Getting Started with Arbitrum Stylus

Ben Greenberg
@hummusonrails

ARBITRUM FOUNDATION

Hey, there. I'm Ben.

Senior DevRel Engineer

ARBITRUM

Online Everywhere at

HUMMUS ON RAILS

ARBITRUM FOUNDATION

Workshop Objectives

01

Understand how Stylus extends Solidity with the MultiVM

02

Learn how to deploy Solidity contracts alongside Stylus contracts

03

Explore Solidity<>Stylus interoperability in real dApps

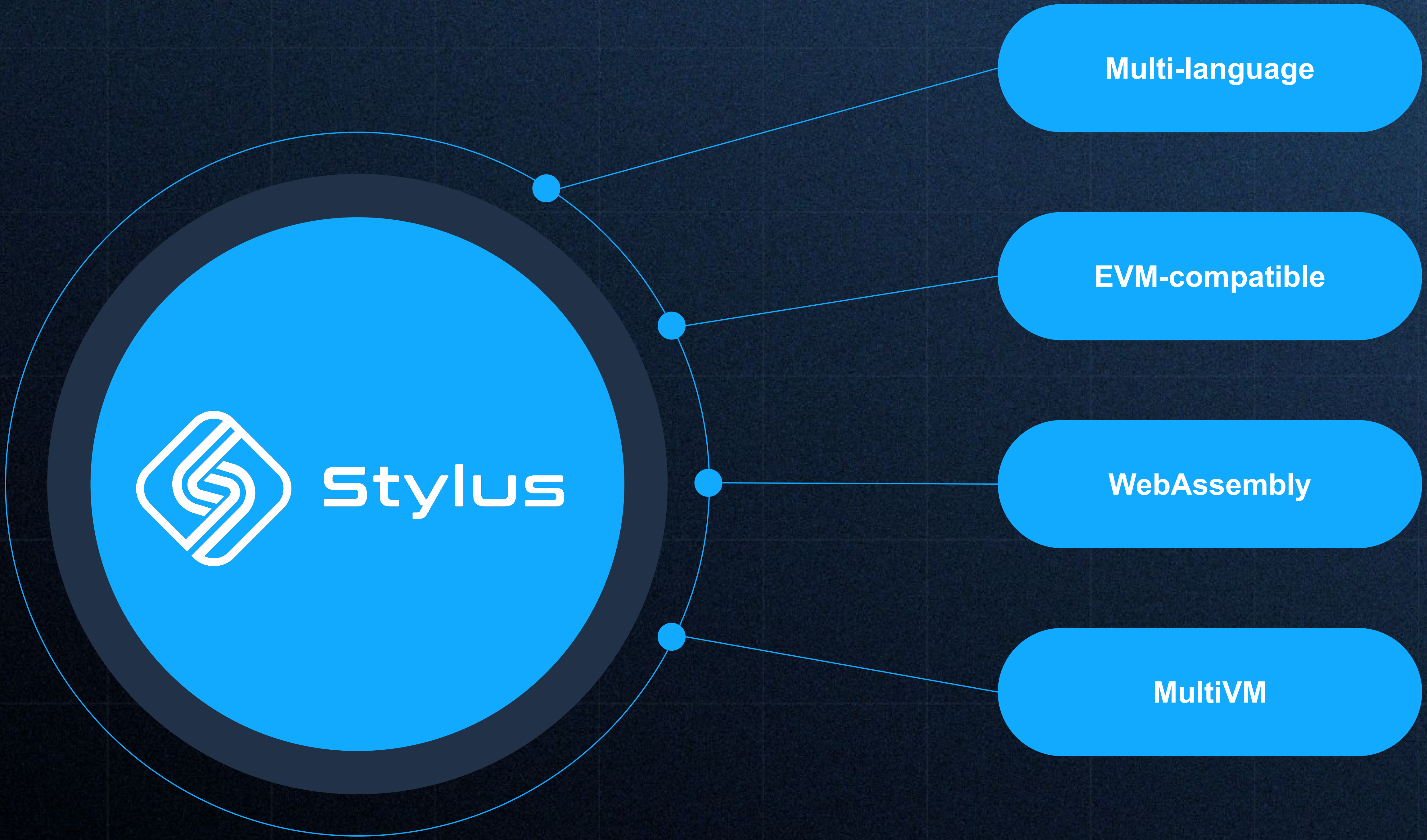
You're already a Web3 dev.

You bring Solidity.
Stylus brings cheaper memory and compute.

ARBITRUM
FOUNDATION

Stylus **Fundamentals**

ARBITRUM
FOUNDATION



A new way to write smart contracts in Rust, C/C++, and any Wasm-friendly language

Runs alongside the EVM on Arbitrum, not a replacement

Leverages WebAssembly for faster, safer, and more flexible execution

Stylus runs alongside the EVM with shared state and storage. Arbitrum's co-equal VM design enables both to access host I/O interchangeably.

Why Stylus Exists

EVM is limited to 256-bit, high-cost ops leading to higher gas costs and inefficiencies

Solidity stays. Stylus brings scalable Wasm.

Stylus gives you CPU & memory-efficient Wasm execution, rich type safety and Rust-based tooling

Key Concepts in Stylus

sol_storage!	Mimics Solidity layout but written in Rust
#[entrypoint]	Defines the root struct of the contract
#[public]	Marks methods that are externally callable
Host I/O	Enables calls between Stylus and Solidity contracts
MultiVM	Runs Wasm and EVM in parallel, shared state tree

