

# Blockchain Scalability Trilemma

BSc Blockchain, Crypto Economy & NFTs

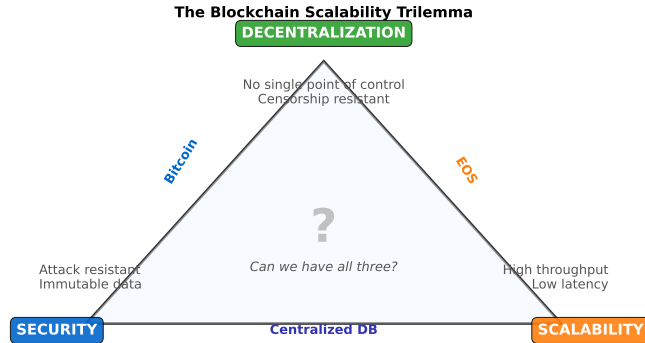
Course Instructor

Module A: Blockchain Foundations

By the end of this lesson, you will be able to:

- Explain the blockchain scalability trilemma
- Analyze trade-offs between security, decentralization, and scalability
- Understand Layer 1 scalability bottlenecks
- Compare throughput limitations across blockchains
- Evaluate vertical vs. horizontal scaling approaches
- Recognize emerging scalability solutions

# The Scalability Trilemma



*Coined by Vitalik Buterin: a blockchain can achieve at most two of three properties.*

## The Three Properties:

- 1 **Decentralization:** No single entity or small group controls the network
- 2 **Security:** Resistant to attacks, ensures data integrity
- 3 **Scalability:** High transaction throughput and low latency

## Trade-Off Examples:

- Bitcoin: Decentralized + Secure  $\rightarrow$  Low Scalability (7 TPS)
- EOS: Scalable + Secure  $\rightarrow$  Low Decentralization (21 block producers)
- Centralized Database: Scalable + Secure  $\rightarrow$  No Decentralization

## Core Challenge:

- Larger blocks  $\rightarrow$  fewer nodes can validate (centralization)
- Fewer validators  $\rightarrow$  faster consensus (centralization)
- No free lunch: optimizing one property often degrades another

# Why Each Property Matters

## Decentralization

- Censorship resistance
- Fault tolerance
- Trustlessness
- No single point of failure

## Security

- Immutability
- Double-spend prevention
- Attack resistance
- Data integrity

## Scalability

- High throughput (TPS)
- Low latency (finality)
- Low transaction costs
- Mass adoption support

## Why Blockchain Needs All Three:

- DeFi applications need high throughput AND security
- Mass adoption requires scalability without sacrificing decentralization
- Competition with traditional payment systems (Visa: 24,000 TPS)

## Fundamental Constraints:

### 1 Block Size:

- Larger blocks = more TXs, but slower propagation, higher storage
- Bitcoin: 1-4 MB — Ethereum: variable (gas limit)

### 2 Block Time:

- Faster blocks = higher throughput, but more orphans/forks
- Bitcoin: 10 min — Ethereum: 12 sec — Solana: 0.4 sec

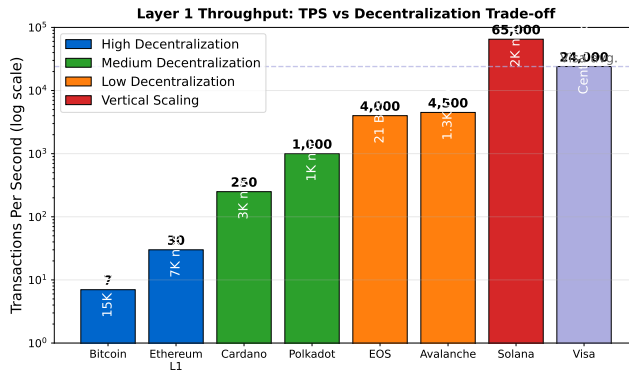
### 3 State Growth:

- Bitcoin: ~500 GB — Ethereum: ~1 TB (archive)
- Full nodes require significant storage

### 4 Computational Overhead:

- Every node verifies every transaction
- Smart contracts computationally expensive

# Layer 1 Throughput Comparison



*Inverse correlation: higher TPS generally means fewer nodes and lower decentralization.*

