L2 EVM Common Core

A Path Beyond (L1) EVM Equivalence

ansgar.eth
Research @ Ethereum Foundation

RollCall & RIPs

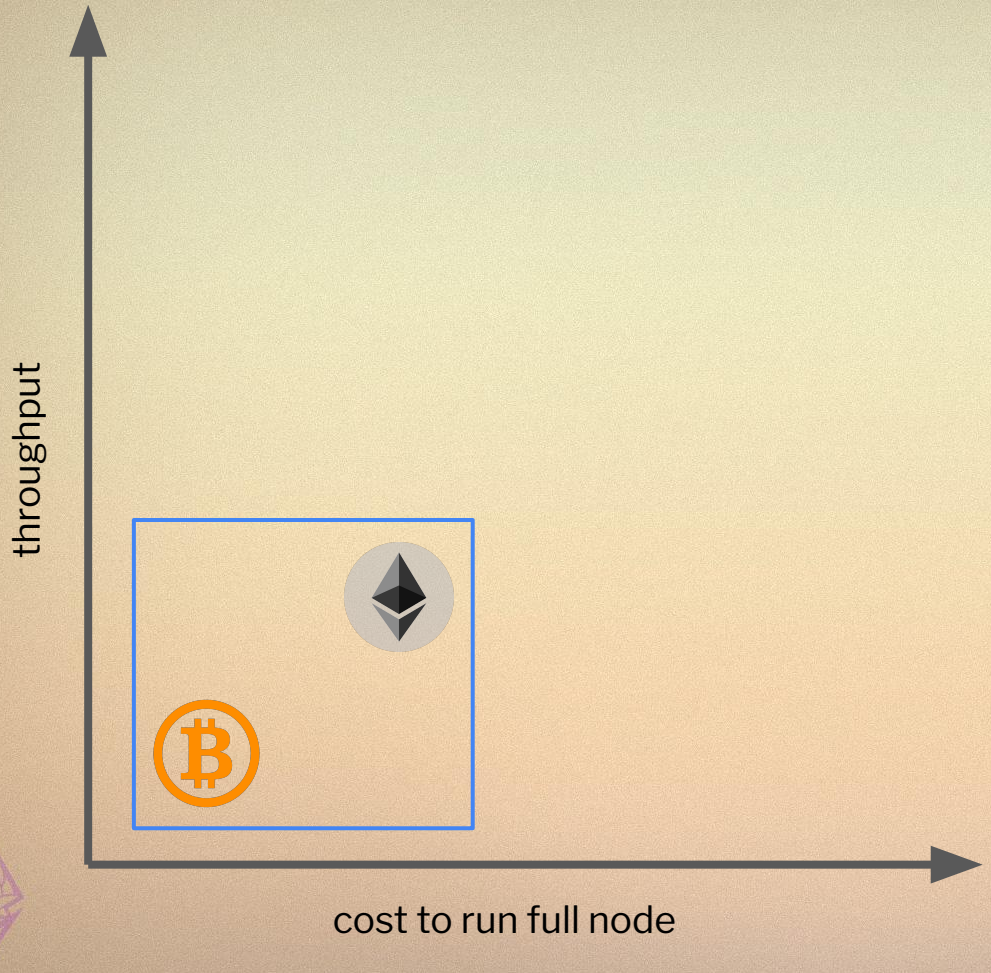
RollCall

- monthly L2 coordination call
- L1 <> L2 connective tissue

RIPs

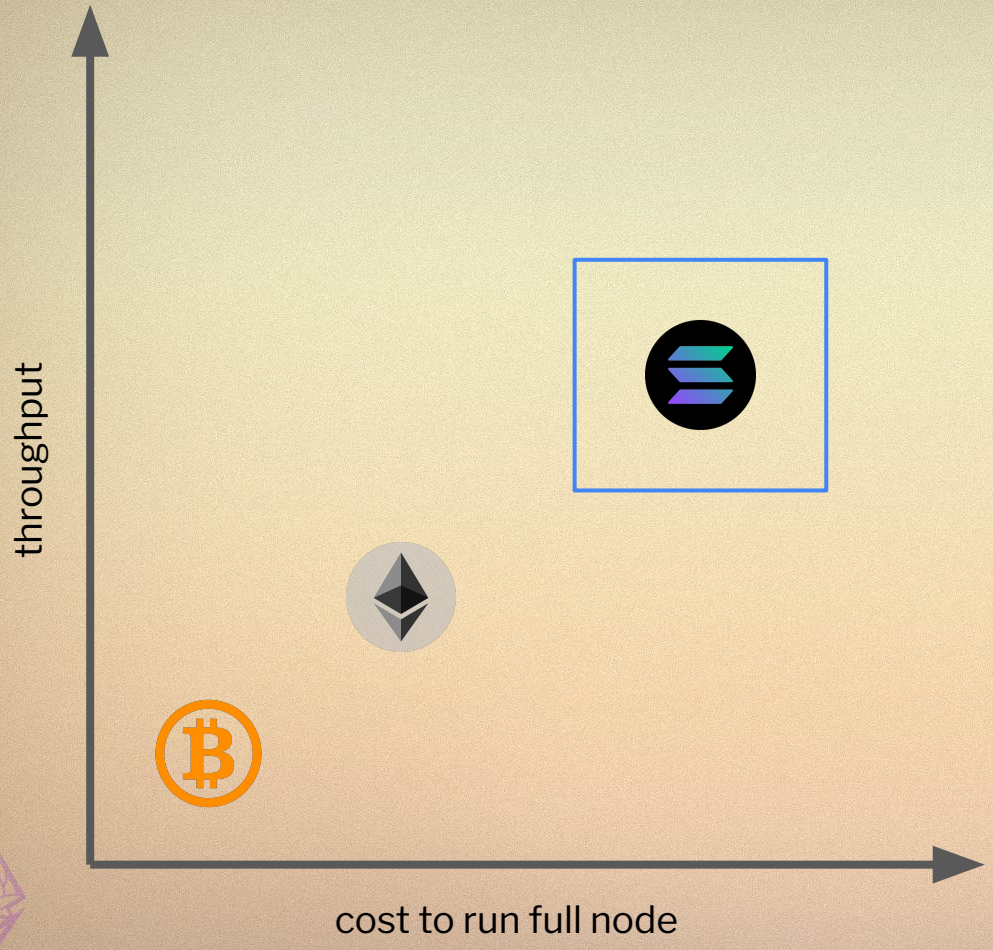
- **R**ollup **I**mprovement **P**roposals
- opt-in EVM features for L2s
- https://github.com/ethereum/RIPs

