Blockchain Fundamentals

Lesson 2: How Blockchain Works

Prof. Joerg Osterrieder

Spring 2026

Learning Objectives

After this lesson, you will be able to:

- Explain the structure of a blockchain block
- Describe how blocks are linked using cryptographic hashes
- Understand the concepts of immutability and tamper-evidence
- Distinguish between public, private, and permissioned blockchains
- Explain how blocks are constructed, validated, and appended in a distributed network, and how competing chains are resolved via the longest chain rule

Prerequisites: Lesson 1 (Introduction to Cryptoeconomics)

This lesson provides the technical foundation for understanding blockchain

Lesson Outline

What is a Blockchain?

Blockchain: A Formal Definition

Definition: A blockchain is a distributed, append-only data structure where blocks of transactions are linked using cryptographic hashes.

Key Properties:

- **Distributed**: Copies stored across many nodes
- **Append-only**: New data added, never removed
- **Cryptographically linked**: Each block references the previous
- **Tamper-evident**: Any change is immediately detectable

Mathematical Notation:

$$\text{Block}_n = (\text{Hash}_{n-1}, \text{Timestamp}, \text{Nonce}, \text{Transactions})$$

The hash chain creates an immutable history of all transactions

From Ledgers to Blockchains

Traditional Ledger:

- Central authority maintains the "truth"
- Entries can be modified or deleted
- Trust depends on the institution

Blockchain Ledger:

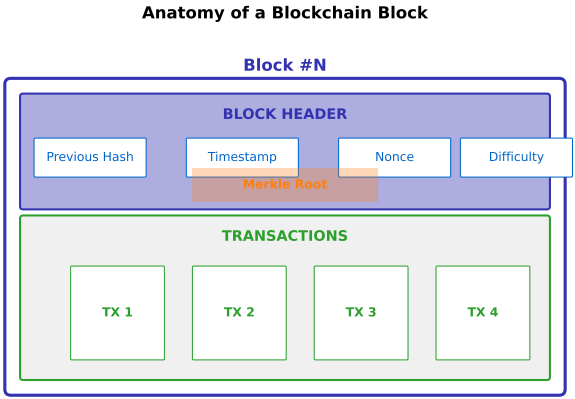
- Everyone maintains a copy
- Entries are permanent and verifiable
- Trust is cryptographically enforced

Analogy: Think of blockchain as a Google Doc where everyone can see all changes, but no one can delete history—and the document is cryptographically sealed.

Blockchain transforms trust from institutional to mathematical

Block Structure

Anatomy of a Block



Each block contains header metadata and a set of transactions

The Block Header

Block Header Components:

- **Previous Hash**: Links to the prior block (creates the chain)
- **Timestamp**: When the block was created
- **Merkle Root**: Fingerprint of all transactions in block
- **Nonce**: Number used for proof-of-work mining
- **Difficulty Target**: How hard the puzzle was to solve

Bitcoin Block Header Size: 80 bytes
Contains: Version (4B) + PrevHash (32B) + MerkleRoot (32B) + Time (4B) + Bits (4B) + Nonce (4B)

The header is what gets hashed to create the block's identity

Transaction Data

What Transactions Contain:

- **Inputs**: References to previous transaction outputs being spent
- **Outputs**: New ownership assignments (addresses and amounts)
- **Signatures**: Cryptographic proof of authorization

Special Transaction Type:

- **Coinbase Transaction**: The first transaction in every block, created by the miner, which generates new bitcoins as the block reward (has no inputs, only outputs)

Example Bitcoin Transaction:

- Alice has 1 BTC from a previous transaction
- She sends 0.5 BTC to Bob
- Change of 0.5 BTC goes back to Alice
- Transaction fee goes to miner

Transactions transfer ownership of digital assets

Merkle Trees: Efficient Verification

What is a Merkle Tree?

- Binary tree of transaction hashes
- Bottom: Individual transaction hashes
- Top: Single "Merkle Root" in block header

Why Merkle Trees?

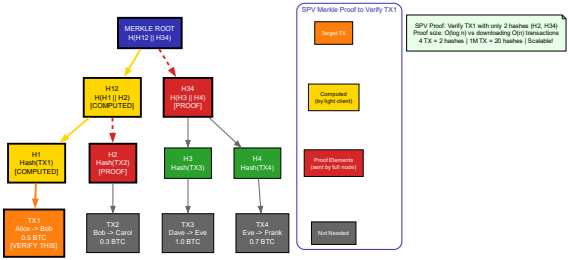
- Verify any transaction with $O(\log n)$ hashes
- Enable "light clients" without full blockchain
- Efficient proof that transaction is in block

Mathematical Property:

$$\text{Root} = H(H(H(TX_1)||H(TX_2))||H(H(TX_3)||H(TX_4)))$$

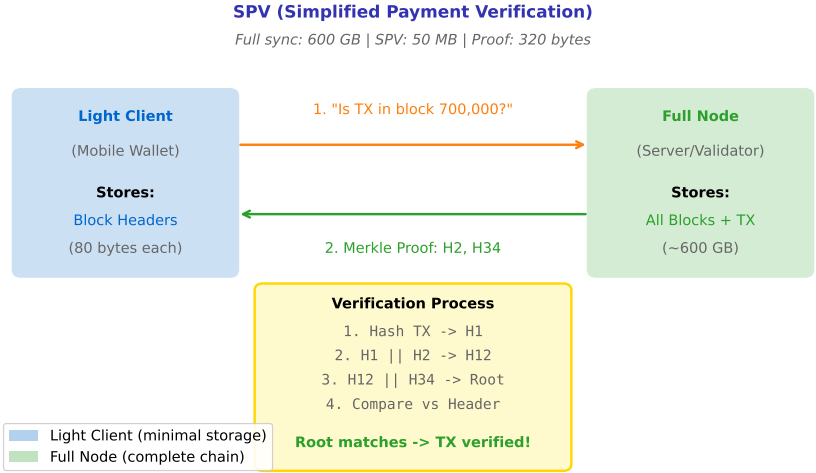
Merkle trees allow efficient and secure transaction verification