# The Block Size Debate: Bitcoin Case Study

## Background:

- Bitcoin originally: 1 MB block size limit
- As adoption grew: blocks filled up → higher fees, slower confirmations

## Big Block Proponents:

- Increase to 8 MB, 32 MB, or unlimited
- Immediate throughput increase, low fees
- Risk: centralization (fewer can run full nodes)

## Small Block Proponents:

- Keep blocks small to preserve decentralization
- Scale via Layer 2 (Lightning Network)
- Accept higher on-chain fees

## Outcome (2017):

- Bitcoin Cash hard fork (8 MB blocks)
- Bitcoin: SegWit soft fork (effective 1-4 MB)
- Market verdict: BTC prioritized decentralization, price  $\downarrow$  BCH

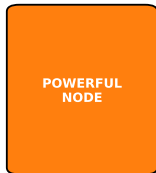


# Vertical vs. Horizontal Scaling

## Scaling Approaches: Trade-offs in Design

### VERTICAL SCALING

(Scale Up)



Bigger blocks  
Faster processing  
Higher hardware req.

Examples: Solana, EOS

Centralization risk

### HORIZONTAL SCALING

(Scale Out)



Parallel processing  
Preserved decentralization  
Complex coordination

Examples: Sharding, L2

Preserves decentralization

*Most major blockchains pursue horizontal scaling to preserve decentralization.*

## Vertical Scaling (Scale Up)

- Increase capacity of individual nodes
- Require more powerful hardware
- Examples: Solana, EOS

### Advantages:

- Simpler implementation
- Immediate throughput gains

### Disadvantages:

- Raises barrier to run nodes
- Centralization risk
- Hits physical limits

## Horizontal Scaling (Scale Out)

- Distribute load across many nodes
- Parallel processing
- Examples: Sharding, Layer 2, Rollups

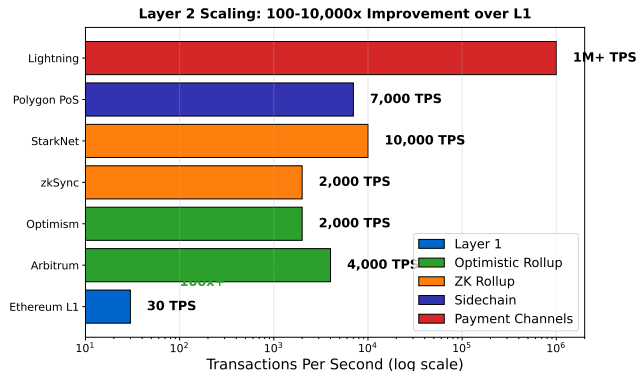
### Advantages:

- Preserves decentralization
- Theoretically unbounded scaling

### Disadvantages:

- Complex implementation
- Cross-shard communication overhead
- Longer development timelines

# Layer 2 Scaling Solutions



*Layer 2 achieves 100-10,000x improvement while inheriting L1 security.*

## 1. Payment Channels (Lightning Network):

- Open channel with on-chain TX, unlimited off-chain TXs
- Use case: micropayments, instant transfers
- TPS: theoretically unlimited

## 2. Rollups (Optimistic, ZK):

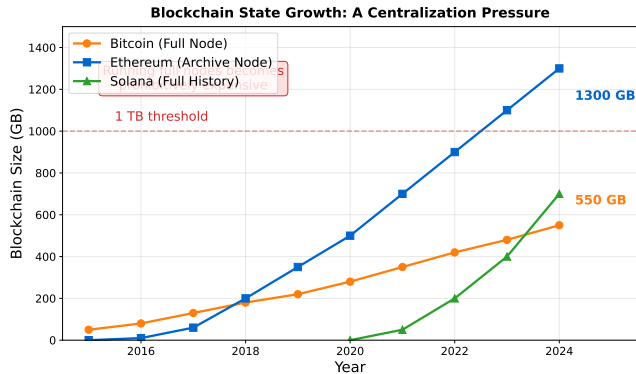
- Bundle hundreds of transactions into one
- Optimistic: assume valid, fraud proofs challenge
- ZK: cryptographic validity proofs
- Use case: DeFi, NFTs, general computation

## 3. Sidechains (Polygon):

- Independent blockchain with bridge to main chain
- Own consensus mechanism
- Weaker security than rollups (separate validator set)