1. History



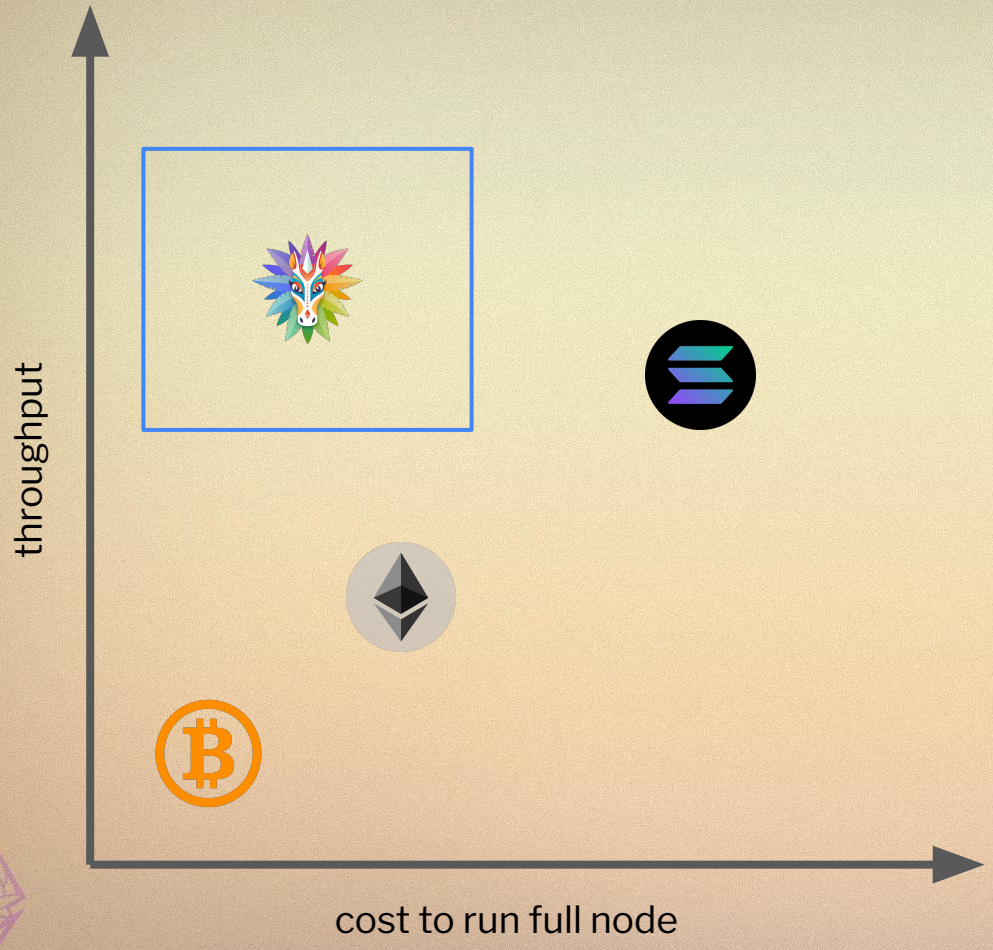
Trustless Chains

- everyone (who wants to) can run the chain locally
- chain throughput limited by consumer hardware capabilities
- “everyone validating everyone’s transactions” doesn’t scale!