ARBITRUM FOUNDATION

Solidity + Stylus: Better Together

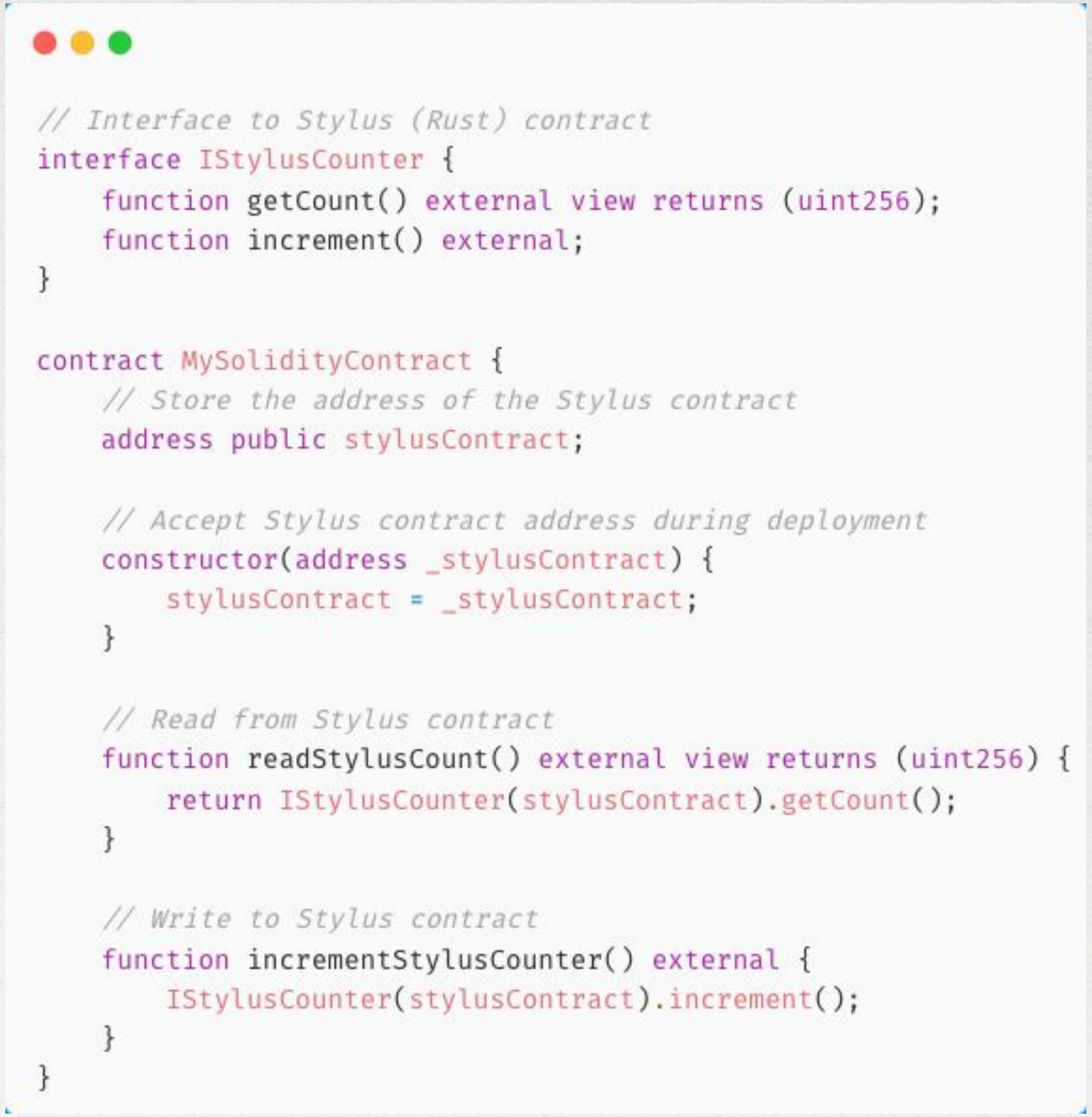
Stylus +

✅ Interop between Wasm and EVM on-chain

✅ Shared storage model and state tree

✅ Call Stylus contracts from Solidity and vice versa

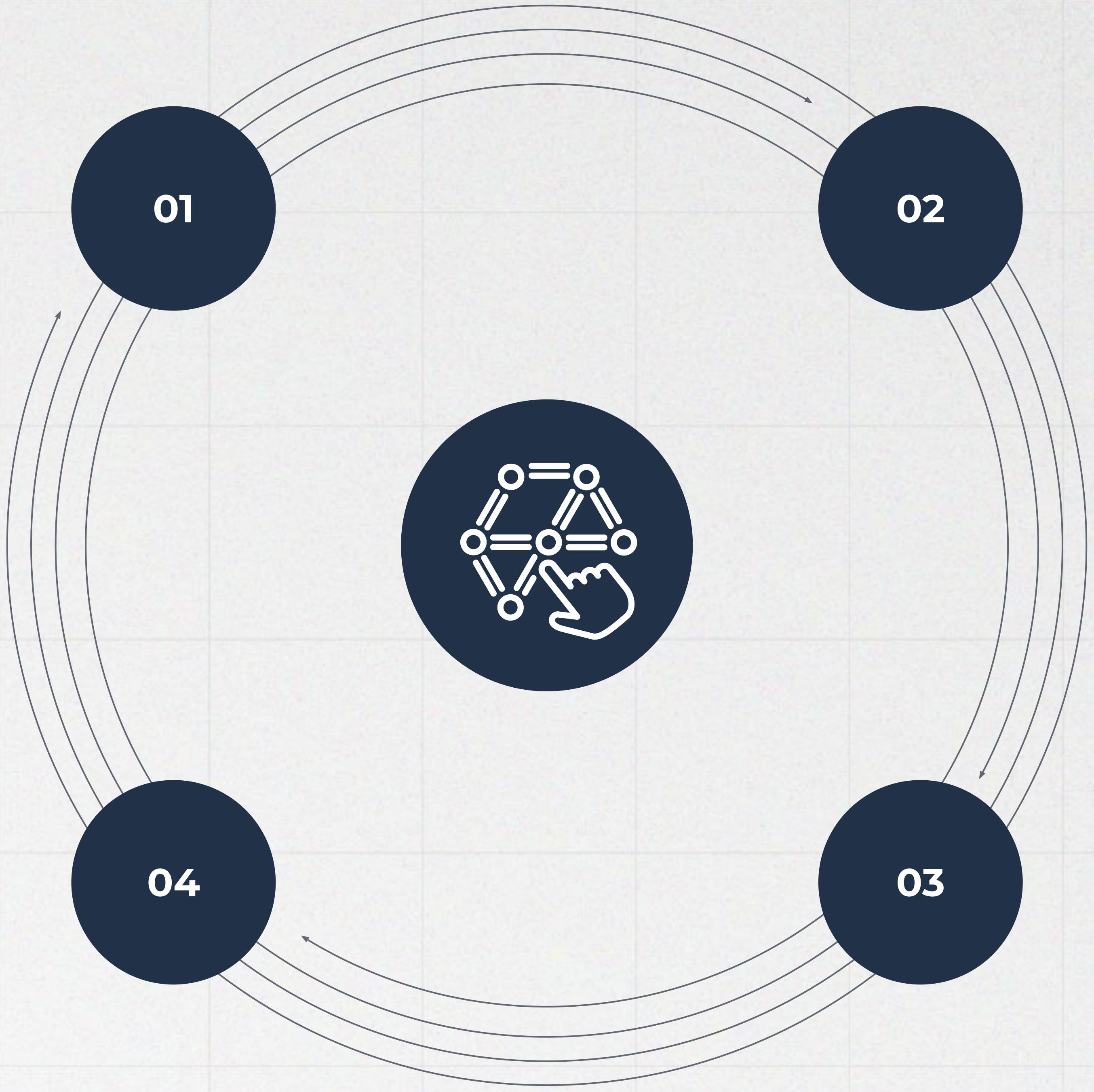
✅ Build hybrid dApps with minimal overhead



```
// Interface to Stylus (Rust) contract
interface IStylusCounter {
    function getCount() external view returns (uint256);
    function increment() external;
}

contract MySolidityContract {
    // Store the address of the Stylus contract
    address public stylusContract;

    // Accept Stylus contract address during deployment
    constructor(address _stylusContract) {
        stylusContract = _stylusContract;
    }

    // Read from Stylus contract
    function readStylusCount() external view returns (uint256) {
        return IStylusCounter(stylusContract).getCount();
    }

    // Write to Stylus contract
    function incrementStylusCounter() external {
        IStylusCounter(stylusContract).increment();
    }
}
```