Merkle Tree Structure



Any change to a transaction changes the Merkle Root—tampering is immediately detectable

Simplified Payment Verification (SPV): Verifying Transactions Without Full Blockchain

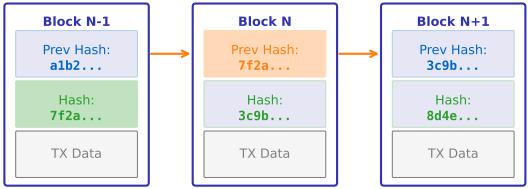


Merkle proofs enable mobile wallets to verify transactions with minimal data

The Chain of Blocks

How Blocks Link Together

Blockchain: Hash Chain + Collision Resistance



Collision Resistance: Why the Chain is Immutable

SHA-256 Properties

- Any change = completely different hash
- Cannot find two inputs with same hash
- Cannot reverse hash to find input

Security Implication

Attacker cannot:

- Modify TX and keep same hash
- Find collision in 2^128 years
- Break the chain link

Each block contains the hash of the previous block, creating an unbreakable chain

Cryptographic Hash Functions

Properties of Hash Functions:

- **Deterministic**: Same input always gives same output
- **One-way**: Cannot reverse to find input
- **Collision-resistant**: Extremely hard to find two inputs with same hash
- **Avalanche effect**: Small input change = completely different hash

SHA-256 Examples:

- "Hello" $\rightarrow$ `185f8db...` (64 hex chars)
- "Hello." $\rightarrow$ `4daa93b...` (completely different!)

Hash functions are the cryptographic glue holding blockchains together

Immutability: Why Tampering Fails

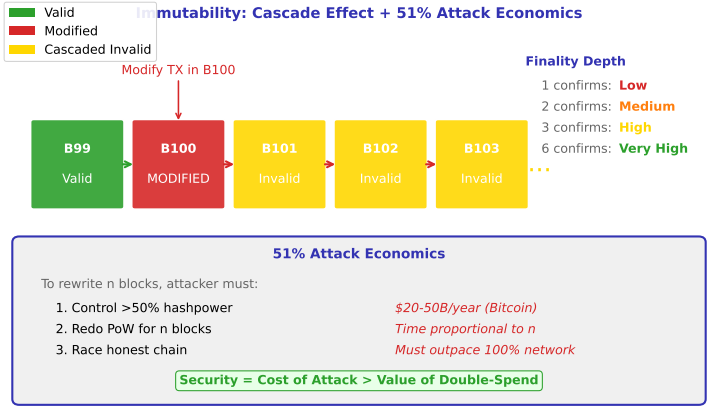
Scenario: Attacker Tries to Modify Block 100

1. Change a transaction in Block 100
2. Block 100's hash changes (avalanche effect)
3. Block 101's "Previous Hash" no longer matches
4. Block 101 is now invalid
5. Must recalculate Block 101's proof-of-work
6. Block 102's "Previous Hash" no longer matches...
7. Must redo ALL subsequent blocks!

Result: Attacker needs more computing power than entire honest network.

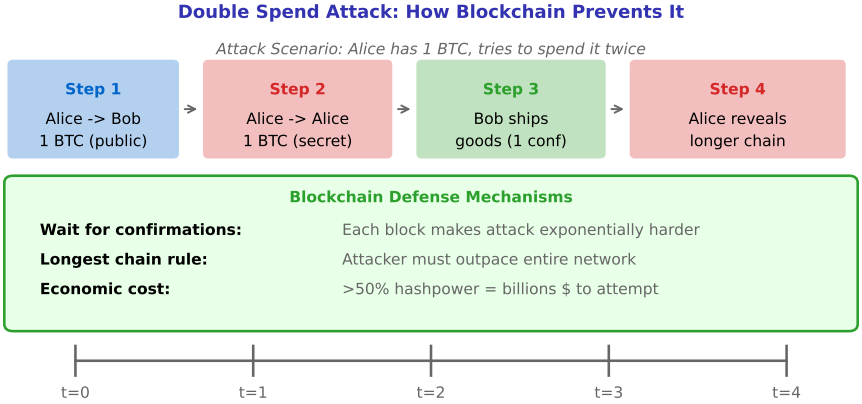
The deeper a block, the more secure it becomes

Visualizing the Cascade Effect



Modifying history requires redoing all subsequent proof-of-work—economically irrational

The Double Spend Problem



`P(success) = (q/p)^n where q=attacker, p=honest, n=confirmations`

6 confirmations: P < 1% with 10% hashpower (P ~1.7% with 30%)

Blockchain prevents double spending through confirmations and economic disincentives

The Genesis Block

Definition: The first block in a blockchain (Block 0 or Block 1).