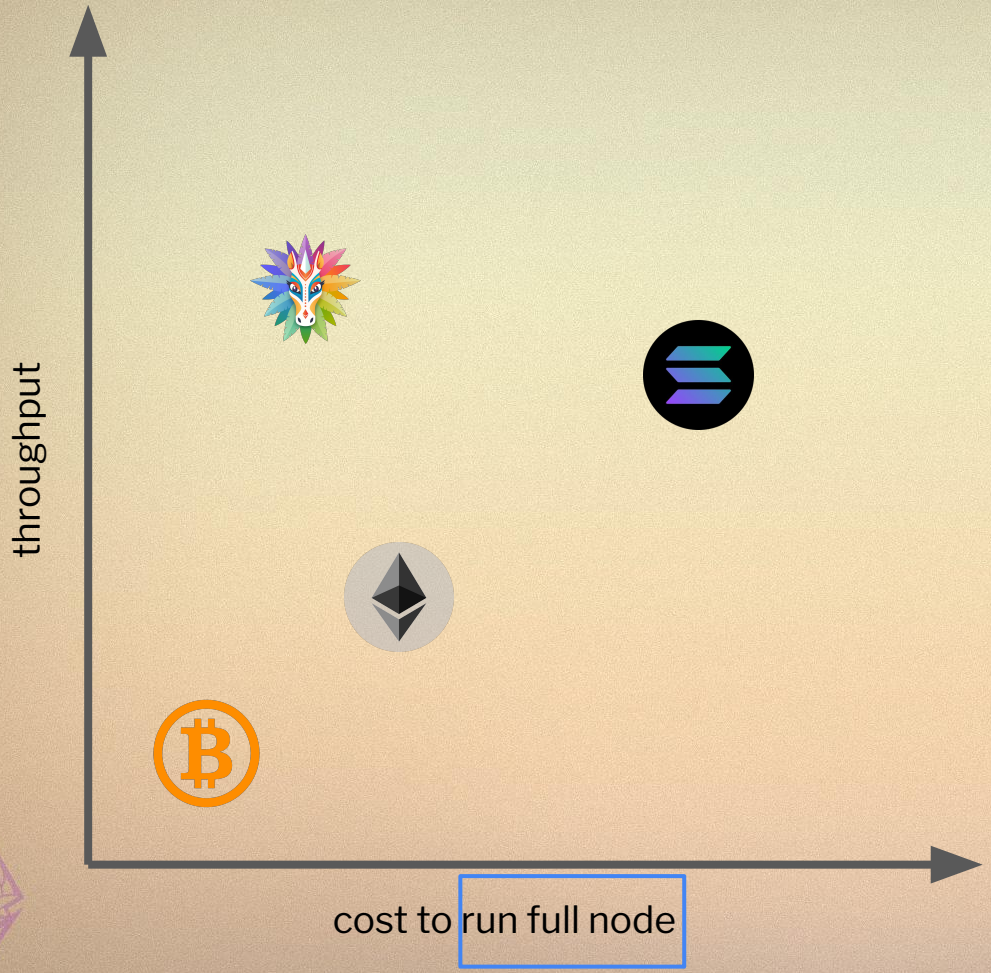
High Throughput Chains

- everyone (who has the money) can run the chain on expensive servers in datacenters
- high throughput
- on disagreement (without local node):
  - go with majority (51% can rewrite rules) or
  - halt and recover via social layer



