

# Bitcoin Commons: A Cryptographic Governance System for Bitcoin Development

BTCDecoded

2025



## Abstract

Bitcoin faces a critical governance asymmetry: while its technical consensus layer is cryptographically bulletproof, its development governance relies on informal social coordination. At Bitcoin's multi-trillion dollar scale, this represents an existential vulnerability.

This whitepaper presents Bitcoin Commons, a complete technical and governance solution that applies Elinor Ostrom's proven commons governance principles through cryptographic enforcement and modular architecture. The system provides formal mathematical specifications, multisig governance requirements, and self-sustaining economics through merge mining revenue.

The system is implemented across seven public repositories, providing mathematical specifications, governance infrastructure, and economic sustainability. For the complete narrative treatment, see *Bitcoin Commons: Decentralizing the Decentralizers*.

## 1. Introduction

Bitcoin solved Byzantine consensus between strangers but ignored consensus between developers. The network's substantial market capitalization demands institutional maturity matching its technical excellence.

The original cypherpunk developers focused on eliminating trusted third parties in transactions but inadvertently created trusted parties in development. This whitepaper presents the complete technical and governance solution to Bitcoin's most critical vulnerability.

### 1.1 The Talent Bottleneck: Orders of Magnitude and Sources

Bitcoin development draws on multiple hard domains simultaneously (C++, applied cryptography, distributed systems, security engineering, economics/game theory, and open-source governance). Each extra domain narrows the pool. Using conservative, sourced baselines and clearly labeled assumptions, we estimate the rarity of a contributor who combines these competencies and is available to work on Bitcoin:

#### Assumptions and sources:

- World population baseline: ~8.1B (UN DESA, World Population Prospects, 2022 Rev.)
- Global developers: ~30M-47M (range spanning widely cited industry estimates, incl. SlashData and similar studies)
- C++ share of developers: ~15%-25% (range spanning major annual developer surveys)
- Adult numeracy (problem-solving proficiency): on the order of 10%-20% globally (OECD PIAAC cross-country evidence; global extrapolation is approximate)
- Bitcoin Core maintainers: single-digit individuals; contributors: hundreds (public repo statistics)

#### Rarity funnel (indicative, overlapping, not strictly independent):

- Strong college-level math (calculus/linear algebra): 3%-5% of population, resulting in 240M-400M
- Professional developers: ~30M-47M (subset, separate baseline)
- C++/systems competency: 15%-25% of developers → 4.5M-11.8M
- Applied cryptography + Bitcoin protocol literacy: 1%-2% of C++ devs → 45k-236k

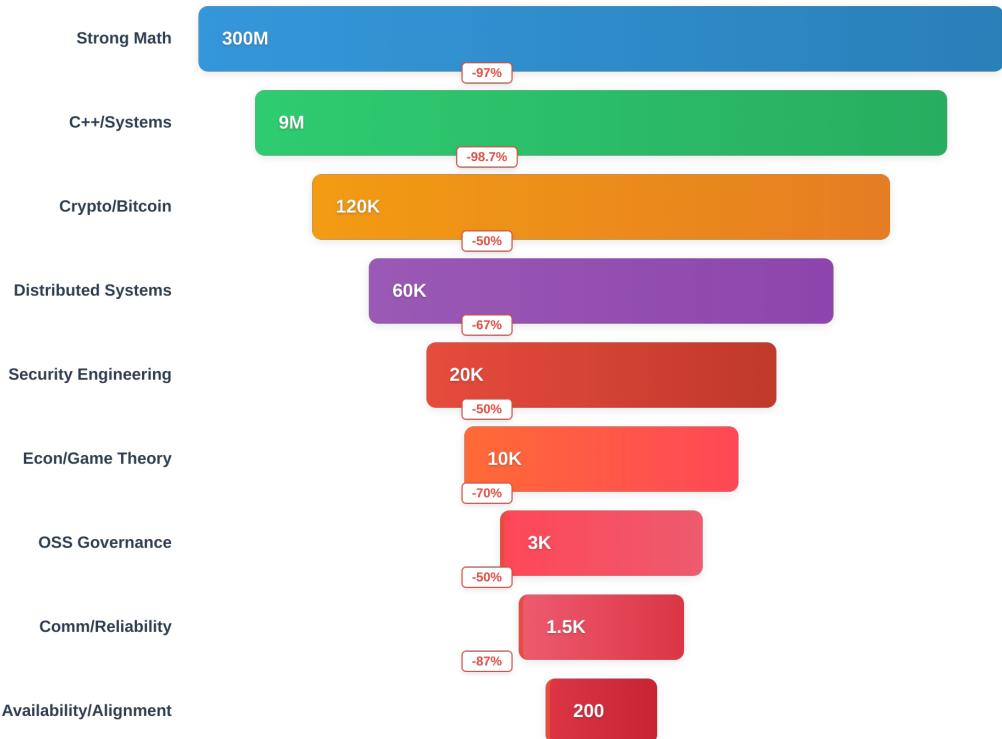
- Distributed systems/P2P networking depth: 30%-50% → 13.5k-118k
- Security engineering mindset (memory safety, adversarial thinking): 20%-30% → 2.7k-35k
- Economics/game-theory literacy: 30%-50%, resulting in 0.8k-17.5k
- Open-source governance (review culture, consensus norms): 10%-30%, resulting in 80-5k
- Communication/reliability under public scrutiny: 30%-50% → 24-2.5k
- Availability/alignment to actually work on Bitcoin: 10%-30% → ~2-750

### **Interpretation:**

- Even with generous ranges, the intersection yields on the order of dozens to a few hundred globally available individuals with the full stack to work reliably on Bitcoin's most sensitive layers.
- Public data corroborates scarcity at the tip: Bitcoin Core has hundreds of credited contributors but only a small, rotating single-digit maintainer set. This human bottleneck contrasts with the cryptographic abundance at the consensus layer.

# Talent Bottleneck Funnel

Indicative orders-of-magnitude scarcity across required competencies



*Figure: Orders of magnitude funnel: combining C++, cryptography, systems, security, economics, and governance yields extreme scarcity. This visualization shows how each domain narrows the available talent pool.*

**Citations** (illustrative anchors): - United Nations DESA, World Population Prospects (2022 Rev.) - SlashData, Global Developer Population Trends - Annual developer surveys (e.g., Stack Overflow) for C++ usage bands - OECD PIAAC, adult skills numeracy distributions (cross-national) - Bitcoin Core repository contributor and maintainer counts (public GitHub data)

## 2. Problem Statement

### Technical Reality

Bitcoin's consensus rules are embedded in 350,000+ lines of C++ code with no mathematical specification. Bitcoin Core maintains 99.5% market share among implementations, creating effective monopoly control over Bitcoin's evolution. The lack of formal specification makes it impossible to build safe alternative implementations or verify consensus correctness.

## **Governance Reality**

Bitcoin's development governance relies entirely on informal social coordination. There are no systematic consequences for bad actors, no formal dispute resolution mechanisms, and power is invisible and unaccountable. The system is vulnerable to capture through relationships rather than rules.