ARBITRUM FOUNDATION

Unified Deployment & Execution



Interop: Stylus <> Solidity
Stylus enables Wasm execution while keeping compatibility with your Solidity contracts. Contracts can call each other across VMs.

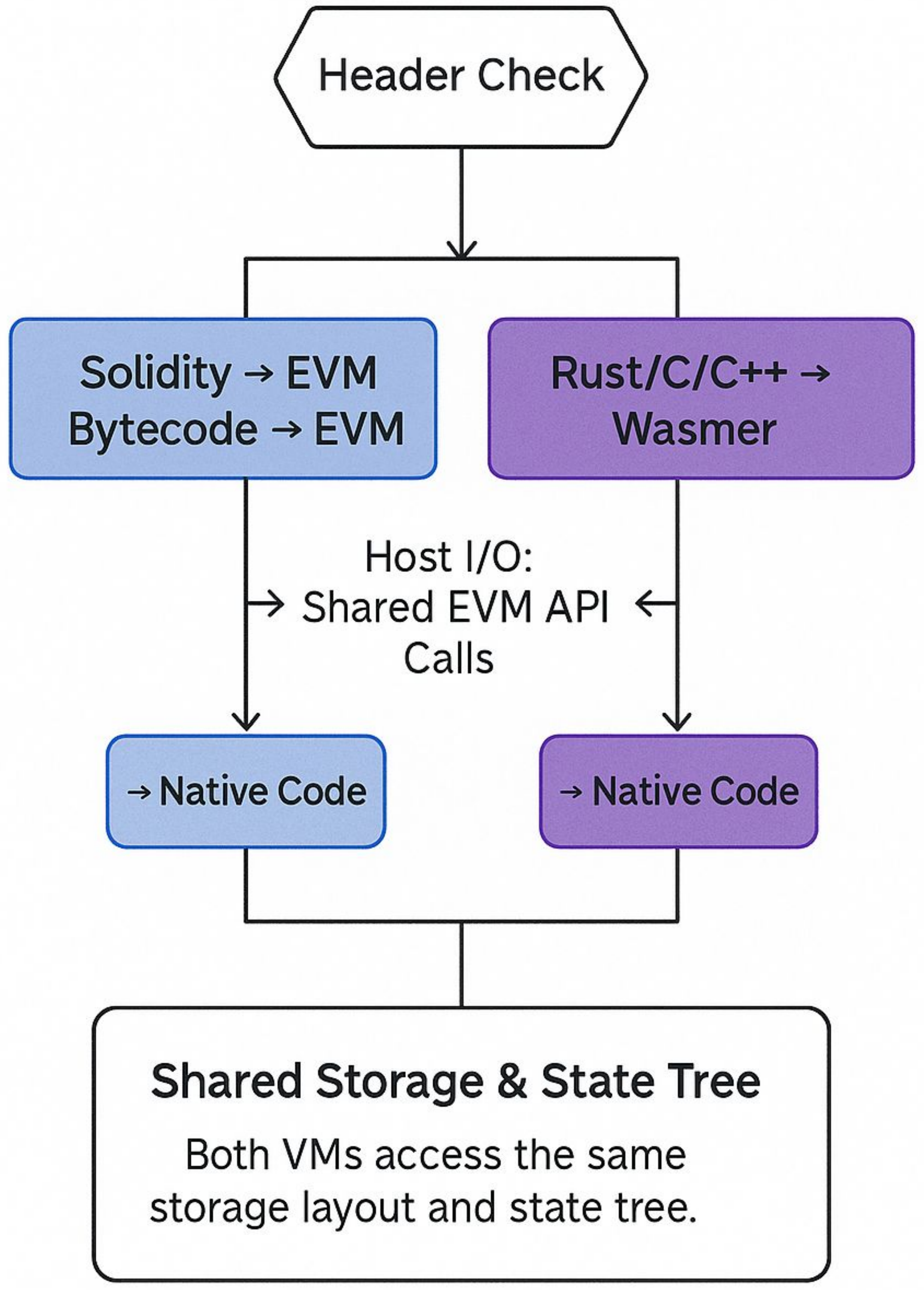
Performance Gains
Deploy Solidity and Stylus contracts to the same Arbitrum chain. Offload expensive logic to Stylus for massive gas savings.

Shared Tooling and Infra
Use existing EVM infra (e.g., Foundry, Hardhat) alongside cargo stylus. ArbWasm handles Wasm contract activation under the hood.

Flexible Language Support
Write high-performance modules in Rust, C, or C++ to extend your Solidity dApp — no rewrite needed.