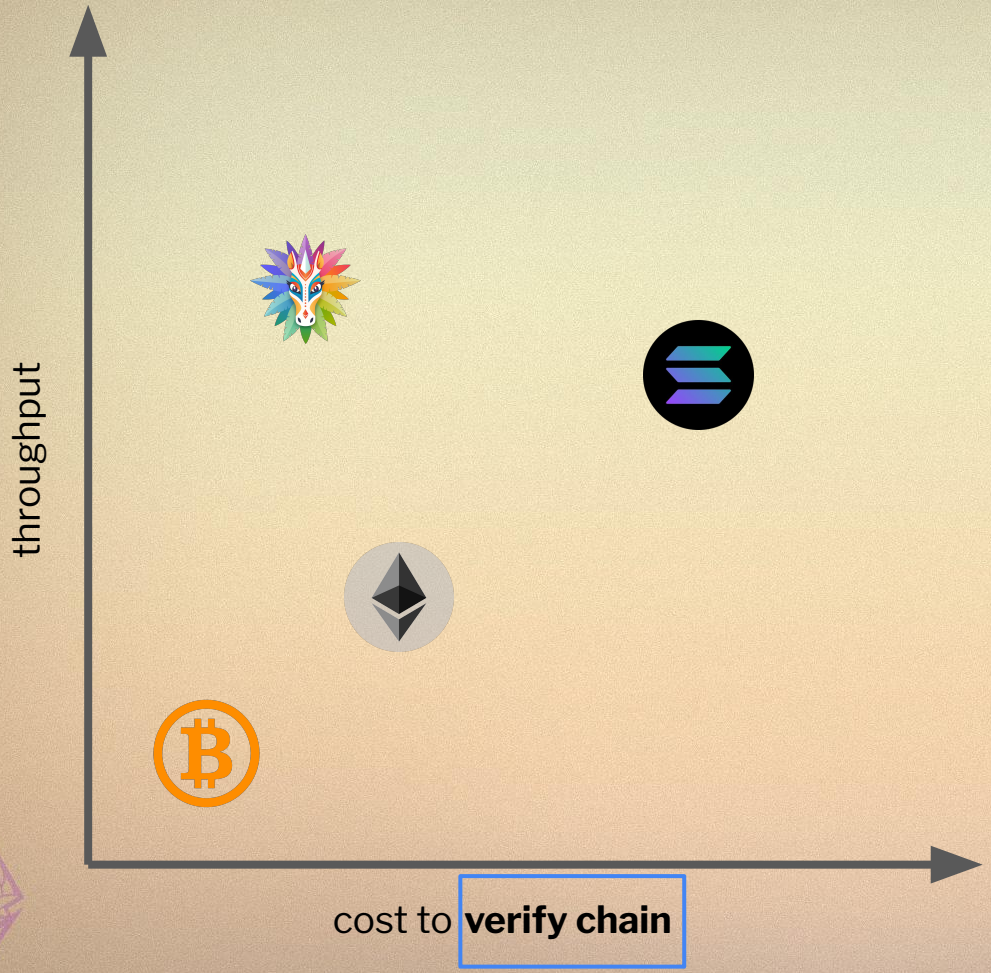
Ideal Chains

- everyone (who wants to) can run the chain locally
- very high throughput
- unicorn zone: how to get there?



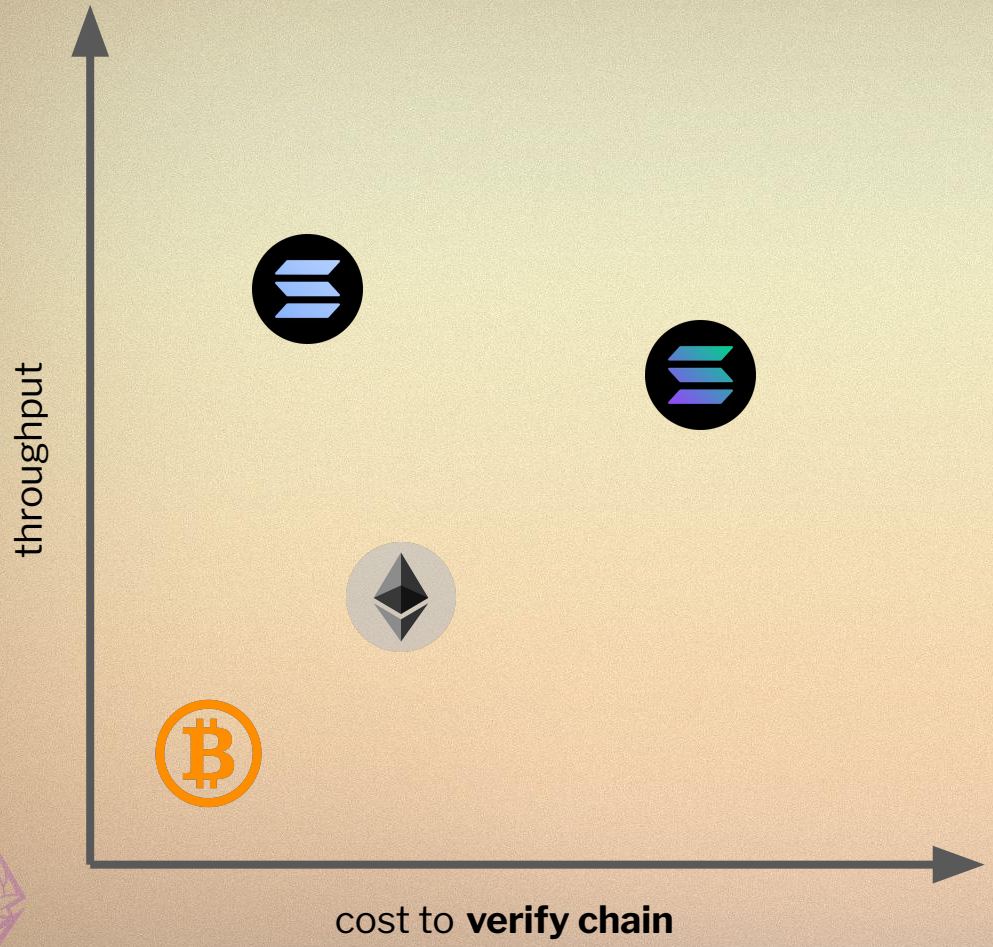
Ideal Chains

- everyone (who wants to) can run the chain locally
- very high throughput
- unicorn zone: how to get there?