## **Historical Context**

Early developers recognized this problem. Gavin Andresen raised governance concerns in his 2014 Princeton talk but was marginalized during the blocksize wars. Mike Hearn attempted governance solutions but proposed hierarchical models inappropriate for Bitcoin's decentralized ethos. Academic researchers (De Filippi & Loveluck, 2016) documented these power structures but provided no actionable solutions.

## **Scale Considerations**

Bitcoin's growth from early stages to multi-trillion dollar scale requires institutional reform. The next crisis, whether AI attacks, regulatory capture, or internal conflicts, won't wait for the community to develop governance solutions reactively.

## **3. Theoretical Framework: The Triple Foundation**

Bitcoin Commons synthesizes three distinct theoretical frameworks, each addressing weaknesses in the others to create governance architecture stronger than any single approach alone.

### **Framework 1: Elinor Ostrom - Commons Governance**

Elinor Ostrom won the 2009 Nobel Prize in Economics for proving that shared resources don't inevitably collapse into chaos or capture. Her research documented principles for governing commons without central authority across centuries of real-world examples.

#### **Ostrom's Seven Principles:**

- 1. Clear boundaries on who decides what** - Defined decision-making authority
- 2. Consequences for violations** - Systematic enforcement mechanisms
- 3. Local dispute resolution** - Formal conflict resolution processes
- 4. Protection from external interference** - Resistance to outside pressure
- 5. Collective choice arrangements** - Meaningful participation in rule-making
- 6. Graduated sanctions** - Proportional consequences for violations
- 7. Monitoring and accountability** - Transparent oversight mechanisms

**What This Provides:** Proven institutional design for shared resources, evidence that decentralized governance can work, framework for coordination without hierarchy.

### Framework 2: F.A. Hayek - Spontaneous Order

Friedrich Hayek's Austrian economics provides the competitive discovery mechanism that enables governance evolution rather than rigid design.

**Hayek's Core Insights:** - **Dispersed Knowledge Problem** - No central planner can know what's needed because knowledge is distributed across many actors - **Competition as Discovery** - Competition reveals information that couldn't be known in advance - **Spontaneous Order** - Best institutions emerge through evolution, not top-down design - **Markets Need Infrastructure** - Competition requires actual alternatives to compete

**What This Provides:** Justification for avoiding central planning, framework for competitive governance discovery, understanding that institutions should evolve through market signals.

### Framework 3: Bitcoin - Cryptographic Enforcement

Bitcoin's innovation provides the enforcement tools that make decentralized governance work at scale without trusted parties.

#### Bitcoin's Core Principles:

- **Don't Trust, Verify** - Cryptographic enforcement replaces social trust
- **Permissionless Innovation** - Anyone can build without asking permission
- **Exit Rights** - Fork option provides ultimate check on power
- **Decentralized Control** - No single point of authority

**What This Provides:** Tools for enforcing rules without trust, proof that decentralized systems can work at scale, model for implementing Hayek's principles digitally.

### The Triple Synthesis

The three frameworks address each other's weaknesses:

**Ostrom's Challenge:** Commons governance historically relied on social pressure, vulnerable to capture at scale **Bitcoin's Solution:** Cryptographic enforcement replaces social pressure with mathematical proof

**Hayek's Challenge:** Competition discovers optimal solutions but requires actual alternatives to compete

**Ostrom's Solution:** Provides institutional framework for multiple governance models to coexist

**Bitcoin's Challenge:** Solved technical consensus but not social governance **Hayek + Ostrom Solution:** Competitive discovery of governance models using proven institutional principles

**The Result:** Governance that is proven (Ostrom's research), evolving (Hayek's competition), and enforceable (Bitcoin's cryptography).

## **Bitcoin Core's Current State**

Bitcoin Core has informal implementations of some Ostrom principles but lacks systematic enforcement. The system has informal boundaries (Core maintainers, BIP editors) but no formal process for selection, removal, or authority limits. Social pressure and reputation damage provide consequences, but there's no systematic enforcement mechanism. Most critically, there's no infrastructure for competitive discovery. Core's 99.5% market share prevents Hayekian competition from working.

## **Mapping Ostrom's Principles to Implementation**

The modular architecture implements Ostrom's seven principles through cryptographic enforcement rather than social pressure. Below is the explicit mapping of each principle to technical implementation:

Principle	Source	Core	Commons
Clear Boundaries	Ostrom	✓	✓✓✓
Proportional Benefits	Ostrom	✓	✓✓✓
Collective Choice	Ostrom	✓	✓✓✓
Monitoring	Ostrom	✓	✓✓✓
Graduated Sanctions	Ostrom	✓	✓✓✓
Conflict Resolution	Ostrom	✓	✓✓✓
Minimal Recognition	Ostrom	✓✓	✓✓✓
Spontaneous Order	Hayek	✓	✓✓✓
Knowledge Problem	Hayek	✓	✓✓✓
Price Discovery	Hayek	✓	✓✓✓
Market Process	Hayek	✓	✓✓✓
Permissionless	Bitcoin	✓	✓✓✓
Trust Minimization	Bitcoin	✓	✓✓✓
Decentralization	Bitcoin	✓	✓✓✓
Censorship Resistance	Bitcoin	✓	✓✓✓
Immutability	Bitcoin	✓	✓✓✓
Privacy by Default	Cypherpunk	✓	✓✓✓
Technical Enforcement	Cypherpunk	✓	✓✓✓
Anti-Surveillance	Cypherpunk	✓	✓✓✓

Figure: Integration of four key philosophies: Hayek (spontaneous order), Bitcoin (cryptographic enforcement), Cypherpunk (privacy through technology), and Ostrom (commons governance).

## Principle 1: Clear Boundaries

- **Base Layer:** Network consensus required for any changes
- **Module Layer:** User choice determines adoption (composable configuration)
- **Economic Layer:** Miner coordination shapes development priorities
- **Implementation:** Architecture enforces boundaries: modules cannot modify consensus code paths

## Principle 2: Consequences for Violations

- **Economic Consequences:** Merge mining leverage enables economic pressure without consensus changes

- **Technical Consequences:** Module quality standards and security audits
- **Reputational Consequences:** Adoption metrics transparent, market signals visible
- **Implementation:** Cryptographic enforcement makes consequences automatic, not social

### Principle 3: Local Dispute Resolution

- **Architecture as Resolution:** Disputes about features result in building competing modules
- **Module Competition:** User choice determines winners, not political consensus
- **Base Layer Protection:** Module conflicts don't threaten consensus layer
- **Implementation:** No central arbiter needed: architecture provides resolution through user configuration

### Principle 4: Protection from External Interference

- **Self-Funding:** Merge mining revenue enables independence from external pressure
- **Distributed Keyholders:** Multi-jurisdictional distribution prevents single-point coercion
- **Fork-Ready Design:** Ultimate protection through governance fork while preserving consensus
- **Implementation:** Cryptographic multisig ensures no single jurisdiction can compel action

## Keyholder Diversity Radar

Phase 3 estimates. Higher is better across axes: jurisdictions, org diversity, rotation cadence, independence, quorum.