**Current Adoption:** Ethereum L2 TVL: \$40B+ (2024)

# State Growth Problem



*Running full nodes becomes prohibitively expensive as state grows.*

## Challenge:

- Blockchain size grows unbounded
- Ethereum state: ~100 GB (accounts, storage)
- Syncing new nodes takes days/weeks
- Centralization pressure (only dedicated users run nodes)

## Solutions:

### 1 State Expiry (Ethereum proposal):

- Inactive state evicted from active storage
- Must provide proof to re-activate

### 2 Pruning:

- Discard old blockchain history
- Keep only recent blocks + current state

### 3 Rent (Cosmos, EOS):

- Users pay ongoing fees to store state
- Incentivizes cleanup of unused data

# Sharding: Horizontal Scaling at Layer 1

## Concept:

- Split blockchain into parallel “shards”
- Each shard processes subset of transactions
- Aggregate throughput = shards  $\times$  per-shard TPS

## Ethereum Sharding Roadmap:

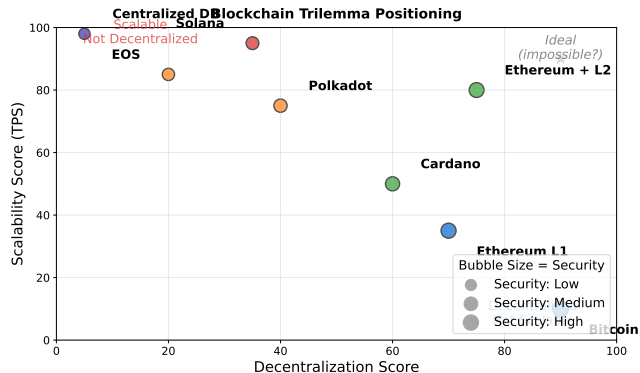
- Current plan: data availability sharding (danksharding)
- EIP-4844 proto-danksharding deployed March 2024
- Target: 100,000+ TPS via rollups + sharding

## Challenges:

- Cross-shard communication complexity
- Shard takeover attacks (low-stake shards vulnerable)
- State management (which shard holds which data?)

**Other Sharded Chains:** Zilliqa, NEAR Protocol, Elrond

# Trilemma Positioning



*Ethereum + L2 approaches the "ideal" corner without sacrificing security.*



# Breaking the Trilemma: Possible?

## Optimistic View:

- Layer 2 separates execution from settlement
- Rollups achieve scalability without sacrificing L1 security
- Data availability sampling enables sharding
- Zero-knowledge proofs compress computation

## Pessimistic View:

- Layer 2 introduces new trust assumptions (sequencers, bridges)
- Increased complexity → more attack vectors
- Still bound by data availability bandwidth

## Consensus View:

- Trilemma remains but can be mitigated
- Modular architecture (separate execution, consensus, DA)
- Accept specialization: L1 for security, L2 for scale
- Not “solving” trilemma but “navigating” it intelligently

- Scalability trilemma: cannot maximize decentralization, security, and scalability simultaneously
- Layer 1 bottlenecks: block size, block time, state growth
- Vertical scaling (bigger nodes) increases centralization
- Horizontal scaling (sharding, Layer 2) preserves decentralization
- Layer 2 solutions achieve 100-1000x throughput improvement
- State growth threatens long-term node decentralization
- Real-world TPS often far below theoretical limits

## Design Philosophy:

Accept that trade-offs exist. Bitcoin prioritizes decentralization/security. Solana prioritizes scalability. Ethereum aims for balance via Layer 2.

- ❶ Why cannot a blockchain simply increase block size indefinitely?
- ❷ How do Layer 2 solutions differ from sidechains in terms of security?
- ❸ What are the implications of state growth for decentralization?
- ❹ Can the trilemma be “solved” or only mitigated?
- ❺ How does sharding introduce new security challenges?
- ❻ Why do most blockchains have lower actual TPS than theoretical TPS?

### Lab activities:

- Navigate Etherscan and Blockstream block explorers
- Analyze transaction details (inputs, outputs, fees, confirmations)
- Trace transaction lifecycle from mempool to confirmation
- Examine block structure (header, transactions, miner rewards)
- Practice forensic blockchain analysis

### Preparation:

- Review Bitcoin transaction structure (Lesson 6)
- Ensure access to Etherscan.io and Blockstream.info