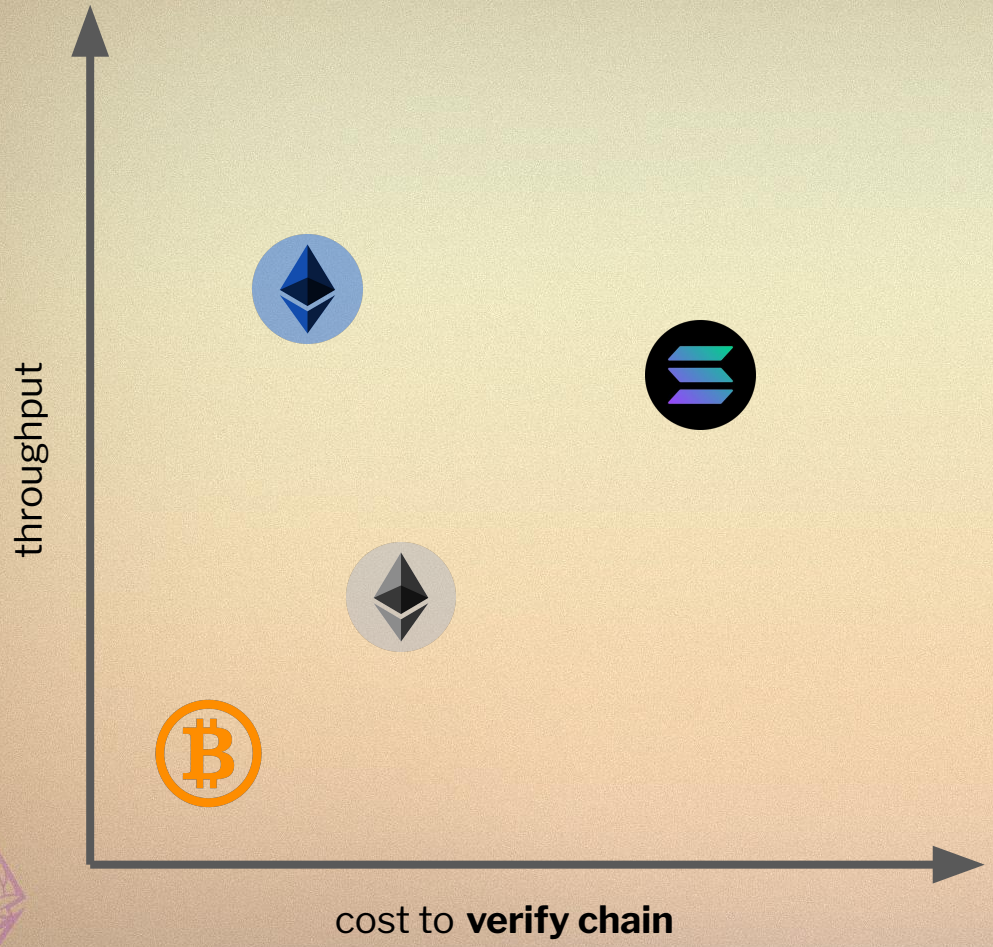
Ideal Chains

- everyone (who wants to) can run the chain locally
- very high throughput
- unicorn zone: how to get there?



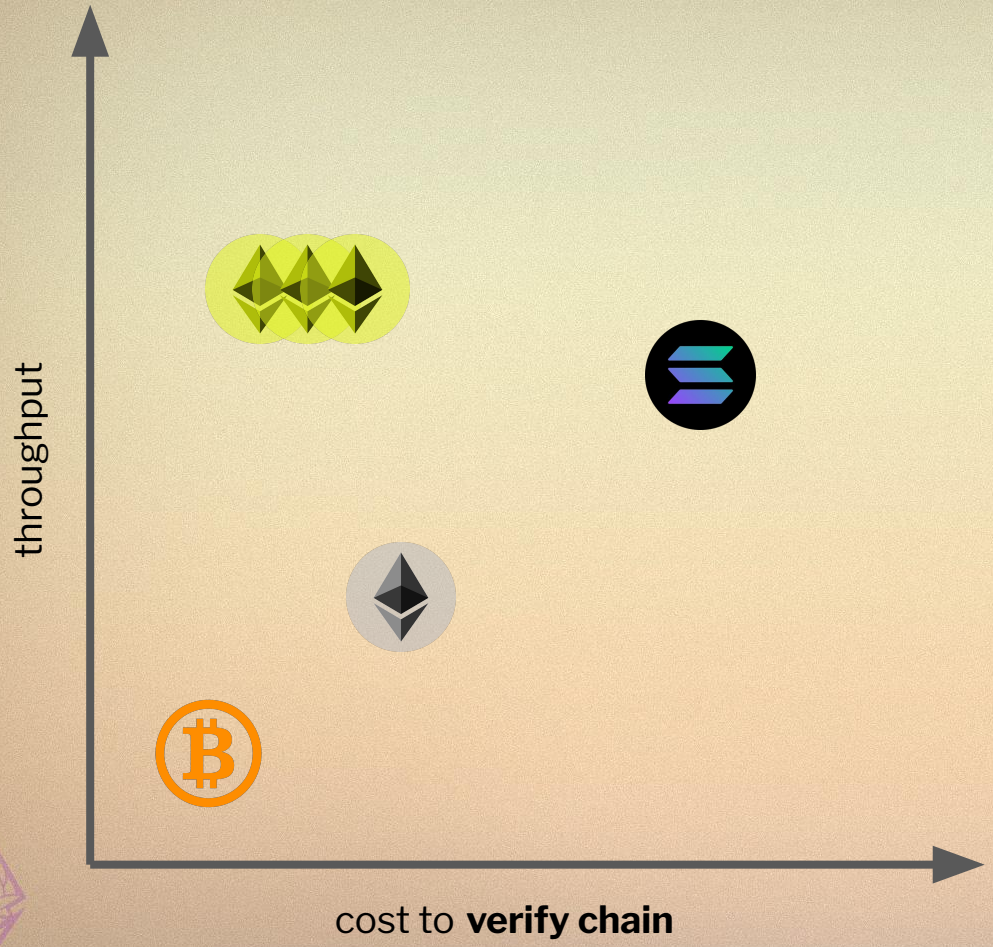
SVM Rollup

- SVM the most battle-tested high-throughput blockchain VM
- challenge: add fault or validity proofs