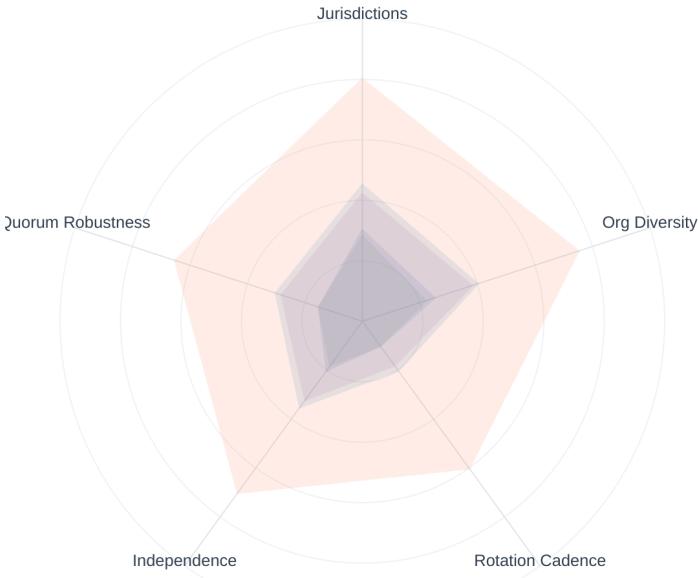


Figure: Jurisdictional and organizational diversity of keyholders across multiple countries and organizations prevents single-point coercion.

### Principle 5: Collective Choice Arrangements

- **User Sovereignty:** Users compose their own stack via configuration
- **Module Adoption:** Adoption metrics function as voting with configuration
- **Affected Parties:** Participation through choices, not committees
- **Implementation:** GUI-based module selection enables collective choice through user preferences

### Principle 6: Graduated Sanctions

- **Proportional Responses:** Minor criticism leads to moderate fork, then major depreciation, then governance fork
- **Economic Leverage:** Graduated economic pressure through merge mining revenue allocation
- **No Consensus Changes Required:** Sanctions operate at module/economic layer
- **Implementation:** Multisig thresholds vary by change category (2-of-3 extension to 6-of-7 constitutional)

## Principle 7: Monitoring and Accountability

- **Cryptographic Signatures:** All governance actions cryptographically signed and verifiable
- **Transparent Metrics:** Module adoption, revenue flows, and decision provenance all auditable
- **Three-Layer Verification:** GitHub (merge control), Nostr (real-time transparency), OpenTimestamps (immutable proof)
- **Implementation:** Automated monitoring through cryptographic verification, not social trust

### Audit Trail Completeness Map

Phase 3 estimates. Cells indicate availability of verifiable artifacts across implementations.

	Bitcoin Core	Bitcoin Knots	btcid	Libbitcoin	Bitcoin Commons
Release Signatures	Complete	Complete	Partial	Partial	Complete
Deterministic Builds Proof	Partial	Partial	Partial	Partial	Complete
Review Logs (linked to PRs)	Partial	Partial	Partial	Partial	Complete
CI Verification Proofs	Partial	Partial	Partial	Partial	Complete
OpenTimestamps / Timestamps	Partial	Partial	Partial	Partial	Complete
Merkle Proofs (Commit Sets)	No	No	No	No	Complete

Figure: Audit-trail completeness across governance layers ensures all decisions are fully evidenced and verifiable.

### Comparison with Bitcoin Core:

Bitcoin Core has informal beginnings of these principles but lacks systematic enforcement:

- Clear Boundaries: Informal (maintainers, BIP editors), no formal selection/removal process
- Consequences: Social pressure only, no systematic enforcement
- Dispute Resolution: BIP process advisory only, no binding mechanism
- External Protection: No systematic protection, individuals can be pressured
- Collective Choice: BIP process exists but no formal consensus mechanism
- Graduated Sanctions: Informal social pressure, no systematic escalation
- Monitoring: Public GitHub/mailing lists, no formal accountability system

The pattern: Bitcoin Core has informal implementations that worked at billion-scale but become vulnerable at multi-trillion scale. Commons implements all seven principles through technical architecture and cryptographic enforcement.

## 4. Technical Solution: The Orange Paper

### Problem

Bitcoin's consensus rules are embedded in code, not mathematically specified. This makes them impossible to verify, understand, or implement independently. The 2018 inflation bug (CVE-2018-17144) existed in Bitcoin Core for years before discovery. This is exactly the class of error formal verification eliminates.

### Solution

The Orange Paper provides a formal mathematical specification of Bitcoin's consensus protocol through AI-assisted extraction from Bitcoin Core's codebase. The specification includes:

- Mathematical foundations (set theory, cryptographic primitives, network protocols)
- State transition functions (block validation, transaction validation, consensus rules)
- Economic model (mining rewards, fee calculations, difficulty adjustment)
- Security properties (Byzantine fault tolerance, Sybil resistance, double-spending prevention)

### Benefits

- **Safe alternative implementations:** Independent implementations can verify against mathematical specification
- **Formal verification:** Consensus correctness can be mathematically proven
- **Reduced consensus bugs:** Systematic analysis eliminates entire classes of errors
- **Technical moat:** AI extraction eliminates "not invented here" bias

### AI-Assisted Extraction Methodology

The Orange Paper uses AI-assisted extraction from Bitcoin Core's codebase to formalize consensus rules. This approach:

- Analyzes 350,000+ lines of C++ to identify consensus-critical code paths
- Extracts mathematical relationships from implementation details
- Creates formal specifications that are independent of specific code structure
- Enables verification that specification matches actual network behavior

### Proof Maintenance and Specification Quality

The formal verification process includes ongoing maintenance to ensure specification accuracy:

## Spec Maintenance Workflow

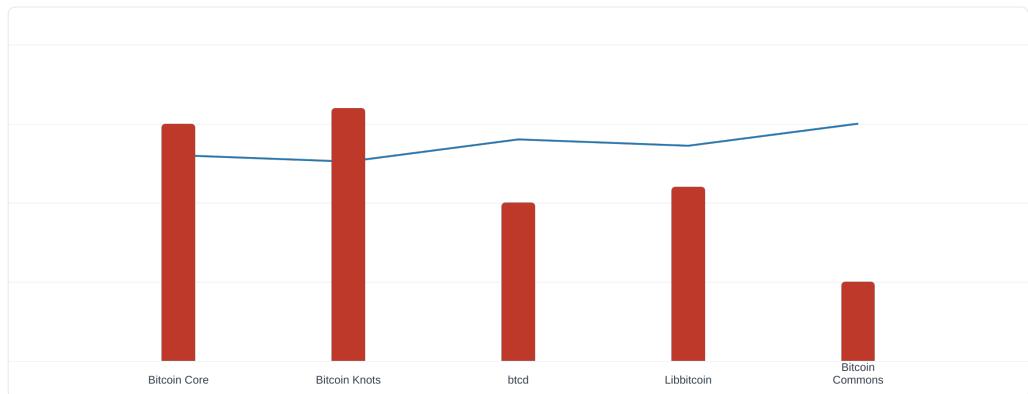
How the Orange Paper stays synchronized with code changes



*Figure: Spec maintenance workflow showing how specification stays synchronized with implementation changes through automated testing and formal verification.*

