Solana

Zichao Yang

Zhongnan University of Economics & Law

Date: December 9, 2025

Blockchain Trilemma

Blockchain Trilemma: the theoretical constraint positing that a decentralized network can simultaneously maximize only two of three properties: decentralization, security, and scalability.

- First-generation protocols, exemplified by Bitcoin, prioritized decentralized security via PoW, sacrificing throughput.
- Second-generation platforms like Ethereum introduced smart contracts, improve performance through secondary layer solutions, such as sidechains, layer-2 solutions, etc.
- Solana represents a third-generation solution designed for scalability and efficiency building on a monolithic, high-performance architecture.

What is Solana?

Solana was founded in 2018 by Anatoly Yakovenko and Raj Gokal, and the network was launched in March 2020 by their San Francisco-based company, Solana Labs.

- **High-Performance L1**: It is a Layer 1 blockchain designed for mass adoption, characterized by high throughput (65k+ theoretical TPS) and sub-second finality (400ms block times).
- **Unique Architecture**: Unlike Ethereum, it utilizes a hybrid consensus model combining **Proof of Stake (PoS)** with **Proof of History (PoH)** to synchronize time across the network without heavy communication overhead.
- **Native Token**: The network's utility token is **SOL**, which is used for transaction fees, staking to secure the network, and governance.

Solana’s Popularity Waves

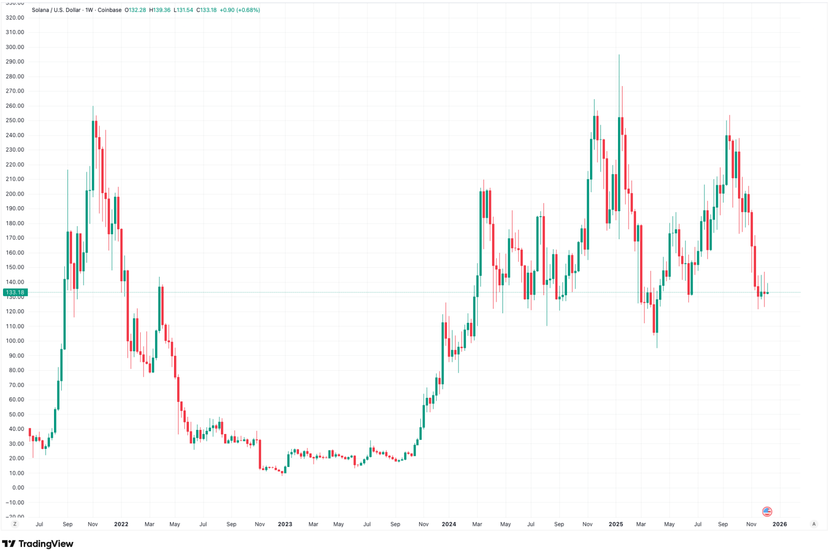


Figure 1: Solana Price Hisotry

Solana's Popularity Waves

- **Rising Star (2021)**: The “Ethereum Killer” Narrative
  - During the 2021 bull market, using Ethereum cost \$50–\$100 per tx. Solana cost \$0.00025 per tx.
  - It was heavily promoted by Sam Bankman-Fried (FTX), which gave it massive visibility (before the 2022 FTX collapse hurt Solana's reputation).
- **Comeback (2024-2025)**: The “Meme Coin Supercycle”
  - Meme coins surged in popularity.
  - Because fees are so low, Solana is the perfect place to trade high-risk, low-value assets like Meme Coins.

Solana Consensus: PoH + PoS

Proof of History (PoH)

- **Role**: Acts as a ”Global Clock” before consensus.
- **Mechanism**: A recursive Verifiable Delay Function (VDF) (SHA-256 loops).
- **Benefit**: Validators do not need to wait and talk to each other to agree on the *time* or order of events.
- **Result**: Massive throughput ($\approx 4,000+$ TPS).

Solana Consensus: PoH + PoS

Proof of Stake (PoS)

- **Leader Schedule**: Solana does not use “Coin Age” (time held) for selection, instead, it uses the leader schedule to determine the leaders for the next epoch. The schedule is known in advance for the whole epoch.
- **Stake-Weighted**: The probability of being assigned a leader slot is proportional to the amount of SOL staked.
- **Voting**: Validators vote on the validity of the PoH hash sequence (Tower BFT).

How the Leader Schedule Works?

- **Epochs**: Solana uses epochs (about 432,000 slots, roughly 2 days) to manage stake and scheduling.
- **Leader Election**: At the start of an epoch, a new schedule is generated using a pseudo-random algorithm seeded by the Proof-of-History (PoH) tick height.
- **Stake-Weighted Selection**: Validators are chosen as leaders for a set of slots (often 4 at a time) in proportion to their staked amount, ensuring fair distribution.
- **Predictable Block Production**: The resulting schedule is a public list mapping validator keys to slot numbers, meaning the entire epoch's block production is known ahead of time.