Fast EVM Rollup

- EVM the most popular blockchain VM
- many existing fault and validity proof mechanisms
- challenge: push EVM to its limits



Multi-Rollup Cluster

- consists of multiple smaller EVM chains
- can re-use existing off-the-shelf chains
- challenge: enable seamless cross-EVM interactions

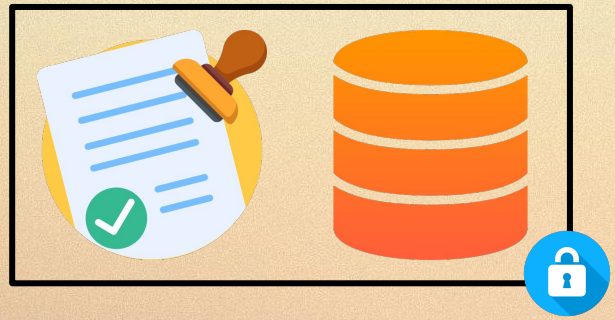
execution chain

execution chain

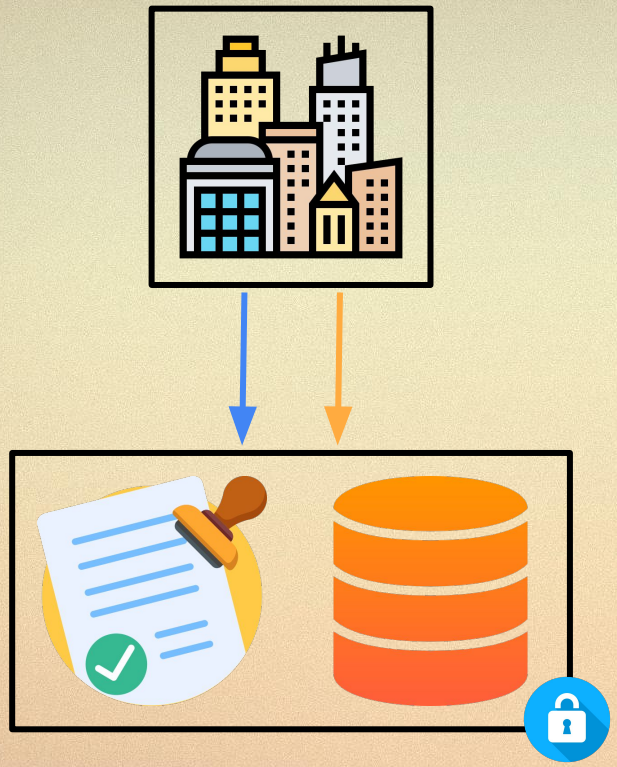
settlement

data availability

execution chain



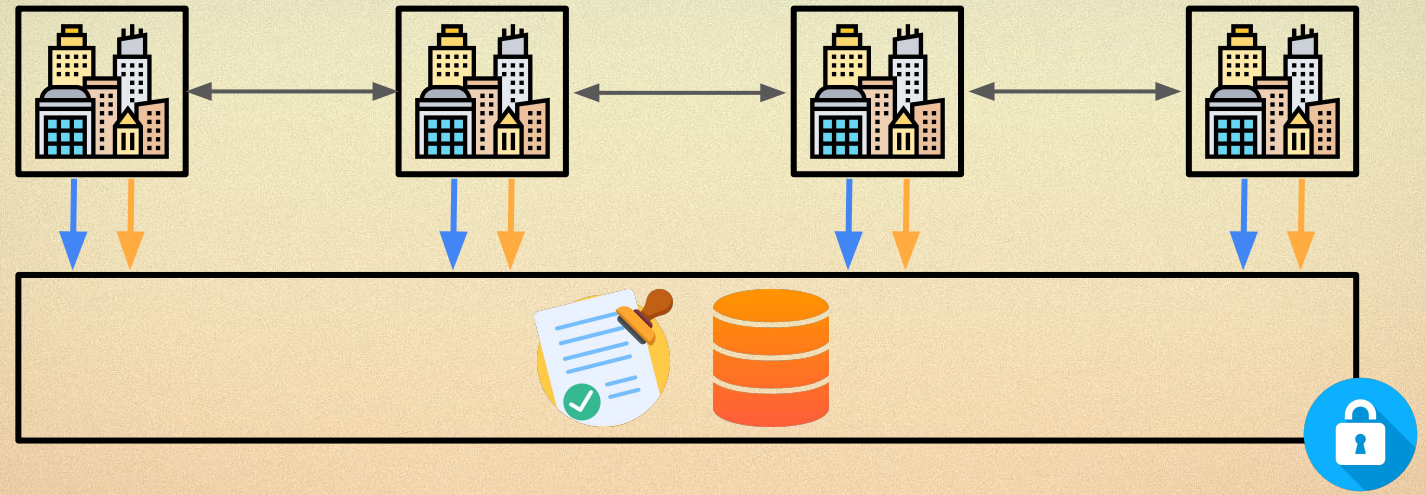
settlement chain



Rollup

- Inherits security from settlement chain
- Can safely run with higher full node requirements (and thus higher throughput)