Bitcoin Genesis Block (January 3, 2009):

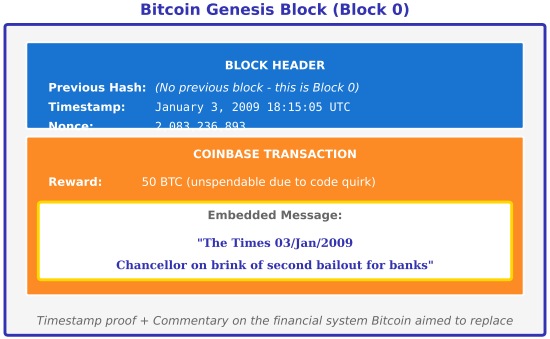
- Previous Hash: 0x0000...0000 (no previous block)
- Embedded message: "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks"
- Reward: 50 BTC (unspendable due to quirk)

Significance:

- Timestamps the beginning of Bitcoin
- Commentary on financial system Bitcoin aimed to replace
- All Bitcoin blocks trace back to this single origin

The Genesis Block is the anchor point of the entire blockchain

Bitcoin's Genesis Block



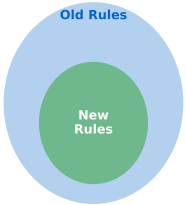
Block Hash: 000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

The embedded message timestamps Bitcoin's creation and hints at its purpose

Forks and Network Upgrades

Hard Forks vs Soft Forks

Soft Fork



New rules are STRICTER

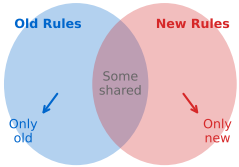
(subset of old rules)

Old nodes: accept new blocks

New nodes: stricter validation

Example: SegWit (Bitcoin 2017)

Hard Fork



New rules EXPAND beyond old

(not a subset)

Old nodes: reject new blocks

New nodes: reject old blocks

Example: Ethereum Classic (2016)

Forks occur when protocol rules change—soft forks are backward compatible, hard forks are not

Soft Forks: Backward Compatible Upgrades

Definition: A protocol change where new rules are a *subset* of old rules.

Characteristics:

- Old nodes accept blocks from new nodes
- New nodes may reject blocks from old nodes
- No chain split if majority upgrades

Examples:

- **SegWit (2017)**: Separated signature data, increased effective block size
- **Taproot (2021)**: Enhanced privacy and smart contract capabilities

---

Soft forks allow gradual network upgrades without forcing all nodes to update

Hard Forks: Breaking Changes

Definition: A protocol change that makes previously invalid blocks valid.

Characteristics:

- Old nodes reject blocks from new nodes
- Results in permanent chain split if not unanimous
- Creates two separate cryptocurrencies

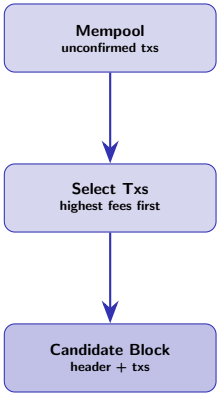
Notable Hard Forks:

- **Ethereum Classic (2016)**: Split after DAO hack reversal
- **Bitcoin Cash (2017)**: Increased block size to 8 MB
- **Bitcoin SV (2018)**: Further increased block size

Hard forks are contentious—they can divide communities and fragment network effects

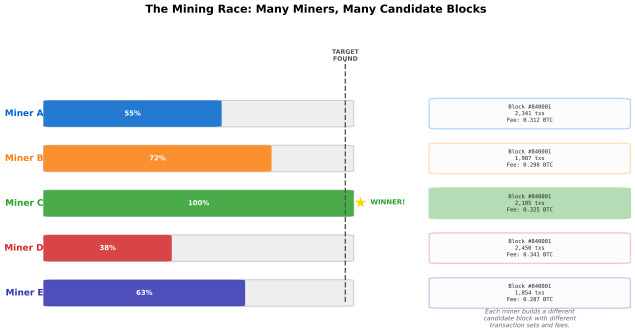
How Blocks Join the Chain

Building a Candidate Block



- Miner selects high-fee transactions from their local mempool
- Constructs **coinbase transaction** *(recall §2)* — block reward + fees to miner's address
- Computes **Merkle root** from all selected transactions
- Assembles **block header**: version, previous block

The Mining Race



Thousands of miners compete simultaneously — each trying different nonces on different candidate blocks

Block Found — Now What?

1. Miner finds nonce where $H(\text{header}) < \text{target}$ — the block is now a **valid block** *(a block whose header hash meets the difficulty target)*
2. Miner immediately broadcasts the block to all connected peers via **gossip broadcast** *(peer-to-peer relay where each node forwards to its neighbours)*
3. Each receiving node performs **independent validation** *(see next slide)*
4. If valid, node **appends** block to its local chain copy and forwards to its peers
5. If invalid, node **rejects** and does not forward — the block dies
6. All other miners **abandon** their current candidate block and start building on top of the new block

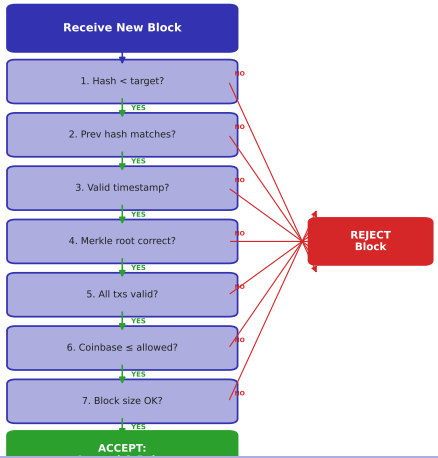
Speed matters — the miner who broadcasts first gets the reward; losers start over

How Nodes Validate a New Block

Block validation — independent verification by each node before accepting a block:

1. Block header hash $<$ difficulty target?
2. Previous block hash matches our chain tip?
3. Timestamp within acceptable range?
4. Merkle root matches included transactions?
5. All transactions individually valid? *(signatures, no double-spend)*
6. Coinbase reward $\leq$ allowed subsidy + fees?
7. Block size within limit?

