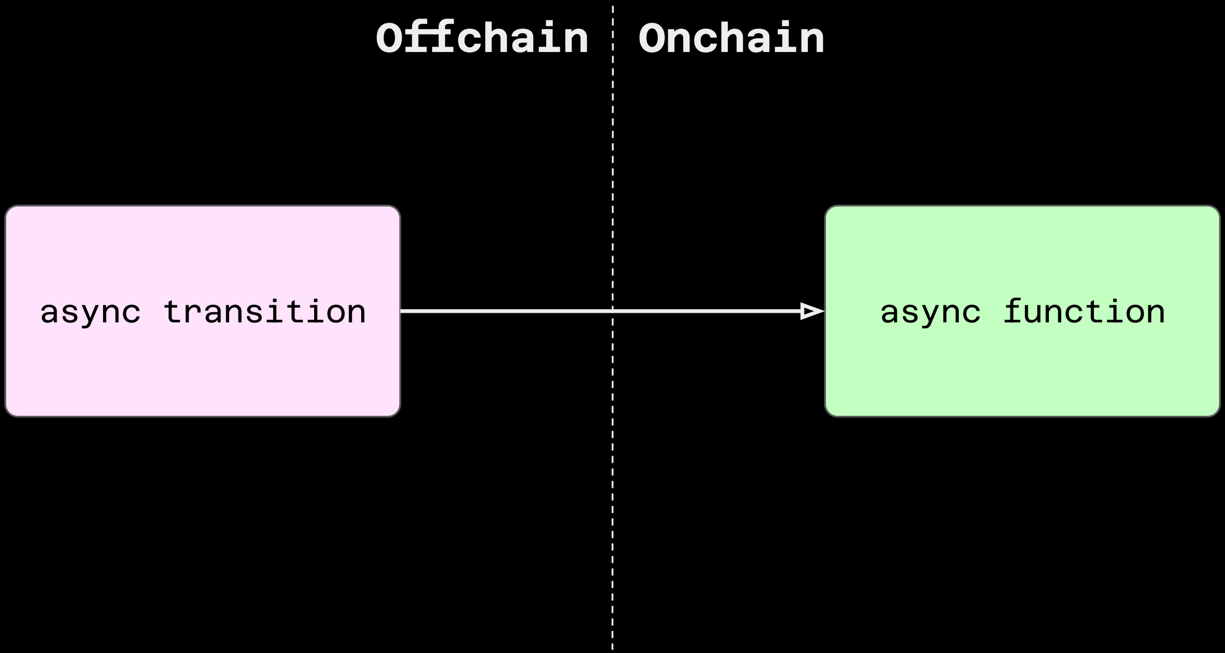


mapping

```
program token.aleo {
    mapping balances: address => u64;
    ...
}
```

Tokens on Aleo

- Modifying Onchain State
  - The async model



Tokens on Aleo

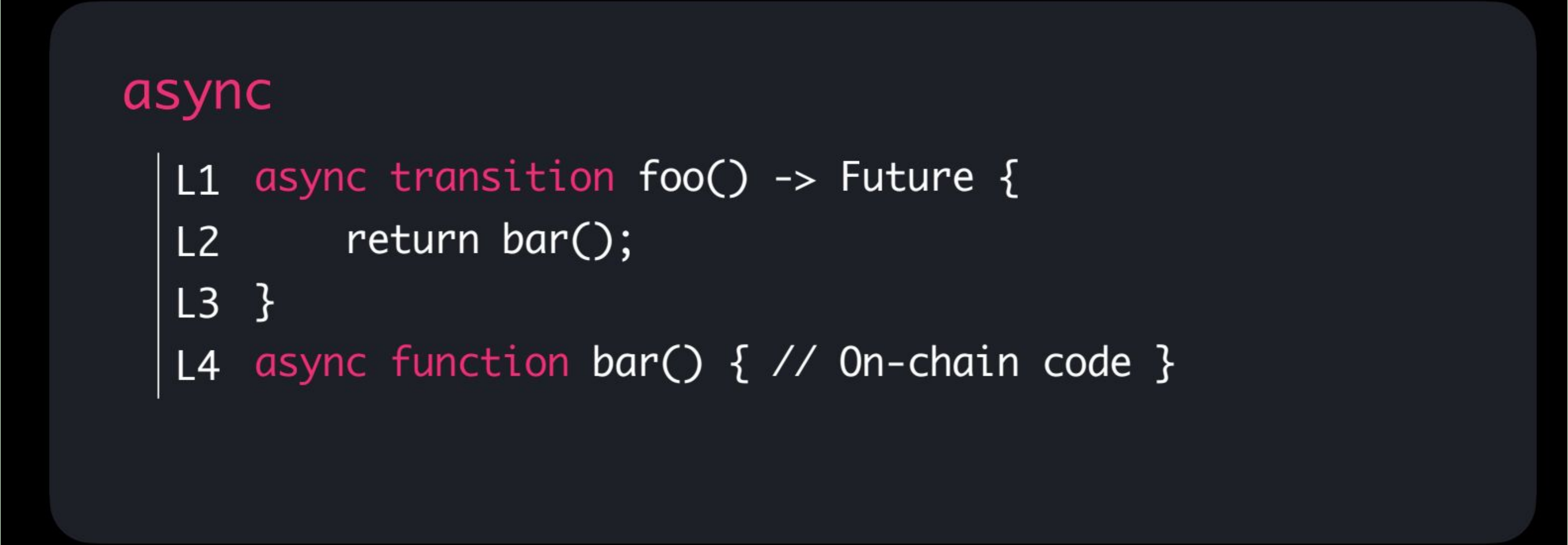
- Modifying Onchain State
  - async transition
    - Offchain computation with ZK proof of execution
    - async keyword signals additional onchain computation to follow
      - Otherwise acts same as regular transition
    - Must return at least a Future
      - Call to an async function

Tokens on Aleo

- Modifying Onchain State
    - async function
        - Onchain computation
        - All inputs are public
        - Can only be called by async transition, not standalone

Tokens on Aleo

- Modifying Onchain State



async

```
async transition foo() -> Future {
    return bar();
}
async function bar() { // On-chain code }
```

Compliance on Aleo

- Maintain updated list of OFAC sanctioned addresses in a mapping onchain
- Add assertions in every function that querying this mapping with transaction sender/recipients
  - Prevent any transfers or swaps of tokens from or to sanctioned addresses

Questions?

Let's code!

Your Mission:

1. Build a token program in Leo using the provided template code. Your program must include:
    1. mint_public & mint_private functions
    2. transfer_public & transfer_private functions
    3. Compliance checks against workshop_ofac.aleo for all of the above
2. Deploy your program to Testnet.
3. Interact with your deployed program onchain:
    1. Publicly mint 100 tokens to your address
    2. Publicly transfer those tokens to <WORKSHOP_ADDRESS>
    3. Privately mint an additional 100 tokens to your address
    4. Privately transfer those tokens to <WORKSHOP_ADDRESS>

Thank You!