### Spec Drift vs Test Coverage

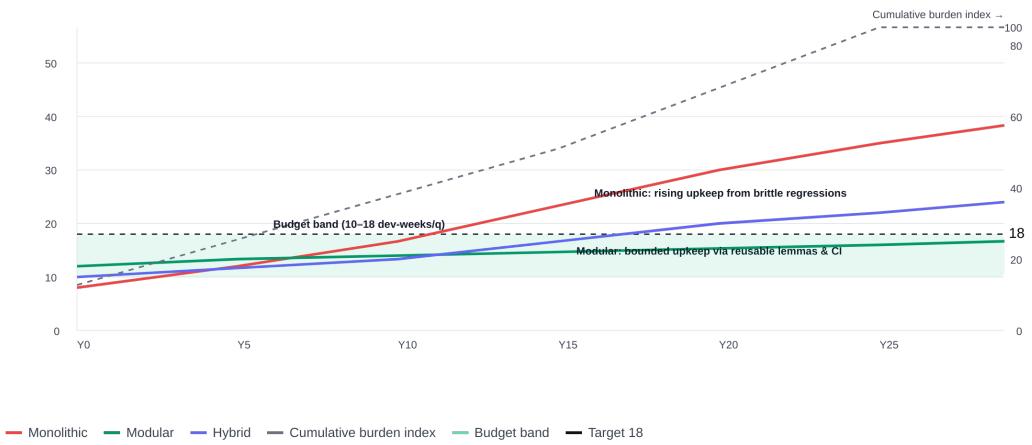
Phase 3 estimates. Bars: spec drift (lower is better). Line: test coverage (higher is better).



*Figure: Spec drift decreases as test coverage increases. Higher test coverage ensures specification accuracy over time.*

## Proof Maintenance Cost

Developer-weeks per quarter spent on proofs upkeep



— Monolithic — Modular — Hybrid — Cumulative burden index — Budget band — Target 18  
Illustrative; convert to empirical once proof inventory + CI metrics are available.

*Figure: Proof maintenance cost: proofs updated per change by area. Highlights refactor hotspots. Commons aims for lower proof churn than Core by maintaining cleaner architecture.*

**Status:** Complete specification available at <https://github.com/BTCDecoded/the-orange-paper>. The specification is actively maintained and verified against network behavior through automated testing.

## 5. Architectural Solution: Modular Governance

### Three-Layer Stack

The modular architecture consists of three layers that transform governance conflicts from political battles into architectural choices:

#### Layer 1: Mandatory Consensus (Base Node)

- Bitcoin's consensus rules, unchangeable without network agreement
- Cryptographically enforced, defines what "Bitcoin" means
- Examples: block validation, transaction validation, fork choice rules

#### Layer 2: Optional Modules (Extension System)

- User-controlled features that can be enabled or disabled
- Communities can fork/modify/compete, user choice determines winners

- Examples: Lightning Network, merge mining, Taproot Assets, privacy enhancements

### Layer 3: Economic Coordination (Revenue Model)

- Self-sustaining development through merge mining revenue
- 1% fee on merged chain rewards, scales with adoption
- Revenue allocation: 60% core / 25% modules / 10% audits / 5% ops

### Module Isolation

Modules run in separate processes with strict boundaries:

- **Process isolation:** Each module runs in its own process space
- **Memory boundaries:** Modules cannot access base node memory directly
- **Communication protocols:** Well-defined APIs for module interaction
- **Crash containment:** Module failures cannot affect consensus

**What modules CANNOT do:** Modify consensus rules, alter block validation, cause network splits

**What modules CAN do:** Process their own state, crash without affecting base node

### Containment Strategy

The modular architecture satisfies both camps simultaneously:

- **“Don’t Change Bitcoin” Camp:** Gets pure Bitcoin base layer with no modifications
- **“Make Bitcoin Useful” Camp:** Gets optional features through modules
- **Miners:** Get additional revenue from merge mining

**The Module System IS The Governance System:** Instead of governing through committees deciding features, we govern through architecture enabling choice. The module system isn't just technical: it's the governance mechanism itself, implementing Ostrom's collective choice arrangements through user configuration, Hayek's competitive discovery through module competition, and Bitcoin's permissionless innovation through fork-ability.

# Architecture Diagrams

## Tiered Architecture

From mathematical foundation to governance infrastructure



Figure: Tiered architecture: Tier 1 = Orange Paper + Consensus Proof (mathematical foundation); Tier 2 = Protocol Engine (protocol abstraction); Tier 3 = Reference Node (complete implementation); Tier 4 = Developer SDK + Governance (governance infrastructure).

## How the Stack Works in Practice

End-to-end path for a newly received block



Figure: End-to-end data flow through Reference Node, Consensus Proof, Protocol Engine, modules, and governance. Clear separation of concerns: each tier depends only on the layers below; modules cannot affect consensus.

## Module Isolation Technical Details

**Process Isolation Mechanisms:** - Each module runs in its own process space with isolated memory - Modules communicate only through well-defined APIs - Base node validates all blocks using Orange Paper specification regardless of enabled modules - Module state completely separate from consensus state (UTXO set)

**API Boundaries:** - Modules can only interact with base layer through documented interfaces - No direct access to consensus functions or core data structures - Module failures isolated and cannot propagate to base node - Crash containment guaranteed by process boundaries

## Module Quality Control Process

Quality gates ensure module ecosystem safety and standards



Figure: Module quality control process ensuring security, performance, and community validation before module adoption.

## Fragmentation Analysis:

### Fragmentation Analysis Comparison

Consensus forks (network splits) vs Governance forks (no network splits)

Consensus Forks	Governance Forks
<b>Network Splits</b> Creates separate blockchains with incompatible consensus rules. Users must choose which chain to follow.	<b>No Network Splits</b> Same consensus rules, different governance structures. All nodes remain on same blockchain.
<b>Examples</b> Bitcoin Cash (2017), Bitcoin SV (2018). Results in permanent network division.	<b>Examples</b> Bitcoin Knots adoption (2025). Users switch implementations without network split.
<b>Impact</b> Fragments network, reduces security, creates confusion, splits community.	<b>Impact</b> Enables governance diversity while maintaining consensus unity. No fragmentation.
<b>Risk Level</b> <b>HIGH: Permanent network fragmentation</b>	<b>Risk Level</b> <b>LOW: No network fragmentation</b>

Figure: Fragmentation analysis showing that governance forks don't split the network. All implementations validate same Bitcoin consensus while enabling governance competition.

Governance forks preserve the consensus layer while allowing governance changes. Users can fork governance rules while keeping the same Bitcoin consensus. This is the ultimate accountability mechanism. Knots adoption (25% in five months) proved multiple implementations coexist without fragmentation.