Block Validation: Every Node Checks Independently

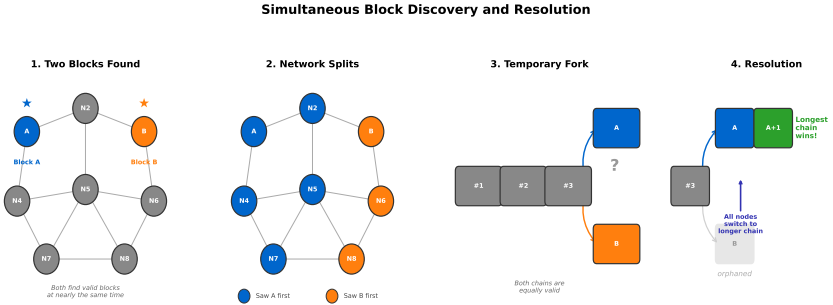


The Distributed View Problem

- In a distributed network, there is **no central authority** deciding the "true" chain
- Each node maintains its **own local copy** of the blockchain
- Network propagation takes **seconds to minutes** — not instant
- At any moment, different nodes may have different **chain tips** — the most recent block in a node's local copy of the blockchain
- This creates a fundamental question: **whose chain is "correct"?**
- The answer: no single chain is correct — the network achieves **eventual consistency**, the property where all nodes converge to the same state given enough time

The blockchain has no "master copy" — truth emerges from consensus, not authority

When Two Miners Win Simultaneously



A temporary fork — two competing valid chain tips existing simultaneously — is NORMAL; it happens every few days on the Bitcoin network

The Longest Chain Rule

The Rule:

- When a node sees two competing valid chains, it follows the one with the most cumulative proof-of-work — the **longest chain rule**
- "Longest chain" is shorthand — technically it is the **heaviest chain** *(most cumulative work, not just block count)*
- Miners always build on the longest chain they know about — this is **chain selection**
- When a longer chain arrives, nodes **automatically switch** — no vote needed

Why It Works:

- Honest miners have majority hash power $\Rightarrow$ their chain grows faster on average
- Miners always build on the longest chain because orphaned blocks earn no reward
- The probability of a fork persisting **decreases exponentially** with each subsequent block
- This creates **convergence**: the network reliably agrees on one chain without a coordinator

Satoshi's key insight: honest miners converge on one chain because building on the longest chain maximises reward

Chain Reorganizations (Reorgs)

What is a chain reorganization?
When the network switches to a longer valid chain, abandoning the previous tip.

How It Happens:

1. Two miners find valid blocks at similar times
2. Network temporarily splits (each node follows first seen)
3. One chain extends faster, becomes the "canonical" chain
4. Shorter chain is abandoned ("orphaned")

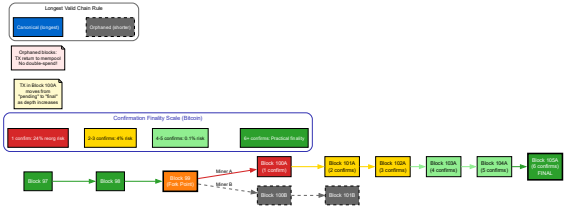
Implications:

- Transactions in orphaned blocks return to mempool
- Miners lose rewards from orphaned blocks
- Deep reorgs (6+ blocks) are extremely rare and concerning

51% Attack: An attacker with majority hash rate could intentionally cause deep reorgs.

Reorgs are normal at 1-2 blocks; deeper reorgs signal potential attacks

Visualizing Chain Reorganization



The longest valid chain rule ensures network eventually converges on one truth

Understanding Confirmations

What is a Confirmation?
Each new block added after your transaction = 1 confirmation.

Confirmation Depth and Security:

- **0 confirmations**: Unconfirmed (in mempool)
- **1 confirmation**: In a block, but could be reorged
- **6 confirmations**: Bitcoin standard for "finality"
- **Probabilistic finality**: $\approx (\frac{q}{p})^n$ where q = attacker hash rate — certainty of transaction permanence increases with each confirmation

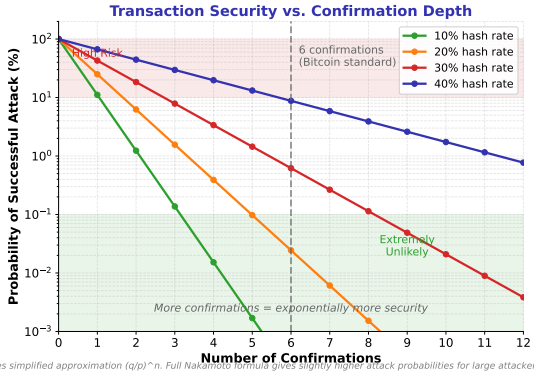
Why 6 Confirmations?
For an attacker with 10% hash rate: $P(\text{catch up after 6 blocks}) < 0.1\%$

Exchange Requirements:

- Bitcoin: Typically 3-6 confirmations (30-60 min)
- Ethereum: 12-35 confirmations (3-7 min)

