Blockchain Scalability Trilemma

BSc Blockchain, Crypto Economy & NFTs

Course Instructor

Module A: Blockchain Foundations

Learning Objectives

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
- Assess real-world blockchain performance

The Scalability Trilemma

Coined by Vitalik Buterin (Ethereum co-founder):

A blockchain can achieve at most two of these 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 -¿ Low Scalability (7 TPS)
- EOS: Scalable + Secure -¿ Low Decentralization (21 block producers)
- Centralized Database: Scalable + Secure -¿ No Decentralization (single operator)

Core Challenge:

- Increasing throughput typically requires:
  - Larger blocks -¿ fewer nodes can validate (centralization)
  - Fewer validators -¿ faster consensus (centralization)
  - Weaker security assumptions -¿ attack vulnerability (insecurity)
- No free lunch: optimizing one property often degrades another

Why Decentralization Matters

Benefits of Decentralization:

1. **Censorship Resistance:**
   - No single entity can block transactions
   - Critical for financial freedom
   - Enables permissionless participation
2. **Fault Tolerance:**
   - Network survives node failures
   - No single point of failure
   - Geographic redundancy
3. **Trustlessness:**
   - Users do not need to trust any central authority
   - Anyone can verify blockchain state
   - Reduces corruption and rent-seeking

Centralization Risks:

- Government coercion (shut down operators)
- Regulatory capture (compliance requirements exclude users)
- Monopolistic behavior (rent extraction, censorship)
- Trust assumptions (defeats blockchain purpose)

Why Security Matters

Security Requirements:

1. **Immutability:**
   - Confirmed transactions cannot be reversed
   - Historical data cannot be altered
   - Audit trail permanence
2. **Double-Spend Prevention:**
   - Same funds cannot be spent twice
   - Consensus ensures single valid transaction history
   - Critical for monetary applications
3. **Attack Resistance:**
   - 51% attack economically infeasible
   - Sybil attack prevented by consensus mechanism
   - Network remains available despite attacks

Security Failures:

- 51% attacks: Bitcoin Gold (2018), Ethereum Classic (2020)
- Smart contract exploits: The DAO (2016), Poly Network (2021)
- Weak consensus: various small chains (low hash rate)

Why Scalability Matters

Scalability Metrics:

1. **Throughput (TPS):**
   - Transactions processed per second
   - Bitcoin: 7 TPS
   - Visa: 24,000 TPS (peak: 65,000 TPS)
2. **Latency (Confirmation Time):**
   - Time until transaction finality
   - Bitcoin: 1 hour (6 confirmations)
   - Credit card: instant authorization (settlement in days)
3. **Cost per Transaction:**
   - Gas fees in Ethereum
   - Bitcoin transaction fees
   - Affects microtransaction viability

Why Blockchain Needs Scalability:

- Mass adoption requires handling millions of users
- DeFi applications need high throughput
- Micropayments require low fees
- Gaming and social media need instant confirmation
- Competition with traditional payment systems

Layer 1 Bottlenecks

Fundamental Constraints:

1. **Block Size:**
   - Larger blocks = more transactions per block
   - But: slower propagation, higher storage requirements
   - Bitcoin: 1-4 MB (SegWit), Ethereum: variable (gas limit)
   - Increasing block size reduces decentralization (fewer can run full nodes)
2. **Block Time:**
   - Faster blocks = higher throughput
   - But: more orphaned blocks (forks), lower security
   - Bitcoin: 10 min, Ethereum: 12 sec, Solana: 0.4 sec
   - Trade-off: speed vs. consensus stability
3. **State Growth:**
   - Blockchain size grows indefinitely
   - Bitcoin: 500 GB, Ethereum: 1 TB (archive node)
   - Full nodes require significant storage
   - Pruning reduces storage but limits historical queries
4. **Computational Overhead:**
   - Every node verifies every transaction
   - Smart contract execution computationally expensive
   - Parallel processing limited by sequential dependencies

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
- Community divided on solution

Big Block Proponents:

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

Small Block Proponents:

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

Outcome:

- 2017: Bitcoin Cash hard fork (8 MB blocks)
- Bitcoin: SegWit soft fork (effective 1-4 MB)
- Competing visions split community
- Bitcoin prioritized decentralization, BTC price ¿ BCH price (market verdict)