Shared Settlement Chain

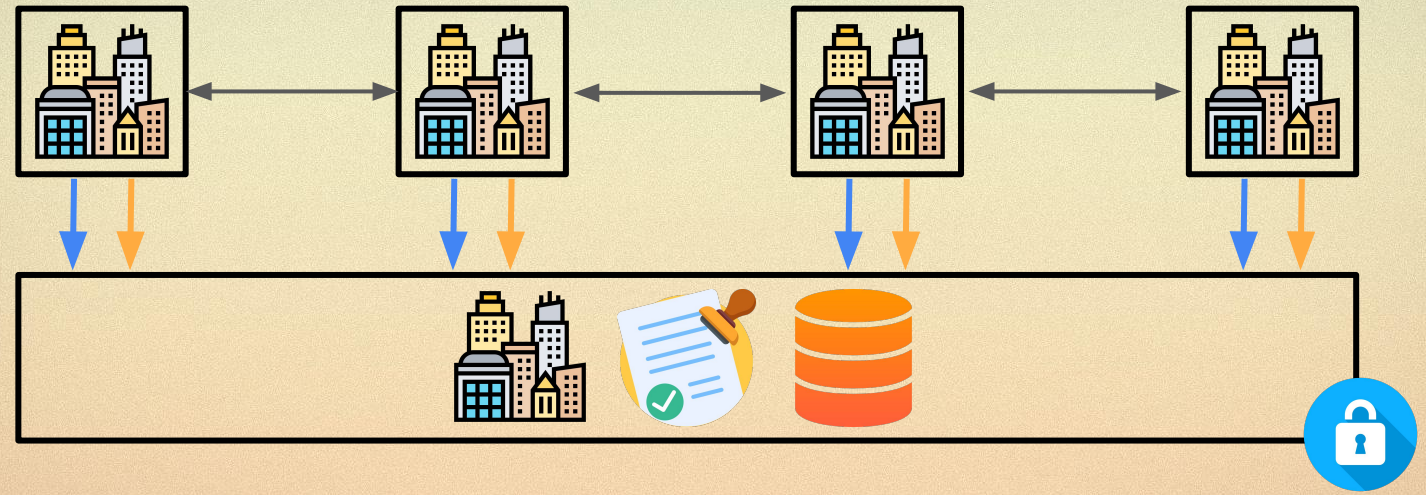


- shared (pooled) security
- secure bridges

Original Plan: Eth2

- new execution environments
- retire eth1 & EVM
- realization: eth1 “good enough” for settlement

Ethereum: Hybrid Execution & Settlement Chain



- low-throughput general purpose execution via EVM
- settlement bottleneck: data availability

EVM Equivalence

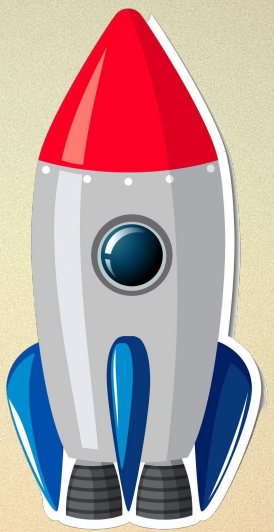
- expectation: L2s with novel execution environments
- instead: network effects **locking L2s into equivalence with L1**
- L2 innovation focus:
  - L2-specific functionality
  - EVM extensions
- geth as the primary L2 client

2. Present

Today: L1 EVM

- since merge, low frequency of new EL features
- several ambitious EL features explored and rejected
  - state expiry
  - native account abstraction
- new EL features mostly tailored to L1 needs:
  - EIP-7702
  - EOF
  - Verkle

Today: L2 EVM



- continued equivalence to L1
- **L1 <> L2 mismatch** in several dimensions:
  - **throughput ambitions**
  - **full node requirements**
  - **block building**
  - **zk**
- geth as main client focused on L1, very limited bandwidth for L2-specific features
- other clients start to gain L2 adoption

2. Future

Future: L2-native Client - rollup-geth

- **L2-focused fork of geth**
- initial collaboration between Nethermind and RollCall
  - team is part of Nethermind
  - strong prior EVM client experience
  - long-term commitment
- collaborative relationship with geth
- support for **L2-specific functionality**
  - optional, can stick with “vanilla” L1 mode