More confirmations = higher security = longer wait time

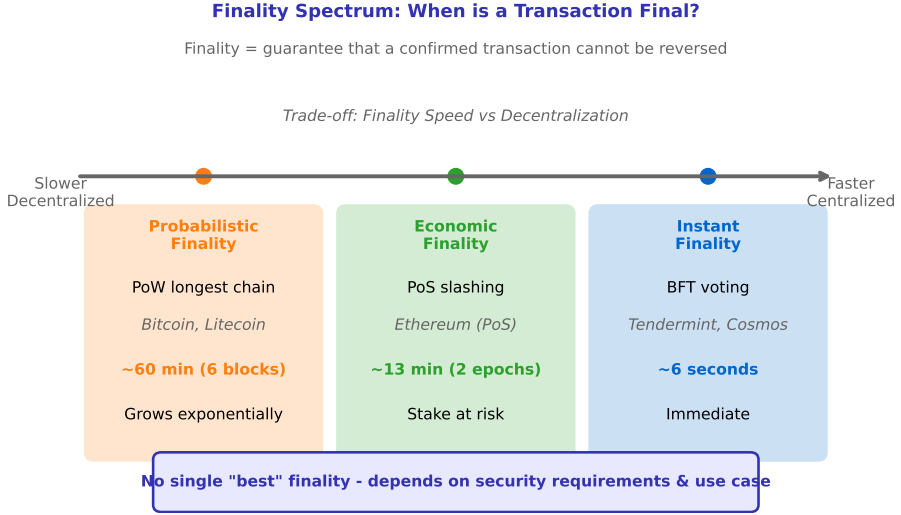
Confirmation Security: The Math



Note: Uses simplified approximation (q/p)^n. Full Nakamoto formula gives slightly higher attack probabilities for large attacker hashrates (>30%).

Security increases exponentially with each confirmation—6 blocks is the standard for finality

The Finality Spectrum



Different consensus mechanisms offer different finality guarantees and trade-offs

Nakamoto Consensus — Putting It All Together

The Process:

1. Miners compete to find valid blocks (PoW)
2. Winner broadcasts; nodes validate independently
3. Temporary forks resolve via longest chain rule
4. **Probabilistic finality** increases with each confirmation
5. Economic incentives align miners toward honesty

(see Lesson 4: Consensus Mechanisms for PoW details)

Properties of Nakamoto consensus:

- **Permissionless**: Anyone can participate without approval
- **Trustless**: No node trusts another — all verify independently
- **Eventually consistent**: Network converges on one chain
- **Probabilistically final**: Reversal becomes exponentially unlikely
- **Economically secure**: Attacking costs more than it gains

Nakamoto consensus: the first solution to distributed agreement in an open, adversarial network

Distributed Networks

Network Architecture Types

Centralized:

- Single point of control and failure
- Examples: Traditional banks, social media

Decentralized:

- Multiple independent nodes
- No single point of failure
- Examples: Email servers

Distributed (Blockchain):

- Every node has full copy
- Consensus determines truth
- Examples: Bitcoin, Ethereum

Blockchain combines decentralization with cryptographic guarantees

Types of Nodes

Full Nodes:

- Store complete blockchain history
- Validate all transactions and blocks
- Enforce consensus rules

Light Nodes (SPV):

- Store only block headers
- Trust full nodes for transaction validity
- Lower resource requirements

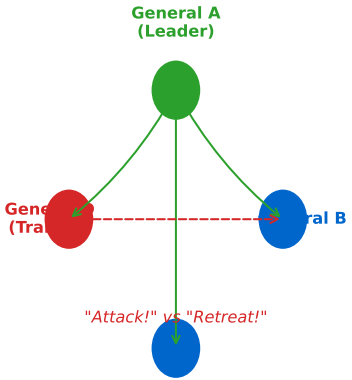
Mining/Validator Nodes:

- Full nodes that also create new blocks
- Compete for block rewards
- Secure the network

Different node types serve different roles in the network

Byzantine Fault Tolerance in Blockchain

Byzantine Generals Problem



Problem: Reach consensus with up to f traitors among n generals

Solution requires: n >= 3f + 1

Blockchain Solution (Nakamoto Consensus)

Probabilistic Consensus

- No fixed membership (permissionless)
- PoW ties votes to computational work
- Longest chain = most accumulated work
- Honest majority => eventual consistency

Traditional BFT vs Nakamoto Consensus

	Classical BFT	Nakamoto
Membership	Fixed (known)	Open (anyone)
Fault Tolerance	n >= 3f+1	>50% honest
Finality	Immediate	Probabilistic
Scalability	Low (O(n^2))	High (O(n))

Bitcoin: First practical solution to BFT in open, adversarial networks

How Transactions Propagate

Transaction Lifecycle:

1. User creates and signs transaction
2. Broadcasts to connected nodes
3. Nodes validate and relay to peers using a **gossip protocol**—each node forwards new information to its connected peers, who forward to their peers, rapidly spreading data across the entire network
4. Transaction enters mempool (waiting area)
5. Miner includes in block
6. Block propagates across network
7. Transaction confirmed when block is accepted

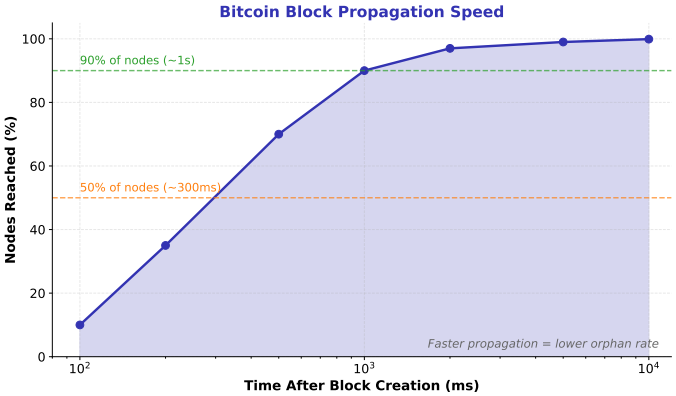
Typical Propagation Time: Seconds to minutes depending on network congestion.