Stylus Execution Paths

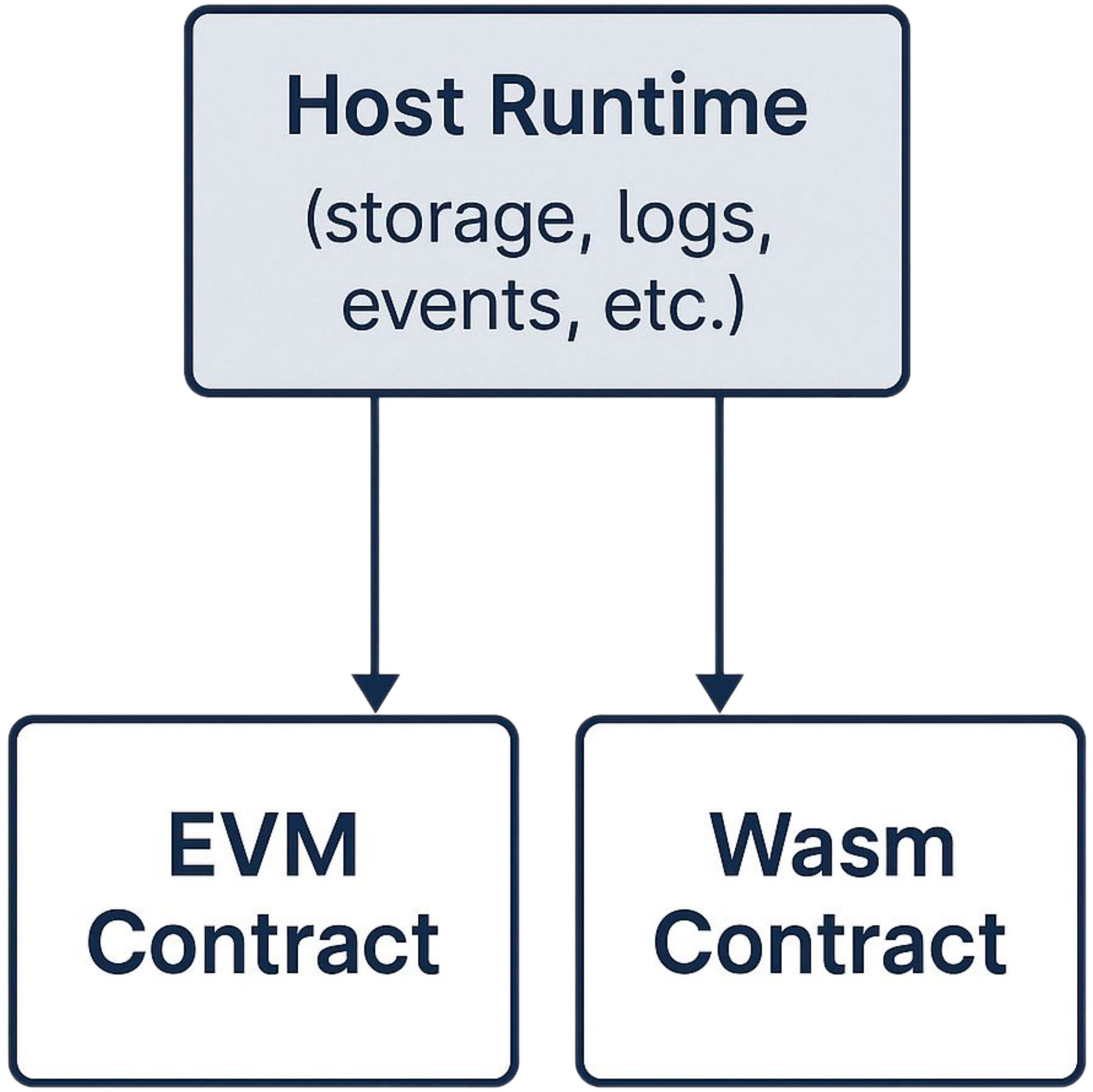
- MultiVM Architecture
  - Both EVM and Wasm execution
  - Select VM based on header
- Interoperable Calls
- Unified Store Access



ARBITRUM FOUNDATION

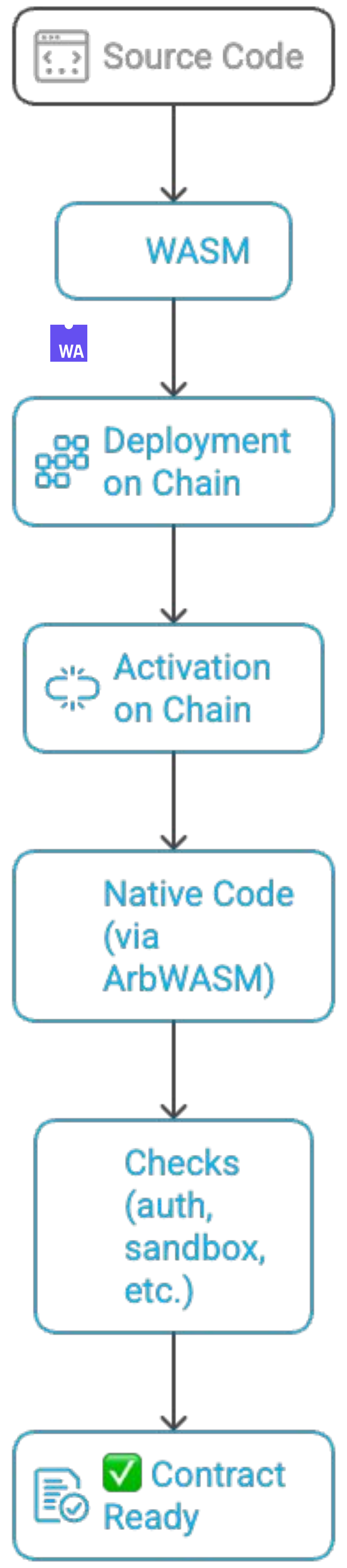
Shared Access Between VMs

- Co-equal VMs share the same host interface
- Shared ABI and host environment
- Consistent behavior across VMs



Stylus Deployment Path

- Written in any language that compiles to Wasm
- Compile to Wasm
- Deploy On-Chain
- Activate via ArbWasm



ARBITRUM FOUNDATION

What is WebAssembly?

- **Portable**: Runs the same across devices and operating systems
- **Fast**: Near-native execution speeds
- **Safe**: Sandboxed, memory-safe execution
- **Language-agnostic**: Compile from many languages
- **Well-supported**: Large and growing developer ecosystem