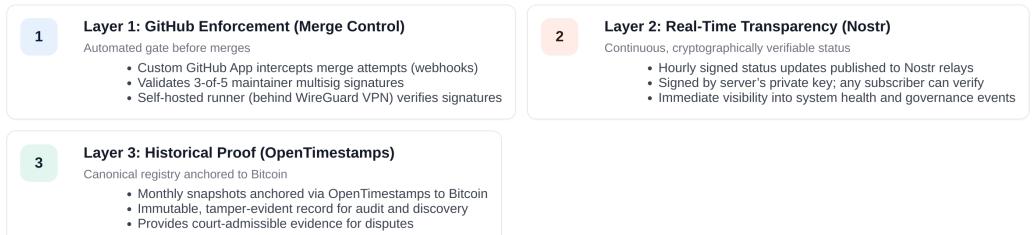
## 6. Cryptographic Governance Enforcement

### Three-Layer Verification

The governance system implements three complementary verification layers that work together to ensure both real-time transparency and immutable historical proof:

## Three-Layer Verification Architecture

Real-time transparency + immutable historical proof with automated enforcement



Nostr = Real-time visibility

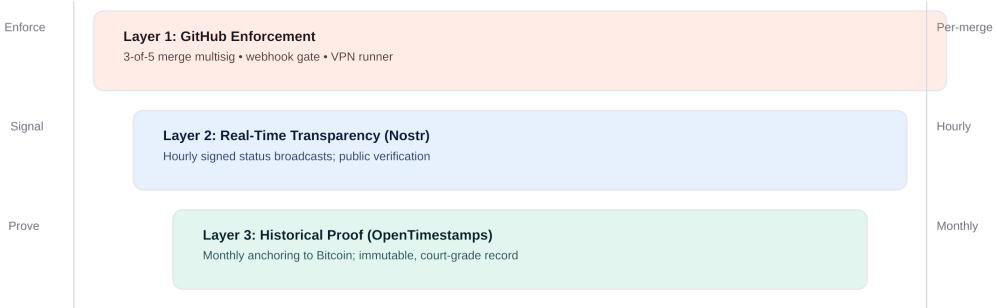
Continuous, signed telemetry for operational awareness

OpenTimestamps = Immutable proof

Periodically anchored state with durable legal provenance

**Why both?** Real-time signals catch issues quickly (Nostr), while periodic anchoring guarantees an immutable audit trail (OpenTimestamps). Together they provide live assurance and historical finality.

*Figure: Overview of the three-layer verification approach showing how GitHub, Nostr, and OpenTimestamps work together.*



*Figure: Three-layer verification: GitHub merge control, real-time Nostr transparency, and OpenTimestamps historical proof.*

**Layer 1: GitHub Enforcement (Merge Control)** - Custom GitHub App validates multisig requirements (varies by layer: 2-of-3 to 6-of-7) - Self-hosted runner behind WireGuard VPN validates signatures using secp256k1 - Even repository admins cannot bypass cryptographic requirements - Signature validation happens before merge approval

**Layer 2: Real-Time Transparency (Nostr)** - Hourly status updates published to Nostr relays - Status includes: binary hash, config hash, recent merges, health metrics - Cryptographically signed by server's unique NPUB (Nostr public key) - Anyone can subscribe and verify server integrity in real-time - Missing updates trigger community alerts within 2 hours

**Layer 3: Immutable Proof (OpenTimestamps)** - Monthly canonical registry anchored to Bitcoin blockchain - Critical events (key rotations, deployments) timestamped immediately - Creates cryptographic proof of governance state at specific block height - Provides court-admissible evidence for dispute resolution - Works independently of any single server or relay

**Cross-Layer Verification:** Three independent layers verify governance actions and each other. Risk at one layer does not compromise the others. This defense-in-depth approach ensures governance integrity even if one verification method is compromised.

## Repository Hierarchy

Different signature thresholds based on risk level: - **Constitutional (Orange Paper):** 6-of-7 maintainer signatures, 180-day review - **Implementation:** 4-of-5 signatures, 90-day review - **Application:** 3-of-5 signatures, 30-day review - **Extension:** 2-of-3 signatures, 7-day review

## Emergency Response

Emergency situations require higher signature thresholds (4-of-5, 5-of-5) and extended time windows based on risk level, with automatic expiration to prevent permanent emergency powers. The tiered system escalates requirements proportionally to the severity of the situation while maintaining governance integrity.

## Security Architecture: Push-Only Design

### Security Architecture Details:

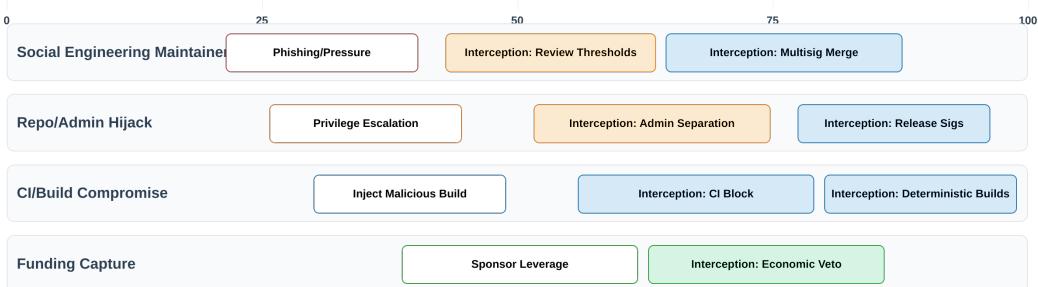
- **No HTTP Endpoints:** Governance servers have no incoming HTTP endpoints (minimal exposure surface)
- **VPN Isolation:** Servers communicate outbound only through WireGuard VPN
- **Self-Hosted Runner:** GitHub runner behind WireGuard VPN for signature validation
- **Data Flow:** Server to GitHub (push) to Nostr (publish) to Bitcoin (anchor)
- **Public Read Access:** GitHub repo, Nostr relays, Bitcoin blockchain (read-only for public)

### Attack Path Protection:

## Attack Path Interception Map

Phase 3 estimates. Timeline shows attempt stages (0–100) and where Bitcoin Commons intercepts.

Attempt    Interception (Process)    Interception (Technical)    Interception (Economic)

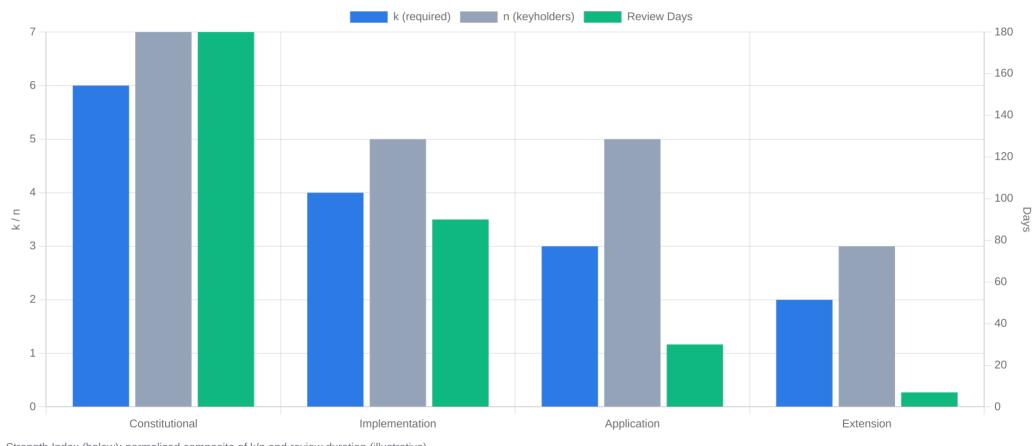


*Figure: Risk interception points across GitHub, Nostr, and OpenTimestamps. Three independent layers verify governance actions and each other.*

## Multisig Threshold Details

### Governance Signature Thresholds by Risk Class

k-of-n thresholds and review windows



Strength Index (below): normalized composite of  $k/n$  and review duration (illustrative)