Peer-to-peer gossip ensures transactions reach all nodes

Transaction Lifecycle Flowchart

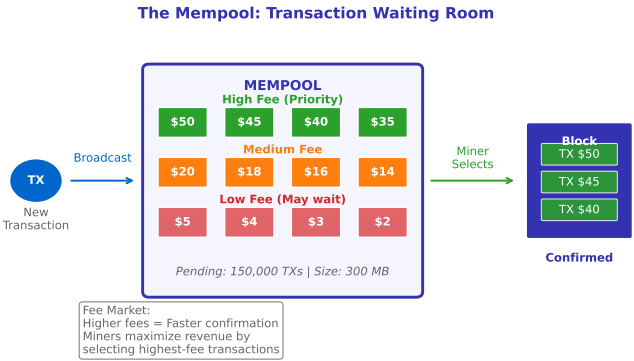
From creation to confirmation: the complete journey of a blockchain transaction

Block Propagation Speed



Faster propagation reduces orphan blocks and increases network efficiency

The Mempool: Transaction Waiting Room



Miners select highest-fee transactions first—creating a fee market for block space

Understanding the Mempool

What is the Mempool?

- Unconfirmed transactions waiting to be included in a block
- Each node maintains its own mempool
- Size varies based on network activity

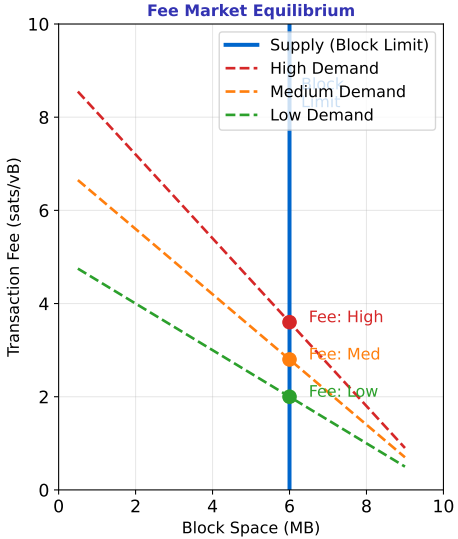
Fee Market Dynamics:

- **High congestion**: Fees spike as users compete for block space
- **Low congestion**: Minimum fees sufficient
- **Bitcoin 2024**: Fees ranged from \$0.50 to \$50+ during peaks

Replace-By-Fee (RBF): Users can "bump" stuck transactions with higher fees.

Understanding mempool dynamics helps users optimize transaction timing and costs

Fee Market Game Theory



Fee Bidding Game

You \ Other Users	Low Fee	High Fee
Low Fee	Wait (maybe)	Long wait
High Fee	Fast confirm	Fast (overpay)

Nash: Bid marginal increment above others

Cryptoeconomic Insight

Fee market aligns incentives:

Users pay for urgency | Miners maximize revenue

Fee markets align user urgency with miner revenue incentives

Blockchain Types

Public Blockchains

Characteristics:

- Anyone can participate (read, write, validate)
- Fully transparent—all transactions visible
- No central authority controls access
- Security through economic incentives

Examples:

- Bitcoin: Digital gold, store of value
- Ethereum: Smart contract platform
- Solana: High-speed transactions

Trade-offs: Maximum decentralization but lower throughput

Public blockchains prioritize censorship resistance and openness

Private Blockchains

Characteristics:

- Single organization controls access
- Participants are known and vetted
- Higher transaction throughput
- Privacy of data maintained

Examples:

- Hyperledger Fabric: Enterprise solutions
- R3 Corda: Financial institutions
- Internal corporate ledgers

Trade-offs: Centralized but efficient and private

Private blockchains are essentially distributed databases with crypto features

Permissioned (Consortium) Blockchains

Characteristics:

- Group of organizations share control
- Known validators, but no single owner
- Balanced decentralization and efficiency
- Suitable for industry consortia

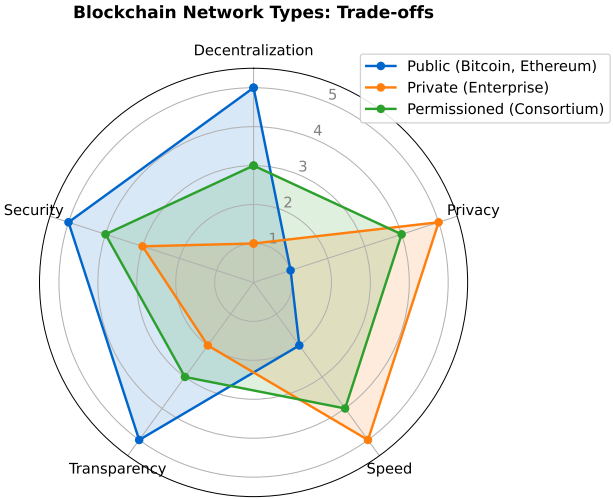
Examples:

- Trade finance networks (we.trade)
- Supply chain tracking (TradeLens)
- Healthcare data sharing

Trade-offs: Moderate decentralization with good performance

Consortium blockchains balance trust among competing organizations

Network Types: Trade-offs



Choose blockchain type based on your specific requirements

Real-World Implementation

Bitcoin's Implementation

Technical Specifications:

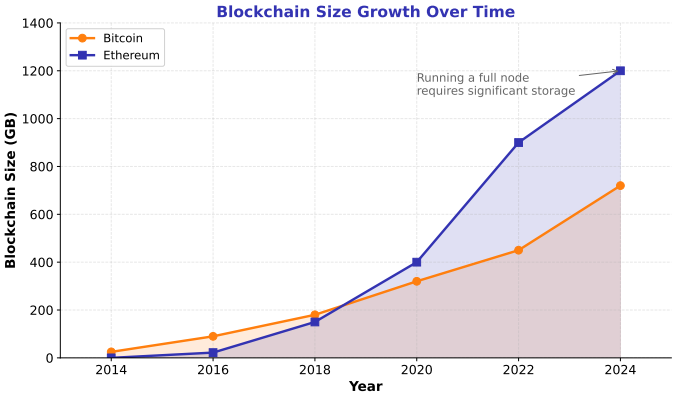
- Block size: 1 MB (with SegWit 4 MB effective)
- Block time: 10 minutes average
- Transactions per block: 2,000-3,000
- Total supply: 21 million BTC

Current State (2024):

- Blockchain size: approximately 720 GB (as of January 2026)
- Network hash rate: 900 EH/s
- Active addresses: Millions daily

Bitcoin has operated continuously since 2009 with near-perfect uptime

Blockchain Storage Requirements



Running a full node requires significant and growing storage capacity

Ethereum's Implementation

Technical Specifications:

- Block time: 12 seconds
- No fixed supply cap
- Gas system for computation pricing
- Smart contract execution

Key Differences from Bitcoin:

- Account-based vs. **UTXO (Unspent Transaction Output)** model—Bitcoin's approach to tracking spendable coins where each UTXO is an unspent output from a previous transaction, consumed entirely when spent
- Turing-complete smart contracts
- Proof of Stake (since 2022)
- Faster blocks, more complex state

Ethereum extends blockchain from ledger to general-purpose computation

The Scalability Challenge

Current Limitations:

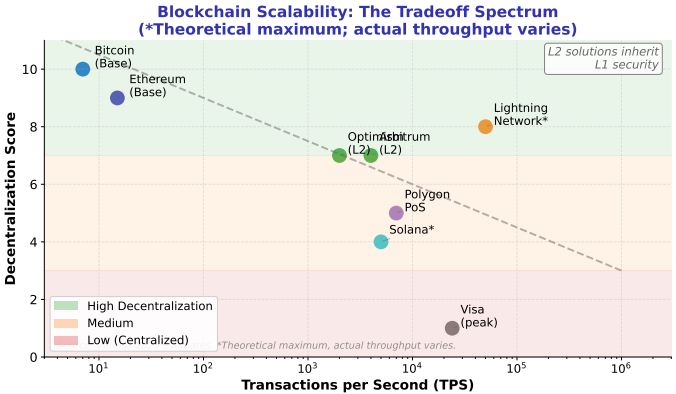
- Bitcoin: 7 transactions per second
- Ethereum: 15-30 transactions per second
- Visa: 24,000 transactions per second

Scaling Solutions:

- **Layer 2**: Lightning Network, **Rollups**
- **Sharding**: Split blockchain into pieces
- **Bigger Blocks**: More data per block (trade-off: centralization)

Scaling while maintaining decentralization is blockchain's key challenge

Scalability vs Decentralization Tradeoffs



Higher TPS often comes at the cost of decentralization—L2 solutions aim to break this tradeoff

Layer 2 Scaling Solutions

Lightning Network (Bitcoin):

- Off-chain payment channels
- Instant, near-zero fee transactions
- Final settlement on Bitcoin mainnet

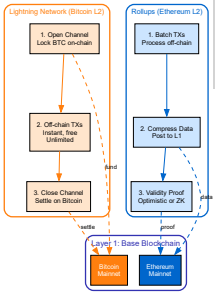
Rollups (Ethereum):

- **Optimistic Rollups**: Assume valid, challenge if fraud (Arbitrum, Optimism)
- **ZK-Rollups**: Cryptographic validity proofs (zkSync, StarkNet)
- Process transactions off-chain, post proofs on-chain

Key Insight: L2 inherits L1 security while dramatically increasing throughput.