Why Wasm is Ideal

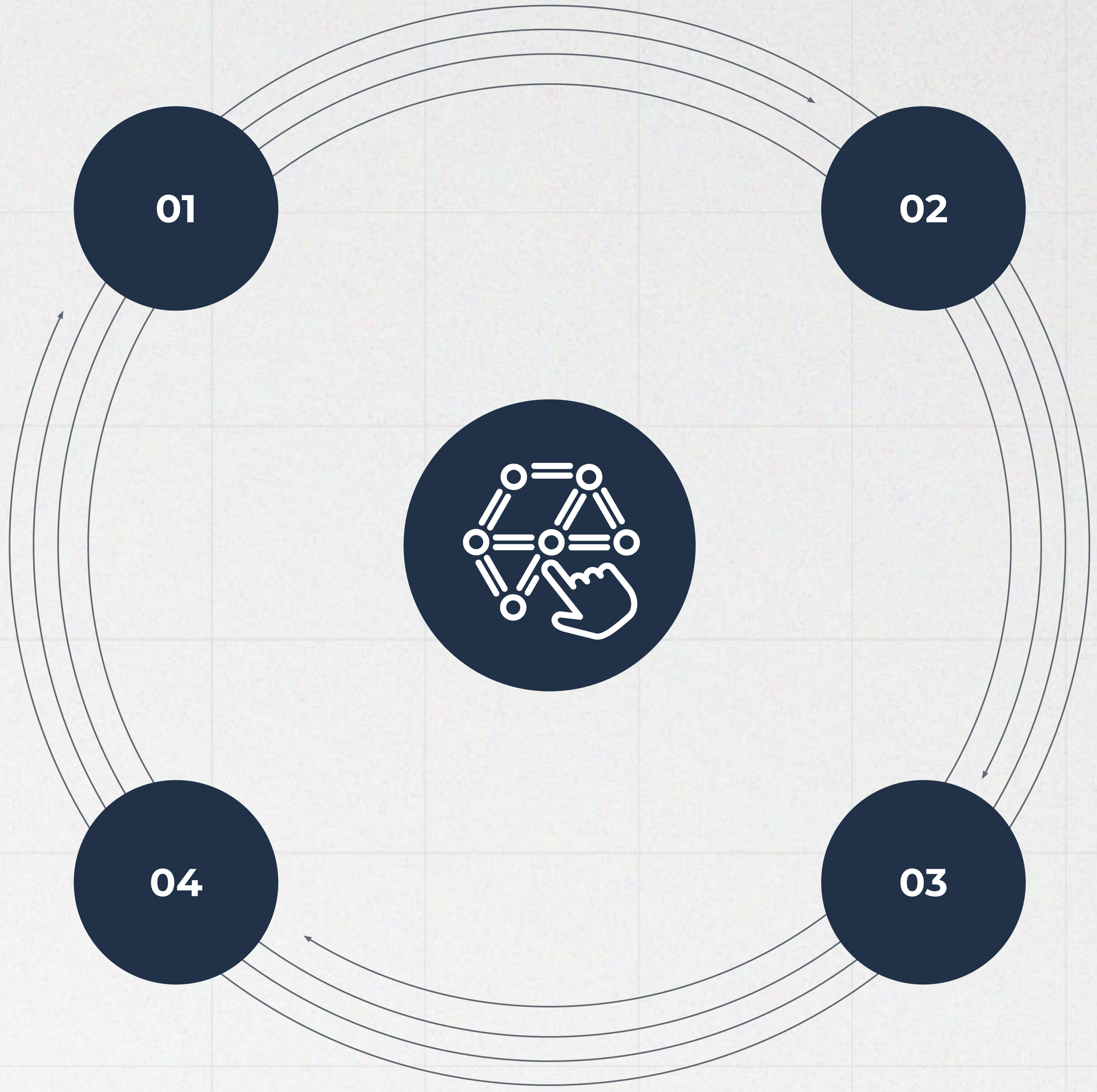
- Optimized for short, deterministic operations
- Reduces gas costs for compute-heavy logic
- Enables safer execution with stricter compile-time checks
- Supports modern toolchains (like Rust and cargo)

What's Being Built with

Stylus x

ARBITRUM
FOUNDATION

Stylus Highlights in Production



Lit Protocol

Onchain access control: Keys, permissions, and identity

Fairblock

Onchain decryption: Unlock data only when rules are met

CVEX

Trading platform using Stylus for faster and cheaper transactions

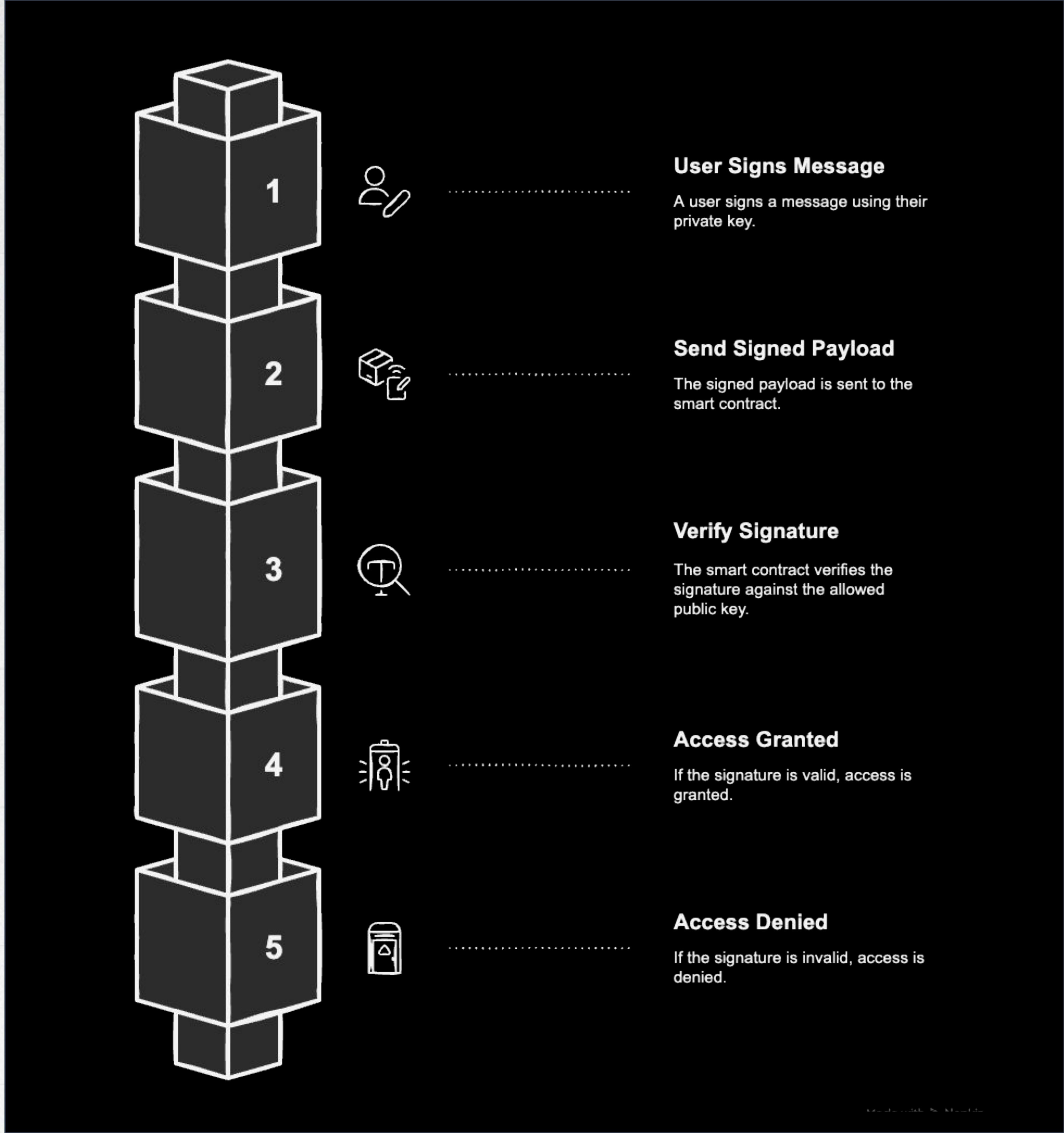
Superposition

A chain that pays users and devs for onchain activity

Stylus x Lit

Access Control with Precompiles

- Key generation and derivation
- Signature checks using precompiled cryptography
- Conditional access logic written in Rust
- Controlled decryption only when conditions are met
- Secure verification of signed requests



User Signs Message

A user signs a message using their private key.