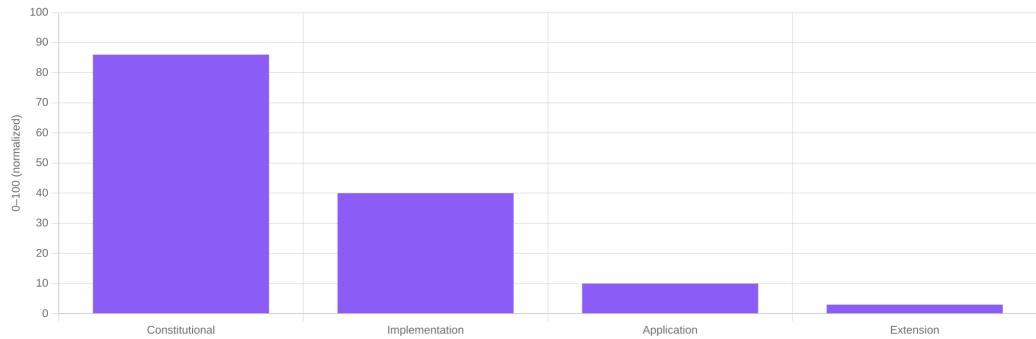
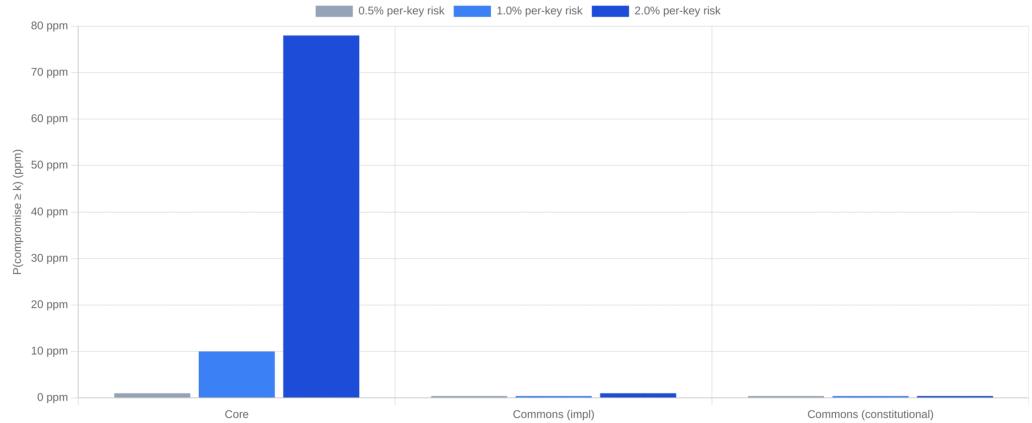


Figure: Governance signature thresholds by change category (constitutional, implementation, application, extension).

## Multisig Threshold Sensitivity

Policy comparison:  $P(\text{compromise} \geq k)$  per assumed per-key risk  $p$  (lower is better)



Risk = chance attackers can reach required signatures ( $\geq k$ ) given per-key compromise probability  $p$ . Assumes independent keys; dispersion reduces effective  $p$ . Policies shown: Core ( $k=3, n=5$ ), Commons-impl ( $k=4, n=5$ ), Commons-constitutional ( $k=6, n=7$ ).

*Figure: Multisig threshold sensitivity: false negative and false positive risk vs threshold. Commons balances safety and throughput through carefully calibrated thresholds.*

**Explicit Thresholds by Layer:** - **Constitutional (Orange Paper):** 6-of-7 maintainer signatures, 180-day review period - **Consensus Changes:** 5-of-5 signatures, 365-day review period (longest review) - **Implementation:** 4-of-5 signatures, 90-day review - **Application:** 3-of-5 signatures, 30-day review - **Extension:** 2-of-3 signatures, 7-day review

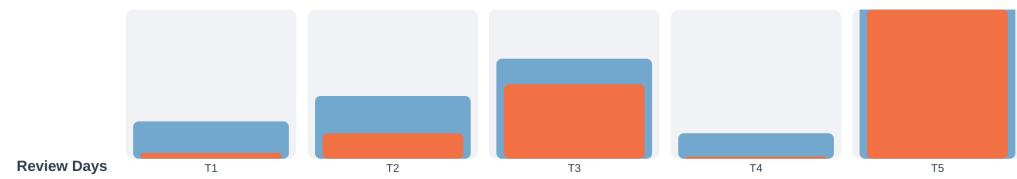
All signatures verified using secp256k1 (same curve as Bitcoin). GitHub App validates signatures before allowing merges. Even repository admins cannot bypass cryptographic requirements.

## Governance Process and Latency

### Governance Process Latency by Tier

Phase 3 estimates. Core: informal ranges; Commons: guaranteed timing. Lower = faster.

■ Core/Knots (informal) ■ Bitcoin Commons (formalized)

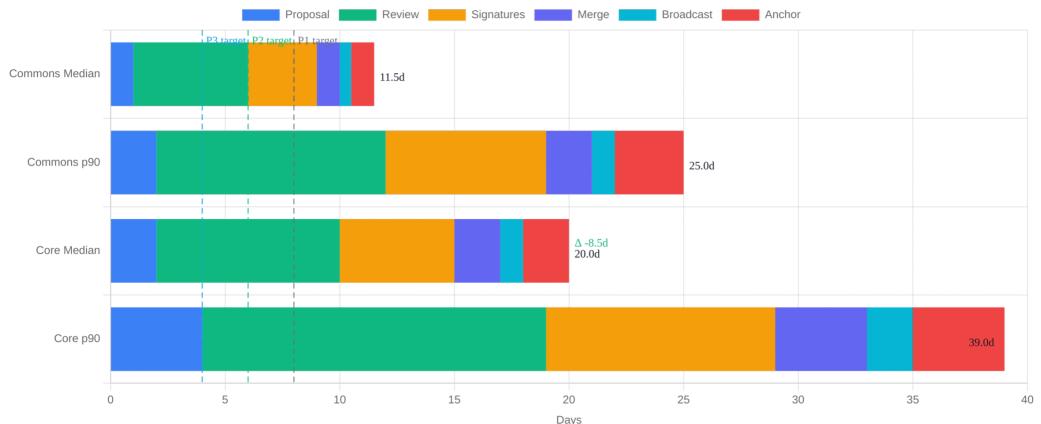


Tiers: T1 Routine, T2 Features, T3 Consensus-Adjacent, T4 Emergency, T5 Governance.