Ethereum's roadmap centers on L2 scaling—"rollup-centric" future

Layer 2 Architecture Comparison



Feature	Lightning	Rollups
Speed	Instant	~1-10 min
Fees	Near zero	Low
Capacity	Limited by channels	High throughput
Smart Contracts	Limited	Full support
Security	Inherits Bitcoin	Inherits Ethereum

Both solutions: Process transactions off-chain, inherit security from base layer

Both Lightning and Rollups inherit security from their base layers while scaling throughput

How Optimistic Rollups Work

Execution Steps:

- **Step 1:** Users submit transactions to an L2 **sequencer**
- **Step 2:** Sequencer batches transactions and posts compressed data to L1
- **Step 3: Challenge period** ($\approx$7 days) — anyone can submit a fraud proof
- **Step 4:** No valid challenge $\Rightarrow$ finalized; fraud detected $\Rightarrow$ batch rolled back

Key Players: Arbitrum, Optimism, Base

Trade-offs:

- Lower fees and higher throughput than L1
- 7-day withdrawal delay due to challenge window

"Optimistic" means batches are assumed valid by default; economic incentives reward challengers who catch fraud

How ZK-Rollups Work

Execution Steps:

- **Step 1:** Transactions executed off-chain by a **prover**
- **Step 2:** A **validity proof** (SNARK or STARK) is generated over the batch
- **Step 3:** Proof + compressed state submitted to an L1 smart contract
- **Step 4:** L1 verifies the proof mathematically — **instant finality**, no challenge period

Key Players: zkSync Era, StarkNet, Polygon zkEVM, Scroll

Trade-offs:

- Faster finality than optimistic rollups; withdrawals in minutes
- Proof generation is computationally expensive (but improving rapidly)

ZK-rollups are widely considered Ethereum's long-term scaling solution; cryptographic soundness replaces the need for a dispute window

Rollup Comparison and Data Availability

Optimistic vs. ZK-Rollups:

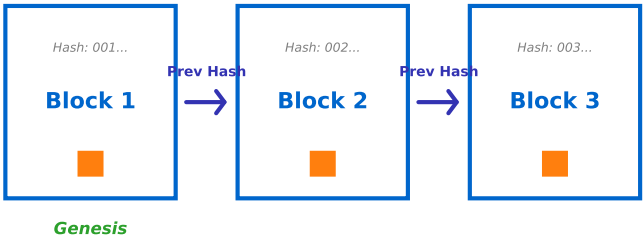
- **Proof type:** Fraud proof (optimistic) vs. validity proof (ZK)
- **Finality:** $\approx$7 days (optimistic) vs. minutes (ZK)
- **Computation:** Simple execution (optimistic) vs. complex proof generation (ZK)
- **EVM compatibility:** Full via OVM (optimistic) vs. growing via zkEVM (ZK)

Data Availability (DA) — where is transaction data stored?

- **On-chain DA (rollups):** Data posted to L1 — most secure, higher cost
- **Off-chain DA (validiums):** Data stored externally — cheaper, but adds trust assumptions
- **EIP-4844 "proto-danksharding":** Introduces **blob transactions** to drastically reduce DA costs for rollups

Ethereum's rollup-centric roadmap treats rollups as the primary scaling layer; EIP-4844 (Cancun upgrade, 2024) reduced rollup fees by $\sim 10\times$

Blockchain Structure



Data Sources & Further Learning

Technical Documentation:

- Bitcoin Developer Documentation
- Ethereum.org Technical Specifications

Blockchain Explorers & Analytics:

- Blockchain.com Explorer

Interactive Learning Tools:

- Anders Brownworth Blockchain Demo

Research & Development:

- MIT Digital Currency Initiative

These resources provide hands-on exploration and up-to-date technical information

Key Takeaways

What You Learned Today:

1. Blocks contain headers (metadata) and transactions (data)
2. Cryptographic hashes link blocks into an unbreakable chain
3. Changing any block invalidates all subsequent blocks
4. Three types: Public (open), Private (single org), Permissioned (consortium)
5. Nakamoto consensus enables trustless agreement: miners compete $\Rightarrow$ nodes validate $\Rightarrow$ longest chain wins $\Rightarrow$ probabilistic finality increases with each confirmation

Core Insight:
Blockchain achieves immutability through the combination of cryptographic hashing and economic incentives—changing history is mathematically possible but economically irrational.

Next lesson: Cryptographic Foundations

Questions for Reflection

1. Why does changing one block require redoing all subsequent blocks?
2. What are the trade-offs between public and private blockchains?
3. How does a Merkle tree enable efficient transaction verification?
4. Why might an organization choose a consortium blockchain?

Discussion Question:
If private blockchains are more efficient, why do public blockchains have more value?

Consider these questions before our next session

Thank You

Questions?

Course materials: digital-ai-finance.github.io/crypto-economics

Appendix: Technical Deep Dives

Case Study: The Bitcoin Block Size Wars (2015-2017)

The Conflict:

- Bitcoin's 1 MB block limit created congestion as adoption grew
- Two camps: "Big Blockers" vs "Small Blockers"

Big Blockers (Bitcoin Cash):

- Increase block size to 8 MB (later 32 MB)
- Prioritize on-chain scaling and low fees
- Concern: Larger blocks reduce node decentralization

Small Blockers (Bitcoin Core):

- Keep blocks small, scale via Layer 2
- SegWit: "Soft fork" increasing effective capacity
- Prioritize decentralization over raw throughput

The debate resulted in Bitcoin Cash hard fork (August 2017)—a fundamental disagreement on blockchain philosophy

Blockchain Statistics (2024)

Bitcoin Network:

- Total nodes: 25,000 reachable (est. 50,000+ total)
- Blockchain size: approximately 720 GB (as of January 2026)
- Hash rate: 900 EH/s (900 quintillion hashes/sec)
- Energy consumption: 150 TWh/year

Ethereum Network:

- Total validators: 900,000 (post-Merge PoS)
- Total staked: 30 million ETH ($90B)
- Daily transactions: 1.1 million
- Active addresses: 500,000 daily

Sources: blockchain.com, etherscan.io, bitnodes.io (January 2024)

These networks process billions of dollars in value daily with near-perfect uptime