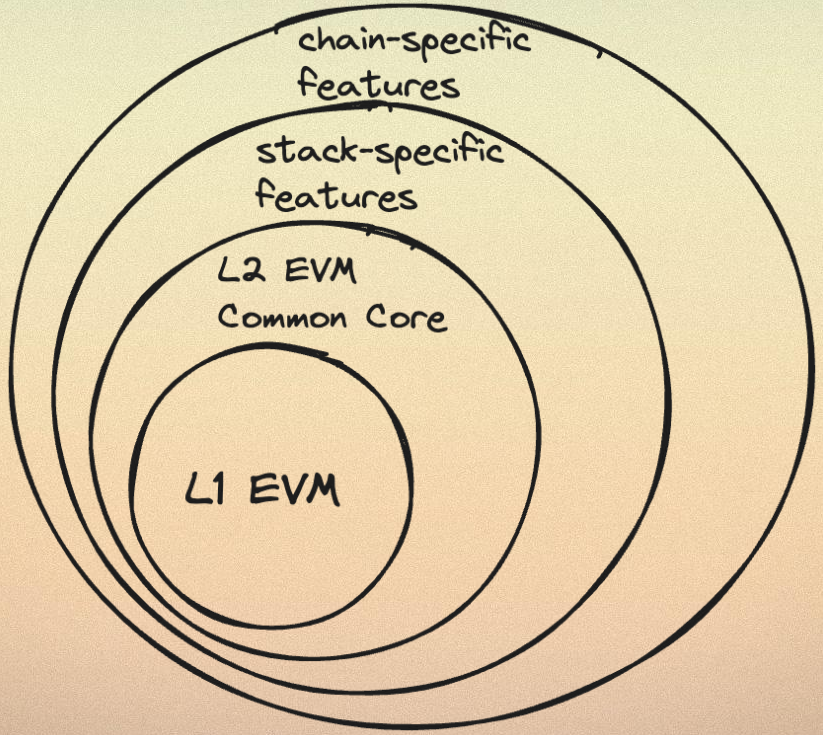
Future: L2-native Clients

- set of clients willing to support L2-specific EVM changes
- practical path towards **shipping RIPs**
- but: on its own **doesn't change L1 EVM equivalence lock-in**
    - good fit for isolated & "under the hood" features

Vision: L2 EVM Common Core

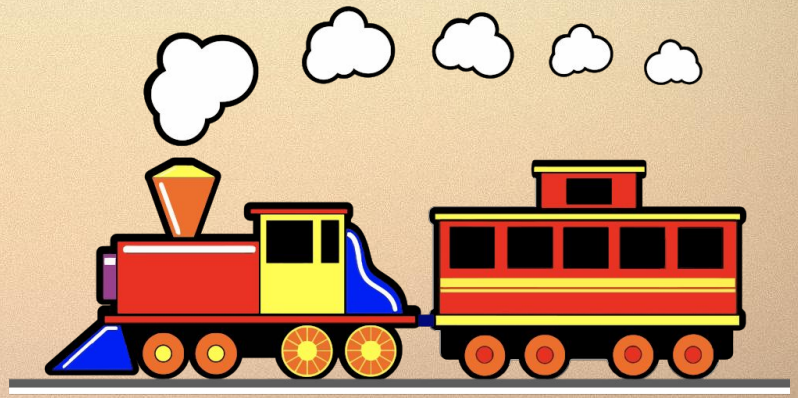
- **shared set of RIPs** supported by all participating L2s
- retain network effect: **still equivalent, but to each other**, not to L1
- become the **standard L2 target** for tooling, wallets, Solidity, …
- shared approach for how to handle upcoming L1 changes (Verkle etc.)
- goal: standardized, but ambitious
- needs governance

Vision: L2 EVM Common Core



Vision: L2 EVM Common Core - R&D

- near-term
  - basic repricing (e.g. zk)
  - 2d transaction type: L2 execution & L1 settlement
  - delayed state root computation
  - ...
- long-term
  - native account abstraction
  - transaction parallelization
  - statelessness / state expiry / state rent
  - multidimensional pricing
  - ...

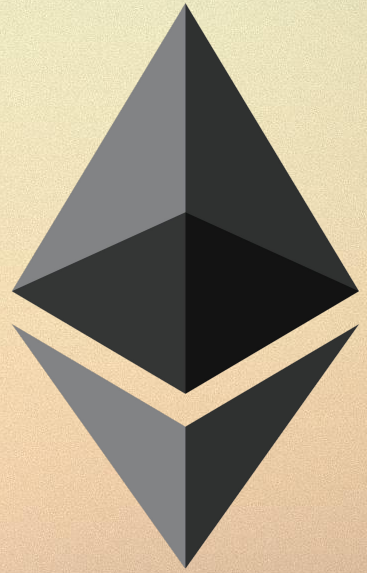


Vision: L2 EVM Common Core - Next Steps

- **determine interest / viability**
- figure out governance
- set up research process
  - initial common core scope
  - relationship zk / optimistic
  - **RollCall breakout call (December 4): common core research kickoff**

Vision: RollCall “Season 2”

- existing roles
  - L2 coordination
  - RIPs
  - L1 <> L2 connective tissue
- **new roles**
  - **L2 client coordination**
  - **L2 EVM common core governance**
  - **L2 EVM R&D**
- possibly split into separate processes



Thank you!

ansgar.eth
Research @ Ethereum Foundation

@ansgar.eth @adietrichs