*Figure: Governance process latency and escalation tiers. Stages map to proposal → review → approvals → merge.*

## Governance Latency Stack

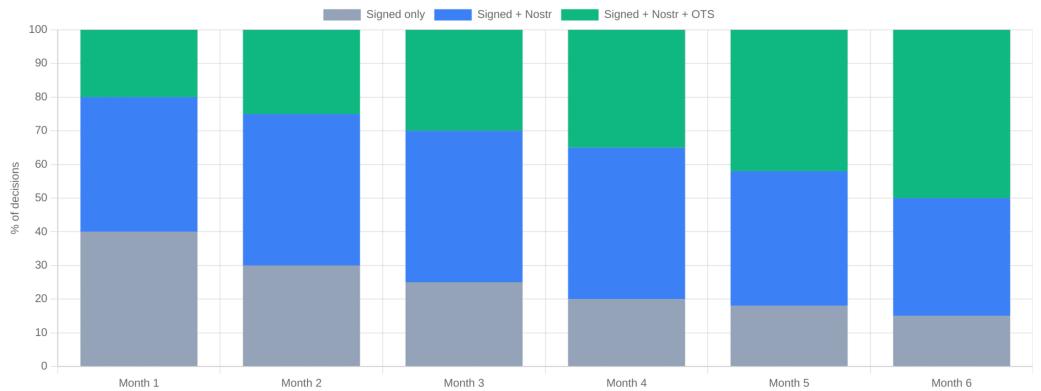
Proposal → Review → Signatures → Merge → Broadcast → Anchor (Median vs p90; phase targets)



*Figure: Governance latency: time by stage. Reduced queueing at gates through automation and process optimization.*

## Decision Provenance Completeness

100% stacked by month: Signed only / Signed+Nostr / Signed+Nostr+OTS



*Figure: Decision provenance: share of fully evidenced decisions across layers. Three-layer verification ensures complete audit trails.*

# Release Pipeline Gate Strength

Phase 3 estimates. Wider = stronger enforcement at that gate.

■ Commons strength (outer bar) · ■ Core coverage (inner bar)

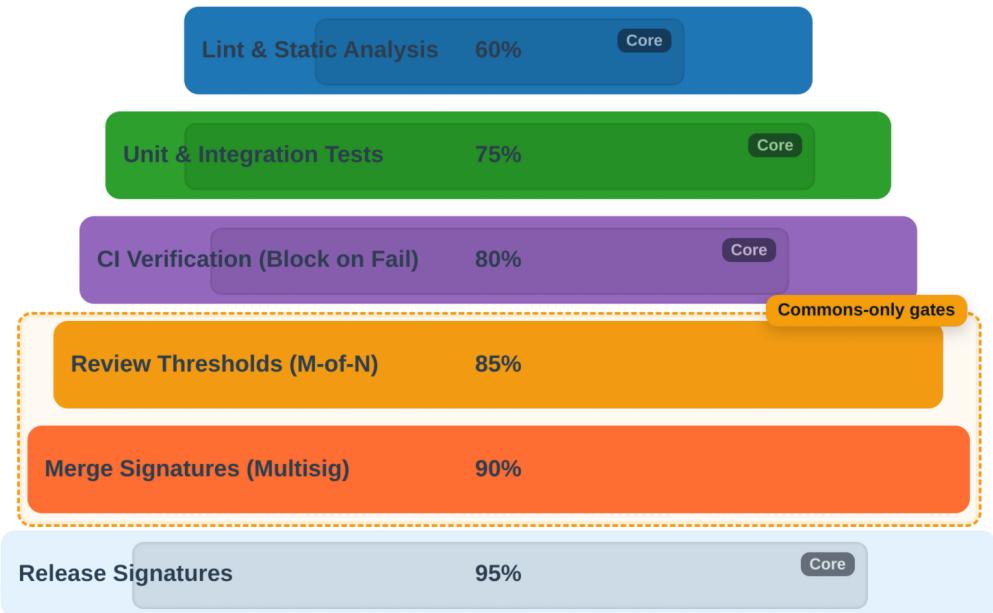


Figure: Gate strength across the release pipeline. Each gate enforces appropriate signature requirements and review periods.

## PR Review Time Distribution

Median grows, outliers become extreme (contributor frustration)

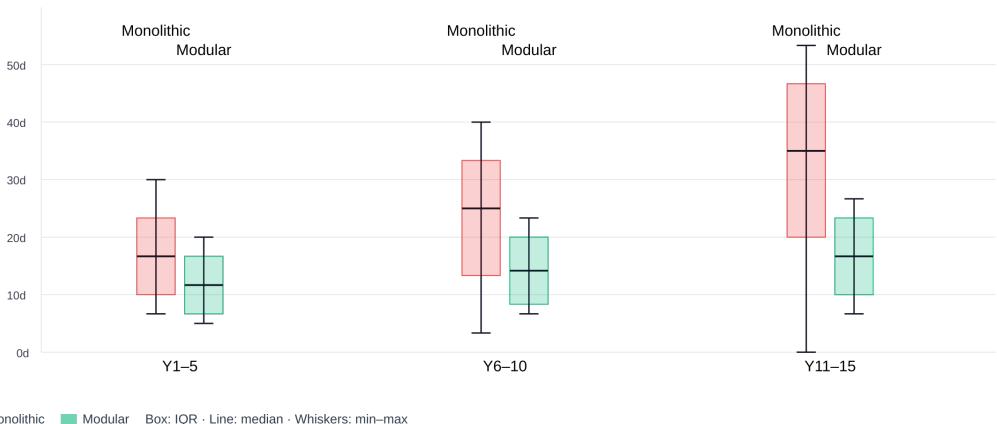


Figure: Pull request review time distribution. Long tails reveal why throughput stalls without process and tooling. Automated validation reduces review bottlenecks.

## 7. Economic Sustainability

### Merge Mining Model

Merge mining allows miners to mine multiple chains simultaneously. When mining Bitcoin, they can also mine secondary chains (RSK, DATUM, Namecoin) without additional computational work. Secondary chain rewards flow through Commons infrastructure, with 1% fee funding development.

### Revenue Allocation

- **60% Core Development:** Base node and critical modules
- **25% Module Developer Grants:** Incentivizes quality modules
- **10% Security Audits:** Ensures quality and safety
- **5% Operations:** Infrastructure and maintenance

### Self-Sustaining Benefits

- No reliance on donations, grants, or VC funding
- Revenue scales with actual usage and miner adoption
- Economic leverage enables rule enforcement without consensus changes
- Miner alignment creates supporting constituency

### Stratum V2 Merge Mining Coordination

Merge mining coordination uses Stratum V2, a modern protocol that aligns with Commons governance principles:  
- **Miners Control Transaction Selection:** Job negotiation decentralizes power  
- **Encrypted Communication:** Reduces risk of hashrate hijacking  
- **Efficient Binary Protocol:** Reduces bandwidth by roughly 50-66 percent  
- **Multiplexed Channels:** Enable merge mining coordination naturally

### Revenue Scaling Examples

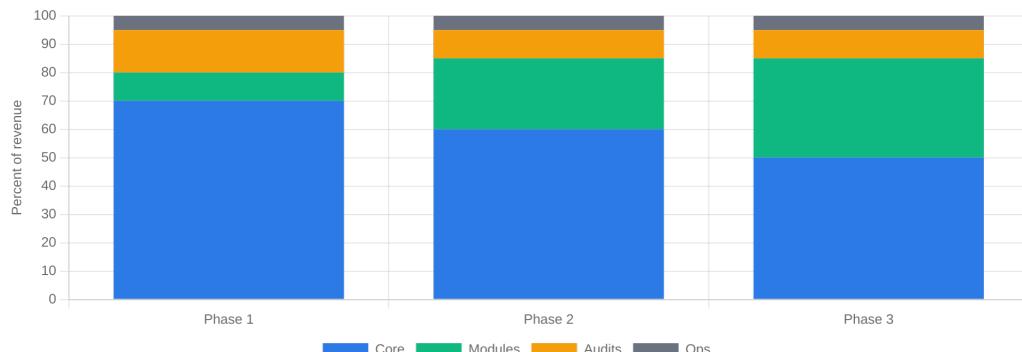
**Calculations:** - If merge-minable chains generate 100 BTC/year in rewards, 1% fee yields ~1 BTC/year for development - At 10 merged coins: ~10 BTC/year revenue - At 100 coins: ~100 BTC/year revenue - Revenue scales with adoption without requiring user payments

**Infrastructure Costs:** - Server costs: \$75-200/month for servers, VPN, and tooling - Annual costs: under ~\$30K including security audits - Model aims to protect substantial Bitcoin value at low overhead

# Economic Model Charts

## Revenue Allocation Breakdown

Allocation of Commons fee revenue (must sum to 100%)



Policy guardrails (illustrative): Core 50–70%, Modules 20–30%, Audits 10–15%, Ops 5–10%.