Send Signed Payload

The signed payload is sent to the smart contract.

Verify Signature

The smart contract verifies the signature against the allowed public key.

Access Granted

If the signature is valid, access is granted.

Access Denied

If the signature is invalid, access is denied.

ARBITRUM FOUNDATION

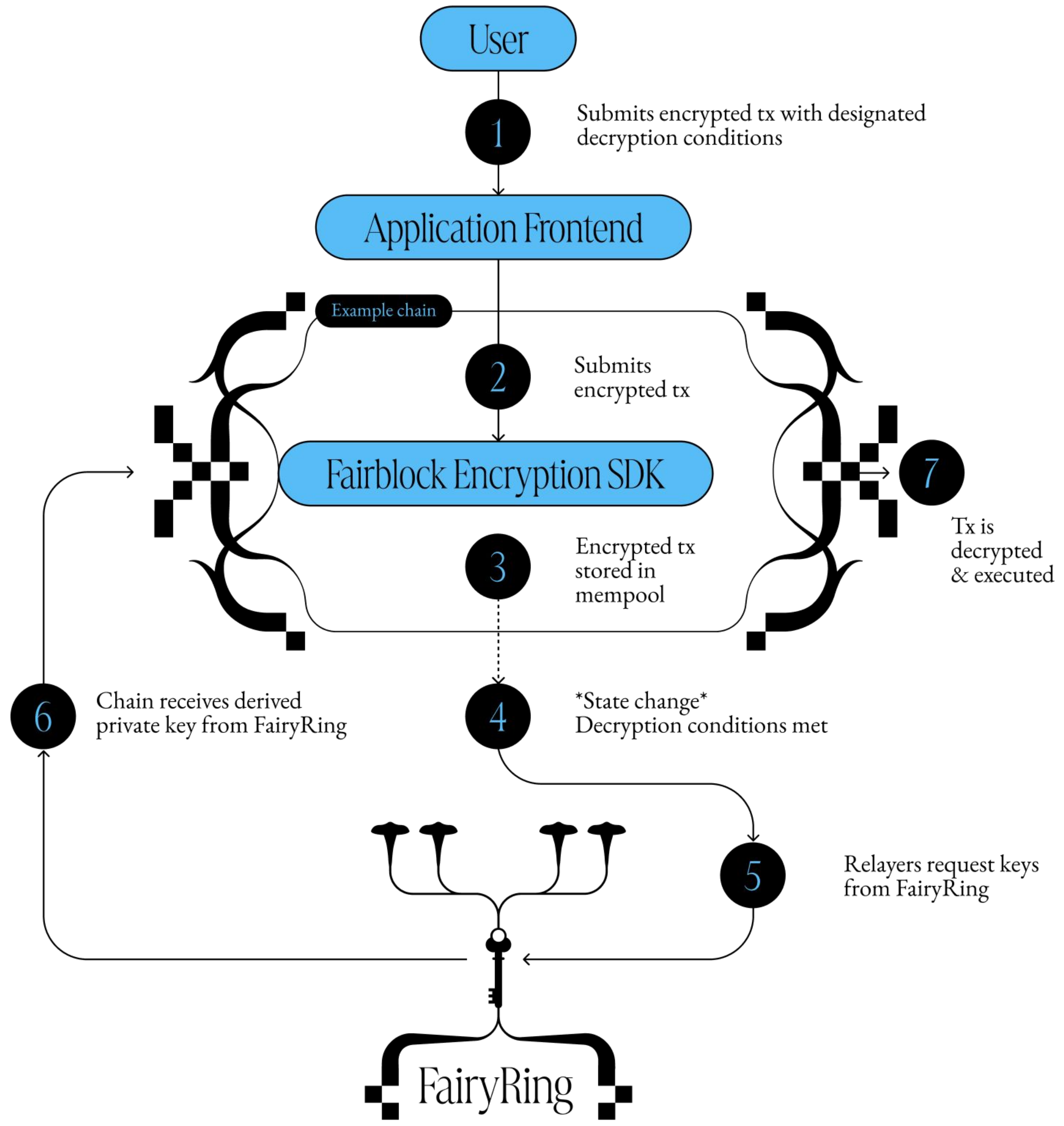
Stylus x Fairblock

Policy-Based Decryption

- Onchain keygen, derivation, and cryptographic logic written in Rust
- Efficient signature checks using Stylus for high-perf Wasm execution
- Conditional access and decryption policies encoded in standard Rust
- Real-time decryption now viable within a block's gas limits
- Enabling private AI, transactions, etc.