Account in Solana

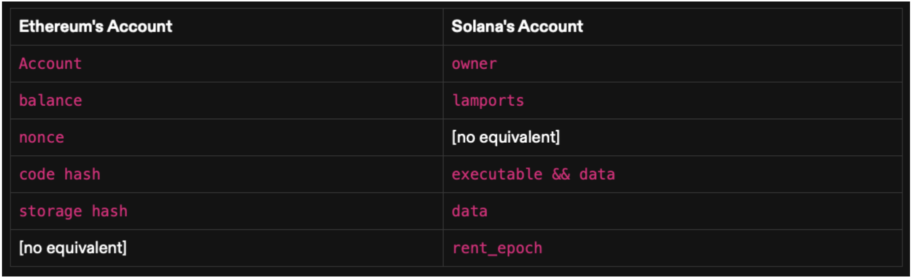
- **Philosophy**: "Everything is an Account." In Solana, all data is stored and executed based on Accounts.
- **Types of Accounts**:
  - **Executable Account (Program Account)**: These are smart contracts that store code, often referred to more simply as “programs”.
  - **Non-Executable (Data)**: These can receive tokens or data but cannot execute code, as the executable variable is set to “false”.

Account in Solana

- **Key Fields in Account**:
  - `lamports`: Balance (like Wei in Eth).
  - `owner`: Program that owns this account.
  - `executable`: Whether the account is executable.
  - `data`: Raw byte array.
  - `rent_epoch`: The epoch when the account was last rented (deprecated).
- **Rent (Storage Collateral)**:
  - Accounts **must** be "Rent Exempt" (hold 2 years of rent in lamports).
  - This acts as a **refundable deposit**.
  - When you close an account, you withdraw the rent back to your wallet.

Account Structures in Ethereum and Solana

A comparison of the account structures in Ethereum and Solana chains reveals the following differences.



Ethereum's Account	Solana's Account
`Account`	`owner`
`balance`	`lamports`
`nonce`	[no equivalent]
`code hash`	`executable && data`
`storage hash`	`data`
[no equivalent]	`rent_epoch`

Figure 2: Account Structures in Ethereum and Solana

Source: Solana Website - EVM vs. SVM: Accounts

Ethereum: Smart Contracts

- **Language**: Solidity, Vyper.
- **State**: **Stateful**. State variables are stored permanently inside the contract's own address.
- **Data Storage**:
  - Uses a **Storage Trie** (key-value mapping).
  - Very expensive (High Gas) to write data.
- **Deployment**: Code is sent to the 0x0 address; creates a new address containing the bytecode.
- **Execution**: **Sequential** (Single-threaded EVM).

Solana: Programs

- **Language**: Rust, C++.
- **State**: **Stateless**. The program logic is “Read-Only”.
- **Data Storage**:
  - Passed in via **External Accounts**.
  - **PDAs** (Program Derived Addresses) map data to specific users/programs.
- **Deployment**: Compiled to **SBF** (Solana Bytecode Format) and stored in a specialized ”Executable Account.”
- **Execution**: **Parallel** (Sealevel Runtime).

Ethereum: Stateful (Coupled)

- **Definition**: Code and Data reside at the *same* address.
- **Mechanism**: A Smart Contract contains persistent storage variables (e.g., `balances`).
- **Execution**: To update state, you send a transaction to the contract.
- **Constraint**: Since the contract holds the global state, transactions interacting with it generally execute **sequentially** to avoid conflicts.

Solana: Stateless (Decoupled)

- **Definition**: Code (Program) and Data (Account) reside at *different* addresses.
- **Mechanism**: Programs are “Read-Only” executable files. They receive data accounts as inputs to modify.
- **Execution**: To update state, you pass the *specific data account* to the program.
- **Advantage**: **Parallelism**. Multiple users can use the same Program to update different Data Accounts simultaneously without blocking each other.

Analogy: *Ethereum is like a **Vending Machine** (One person uses the machine at a time). Solana is like a **Word Processor** (Everyone edits their own files using the same software at the same time).*

Node Architecture

Ethereum Clients

- **Execution Client**: Executes transactions (EVM), maintains state (e.g., Geth).
- **Consensus Client**: Handles PoS agreement (e.g., Prysm).
- **Node Types**: Full Node, Archive Node (all history), Light Node.

Solana Nodes

- **Consensus Node (Validator)**: Participates in voting and block production. High hardware requirements.
- **RPC Node**: Gateway for dApps; does not vote.
- **History Storage**: Due to high TPS, storing full history on every node is impractical. Older history often offloaded to Bigtable.

Summary Comparison

Feature	**Ethereum**	**Solana**
Throughput	$\sim$30 TPS	4,000+ TPS
Consensus	Gasper (PoS)	Tower BFT (PoS + PoH)
Logic/Data	Coupled (Stateful)	Decoupled (Stateless)
Languages	Solidity, Vyper	Rust, C++
Account Cost	Gas for storage	Rent (Lamports)