Vertical vs. Horizontal Scaling

Vertical Scaling (Scale Up)

- Increase capacity of individual nodes
- Require more powerful hardware
- Larger blocks, faster processing
- Examples: Solana, EOS

Advantages:

- Simpler implementation
- Immediate throughput gains
- No protocol changes needed

Disadvantages:

- Raises barrier to run nodes
- Centralization risk
- Hardware costs grow linearly
- Eventually hits physical limits

Horizontal Scaling (Scale Out)

- Distribute load across many nodes
- Parallel processing
- Sharding, Layer 2 solutions
- Examples: Ethereum sharding, Polkadot parachains

Advantages:

- Preserves decentralization
- Theoretically unbounded scaling
- Nodes remain affordable

Disadvantages:

- Complex implementation
- Cross-shard communication overhead
- Security challenges (shard attacks)
- Longer development timelines

Trend: Most major blockchains pursuing horizontal scaling (sharding, Layer 2) to preserve decentralization while increasing throughput.

Throughput Comparison

Layer 1 Throughput (TPS):

Blockchain	TPS	Block Time	Nodes
Bitcoin	7	10 min	15,000
Ethereum	30	12 sec	7,000
Litecoin	56	2.5 min	2,000
Cardano	250	20 sec	3,000
Polkadot	1,000	6 sec	1,000
Solana	65,000	0.4 sec	2,000
EOS	4,000	0.5 sec	21
Avalanche	4,500	1 sec	1,300
Visa (comparison)	24,000	instant	centralized

Observations:

- Inverse correlation: higher TPS -¿ fewer nodes (generally)
- Solana achieves high TPS via vertical scaling (expensive hardware)
- Bitcoin/Ethereum prioritize decentralization over throughput
- No blockchain matches Visa TPS at Layer 1 without centralization

Real-World Performance vs. Theoretical Limits

Advertised vs. Actual TPS:

Blockchain	Theoretical TPS	Actual Avg. TPS
Bitcoin	7	3-5
Ethereum	30	12-15
EOS	4,000	50-200
Solana	65,000	2,000-3,000
Cardano	250	10-20

Why the Discrepancy?

- **Network congestion:** actual demand varies
- **Transaction complexity:** simple transfers vs. smart contracts
- **Spam filtering:** anti-DoS measures limit throughput
- **Economic constraints:** high gas fees discourage frivolous transactions
- **Benchmarking conditions:** theoretical max assumes ideal conditions

Lesson: Advertised TPS often misleading. Real-world performance depends on transaction mix, network health, and economic factors.

Layer 2 Scaling Solutions

Concept:

- Move transactions off main chain (Layer 1)
- Settle periodically on Layer 1
- Inherit Layer 1 security guarantees

Types of Layer 2:

1. **Payment Channels (Lightning Network):**
   - Open channel with on-chain transaction
   - Unlimited off-chain transactions
   - Close channel to settle on-chain
   - Use case: micropayments, instant transfers
2. **Rollups (Optimistic, ZK):**
   - Bundle hundreds of transactions into one
   - Post compressed data to Layer 1
   - 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)
   - Use case: high-throughput applications

Layer 2 Throughput Comparison

Layer 2 Performance:

Layer 2	TPS	Security Model
Lightning Network (Bitcoin)	1,000,000+	Payment channels
Arbitrum (Ethereum)	4,000	Optimistic rollup
Optimism (Ethereum)	2,000	Optimistic rollup
zkSync (Ethereum)	2,000	ZK rollup
StarkNet (Ethereum)	10,000+	ZK rollup
Polygon PoS (sidechain)	7,000	Separate consensus
Ethereum Layer 1	30	Full security

Trade-offs:

- Rollups: inherit L1 security, moderate throughput (100-1000x)
- Sidechains: weaker security, higher throughput
- Payment channels: unlimited throughput, limited use case (payments only)

Current Adoption:

- Ethereum Layer 2 TVL (Total Value Locked): $40+ billion (2024)
- Lightning Network capacity: 5,000+ BTC
- Layer 2 becoming dominant for everyday transactions