ARBITRUM FOUNDATION

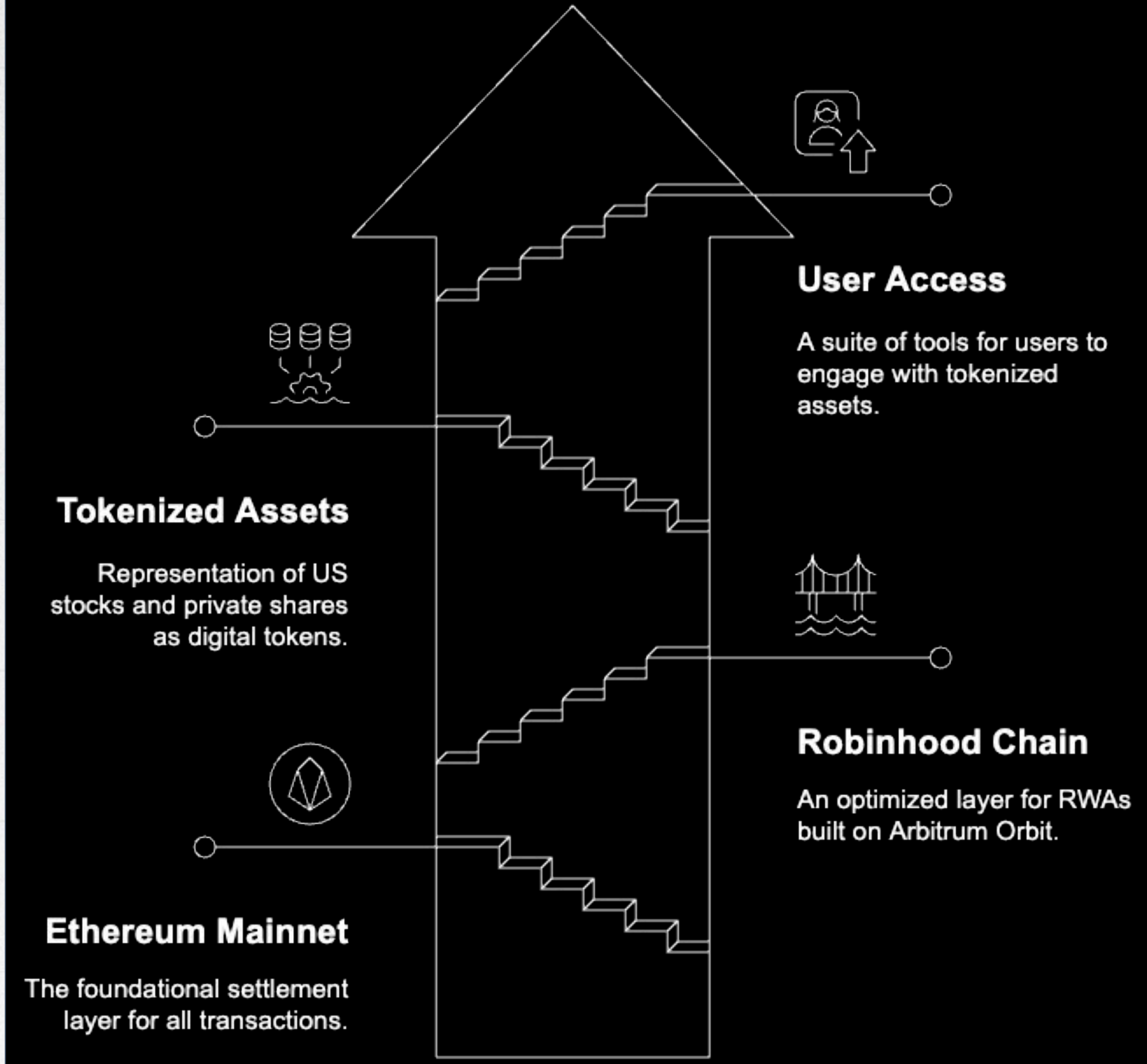
Example User Flow


User
1
Submits encrypted tx with designated decryption conditions
Application Frontend
Example chain
2
Submits encrypted tx
Fairblock Encryption SDK
3
Encrypted tx stored in mempool
4
State change Decryption conditions met
5
Relayers request keys from FairyRing
6
Chain receives derived private key from FairyRing
7
Tx is decrypted & executed
FairyRing

Robinhood

Real-World Assets Onchain

- Launched US stock and ETF tokens for EU users
- Tokens issued on Arbitrum, bringing real-world assets onchain
- Announced custom Layer 2 blockchain based on Arbitrum, optimized for RWA trading
- New chain will support 24/7 trading, seamless bridging, and self-custody

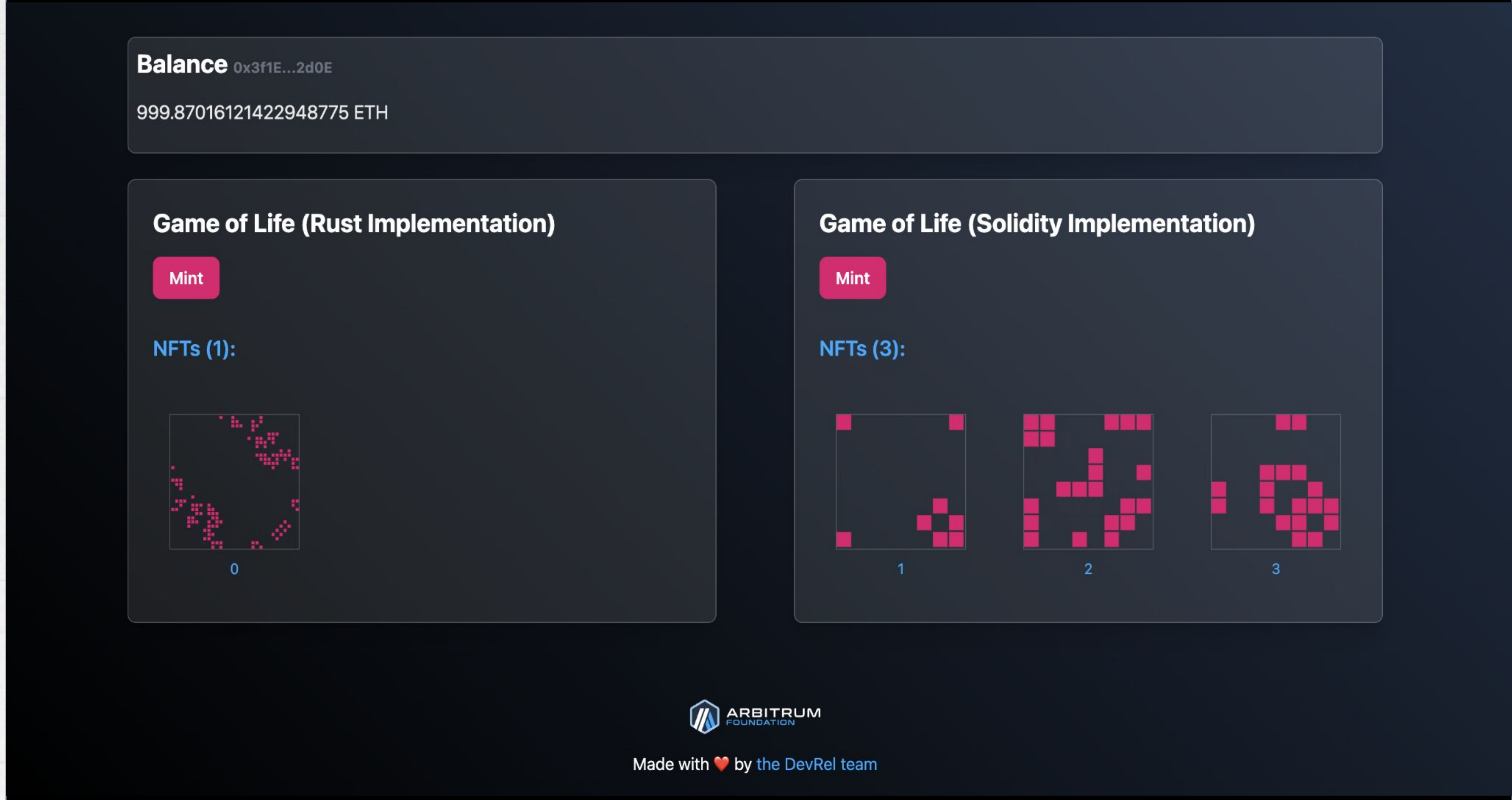

User Access
A suite of tools for users to engage with tokenized assets.
Tokenized Assets
Representation of US stocks and private shares as digital tokens.
Robinhood Chain
An optimized layer for RWAs built on Arbitrum Orbit.
Ethereum Mainnet
The foundational settlement layer for all transactions.

What We’re **Building**

ARBITRUM
FOUNDATION

Game of Life

- Conway's Game of Life is a cellular automaton with simple rules and evolving complexity
- In this workshop, the **smart contract simulates the game's logic**
- You'll implement:
  - Grid setup
  - Step mechanics
  - Read/write state logic
- The Solidity contract becomes the game engine



Balance 0x3f1E...2d0E

999.87016121422948775 ETH

Game of Life (Rust Implementation)

NFTs (1):

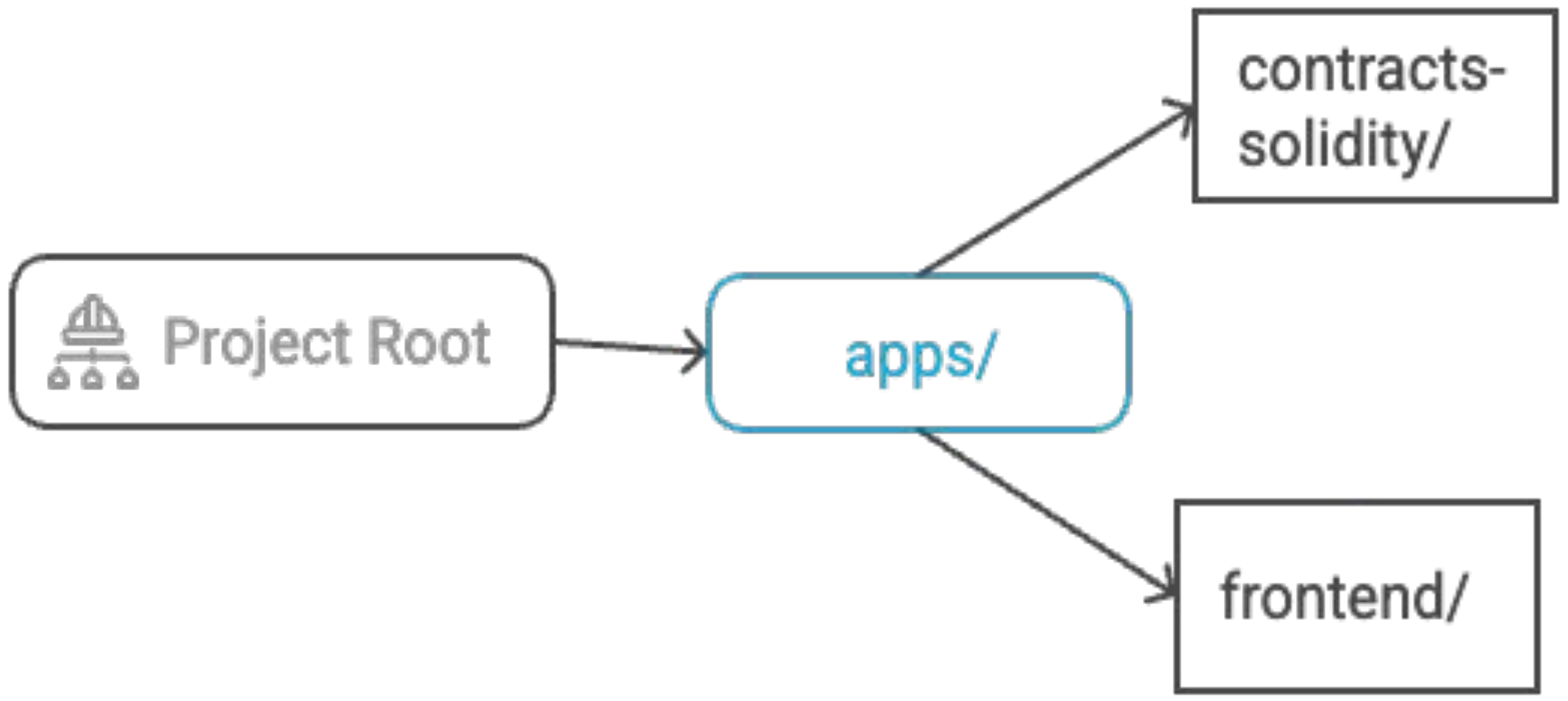
Game of Life (Solidity Implementation)

NFTs (3):

Made with ❤️ by the DevRel team

Project Structure

- **contracts-solidity/**: Your Solidity smart contract lives here. You'll finish the logic for the Game of Life.
- **frontend/**: A prebuilt UI to visualize and interact with the contract. You'll connect it using RPC.



ARBITRUM FOUNDATION

Workshop Flow Overview

Step	Action	Details
1	Setup the Workspace	Open the repo in GitHub Codespaces
2	Explore the Project	Review the contract and frontend existing code
3	Build the Solidity Contract	Implement the TBC sections of the contract logic in contracts-solidity/src/NFT.sol
4	Connect to the Frontend	Hook up the React app to call the smart contract functions
5	Interact	Launch Anvil, deploy contract locally, use UI to interact with the contract

ARBITRUM FOUNDATION

Are You

Ready?

ARBITRUM
FOUNDATION

Launch the Codespace

https://qr.me-qr.com/oxUGndDq

--help

- https://docs.arbitrum.io/stylus/stylus-overview
- https://github.com/OffchainLabs/awesome-stylus
- https://github.com/ArbitrumFoundation/stylus-workshop-gol

Want to Stay Informed?

ARBITRUM
FOUNDATION

@hummusonrails

The Builder's Block Newsletter

ARBITRUM FOUNDATION

Thank you!

HUMMUS ON RAILS

ARBITRUM
FOUNDATION