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:

- Originally: 64 shards planned
- Current plan (post-Merge): data availability sharding (danksharding)
- Rollups use shards for data, execute off-chain
- Target: 100,000+ TPS via rollups + sharding
- Timeline: 2024-2026 (EIP-4844 proto-danksharding deployed March 2024)

Challenges:

- Cross-shard communication complexity
- Shard takeover attacks (low-stake shards vulnerable)
- State management (which shard holds which data?)
- Validator assignment (random, secure rotation)

Other Sharded Chains:

- Zilliqa: 4 shards, 2,500 TPS
- NEAR Protocol: dynamic sharding, 100,000 TPS target
- Elrond: 4 shards, 15,000 TPS

State Growth Problem

Challenge:

- Blockchain size grows unbounded
- Ethereum state: 100 GB (account balances, smart contract storage)
- Bitcoin UTXO set: 5 GB
- Storage costs increase over time

Consequences:

- Harder to run full nodes (requires SSDs, terabytes of storage)
- Syncing new nodes takes days/weeks
- Centralization pressure (only dedicated users run full nodes)

Solutions:

1. **State Expiry (Ethereum proposal):**
   - Inactive state evicted from active storage
   - Must provide proof to re-activate
   - Reduces live state size
2. **Pruning:**
   - Discard old blockchain history
   - Keep only recent blocks + current state
   - Trade-off: cannot verify full history
3. **Rent (Cosmos, EOS):**
   - Users pay ongoing fees to store state
   - Incentivizes cleanup of unused data

Nakamoto's Dilemma Formalized

Mathematical Model:

Let:

- f = block creation rate (blocks per second)
- β = block size (bytes)
- Δ = network propagation delay (seconds)

Security requires:

$$f \cdot \beta \cdot \Delta < \text{constant}$$

Interpretation:

- Increase throughput ($f \cdot \beta$) -¿ increase orphan rate
- High orphan rate -¿ security degrades (wasted hash power)
- To maintain security: throughput $\times$ latency $<$ threshold

Implications:

- Cannot arbitrarily increase block size or frequency
- Network latency sets fundamental limit
- Bitcoin conservatively below limit (prioritizes security)
- Solana pushes limit (requires fast network, powerful nodes)

Breaking the Trilemma: Possible?

Optimistic View:

- Layer 2 solutions separate execution from settlement
- Rollups achieve scalability without sacrificing L1 security/decentralization
- Data availability sampling enables sharding without full nodes downloading all data
- Zero-knowledge proofs compress computation (verify in constant time)

Pessimistic View:

- Layer 2 introduces new trust assumptions (sequencers, bridges)
- Increased system complexity -¿ more attack vectors
- Still bound by data availability bandwidth
- No escaping fundamental trade-offs, only shifting them

Consensus View:

- Trilemma remains but can be mitigated
- Modular architecture (separate execution, consensus, data availability)
- Accept specialization: Layer 1 for security, Layer 2 for scale
- Future: 100,000+ TPS with acceptable decentralization/security trade-offs
- Not “solving” trilemma but “navigating” it intelligently

Key Takeaways

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

Design Philosophy:
Accept that trade-offs exist. Choose priorities explicitly: Bitcoin prioritizes decentralization and security over scalability. Solana prioritizes scalability over decentralization. Ethereum aims for balance via Layer 2.

Discussion Questions

1. Why cannot a blockchain simply increase block size indefinitely to achieve scalability?
2. How do Layer 2 solutions differ from sidechains in terms of security?
3. What are the implications of state growth for blockchain decentralization?
4. Can the trilemma be "solved" or only mitigated?
5. How does sharding introduce new security challenges?
6. Why do most blockchains have lower actual TPS than theoretical TPS?
7. What role does network latency play in blockchain throughput limits?

Next Lesson Preview: L12 Lab - Block Explorer Analysis

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)
- Investigate address activity and balances
- Identify transaction patterns (exchanges, mixers, smart contracts)
- Practice forensic blockchain analysis

Preparation:

- Review Bitcoin transaction structure (Lesson 6)
- Ensure access to Etherscan.io and Blockstream.info
- Prepare questions about specific transactions or addresses to investigate