**Figure: How funds are allocated across core development (60%), modules (25%), audits (10%), and operations (5%).**



**Figure: Why incentives align for miners, developers, and users. Merge mining revenue creates supporting constituency.**

## Economic Alignment



Baseline markers show typical alignment in Core (lower) and Corporate OSS (varies).

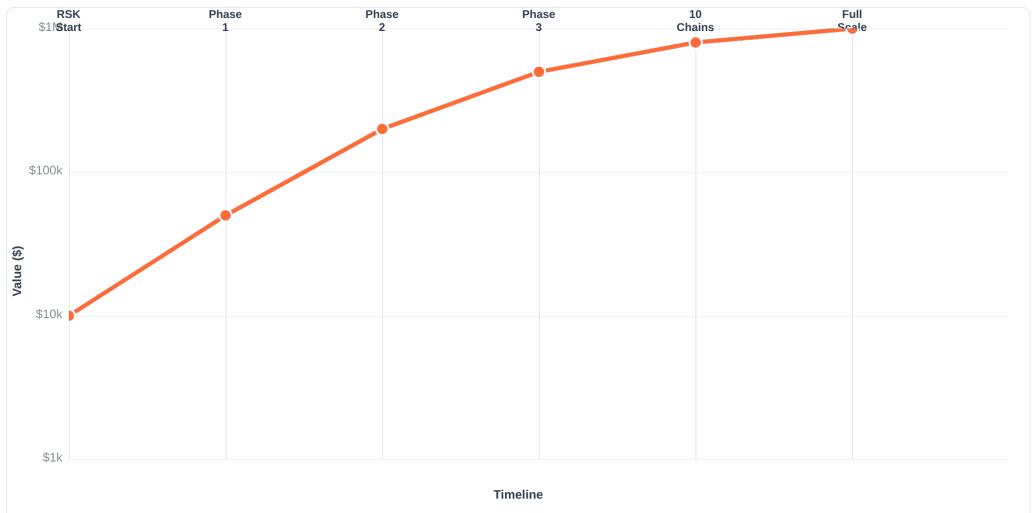
**Figure: Economic alignment showing incentives for miners, developers, and users via merge mining revenue and grants.**



**Figure: Funding model comparison: Core's donation-dependent model vs Commons' self-sustaining merge mining revenue that scales with usage.**

## Economic Scaling Trajectory

\$10k RSK → \$1M/10 Chains - Projected economic value generation across deployment phases.



**Figure: Economic scaling across development phases. Revenue scales with adoption and miner participation.**

## Revenue Model Sensitivity Analysis

Annual revenue from merge mining (1% of rewards) and marketplace (15-30% of module sales)

	10 chains	25 chains	50 chains	100 chains
5% adoption	\$75k 0.015 BTC	\$190k 0.04 BTC	\$375k 0.075 BTC	\$750k 0.15 BTC
10% adoption	\$150k 0.03 BTC	\$375k 0.075 BTC	\$750k 0.15 BTC	\$1.5M 0.30 BTC
20% adoption	\$300k 0.06 BTC	\$750k 0.15 BTC	\$1.5M 0.30 BTC	\$3M 0.60 BTC
50% adoption	\$750k 0.15 BTC	\$1.9M 0.38 BTC	\$3.75M 0.75 BTC	\$7.5M 1.50 BTC

Figure: Revenue model sensitivity analysis showing how revenue scales with chains adopting Commons and Commons adoption (network effects).

## Secondary Chain Value Proposition Comparison

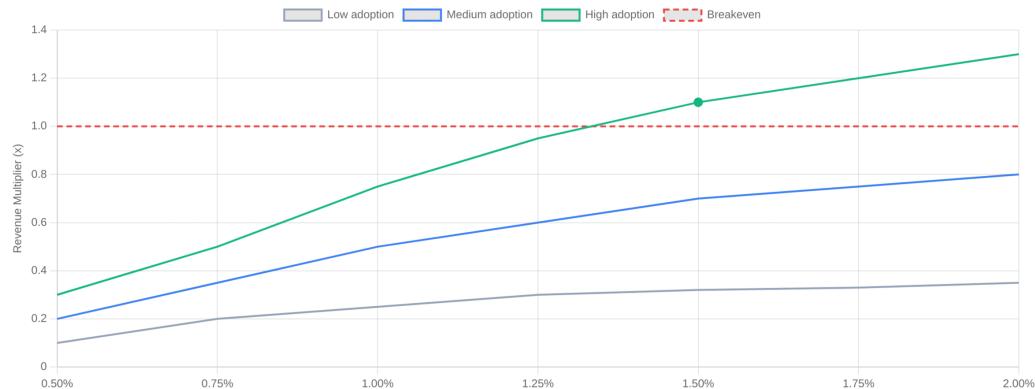
Commons vs Existing Providers vs Custom Infrastructure

Commons	Existing Providers	Custom Infrastructure
Integration Cost <b>Low</b>	Integration Cost <b>Medium</b>	Integration Cost <b>Very High</b>
Access to Hash Power <b>High</b>	Access to Hash Power <b>Medium</b>	Access to Hash Power <b>Low</b>
Governance Transparency <b>Complete</b>	Governance Transparency <b>Low</b>	Governance Transparency <b>Variable</b>
Fees <b>1%</b>	Fees <b>2-5%</b>	Fees <b>High</b>

Figure: Secondary chain value proposition comparison. Commons offers reduced integration cost, access to Bitcoin's hash power, governance transparency, and lower fees (1% vs building infrastructure).

## Miner Economics Sensitivity

Revenue delta vs fee rate across adoption scenarios; breakeven annotated



Adoption = share of merged-reward flow captured (miner participation  $\times$  chain coverage). Breakeven occurs where a scenario line crosses the red dashed 1.0 line at a given fee rate.

*Figure: Miner sensitivity to merge-mined yields. Support persists across ranges due to direct economic incentives.*

## Sustainability Over Time

Divergent trajectories; monolithic cliff around year 12–15



*Figure: Sustainability over time: modular governance aims to sustain change while reducing capture risks compared to monolithic approaches.*

## Economic Veto Threshold Sensitivity

Phase 3 estimates. X = Veto threshold (%); Y = Capture Risk (lower is better). Band shows uncertainty.

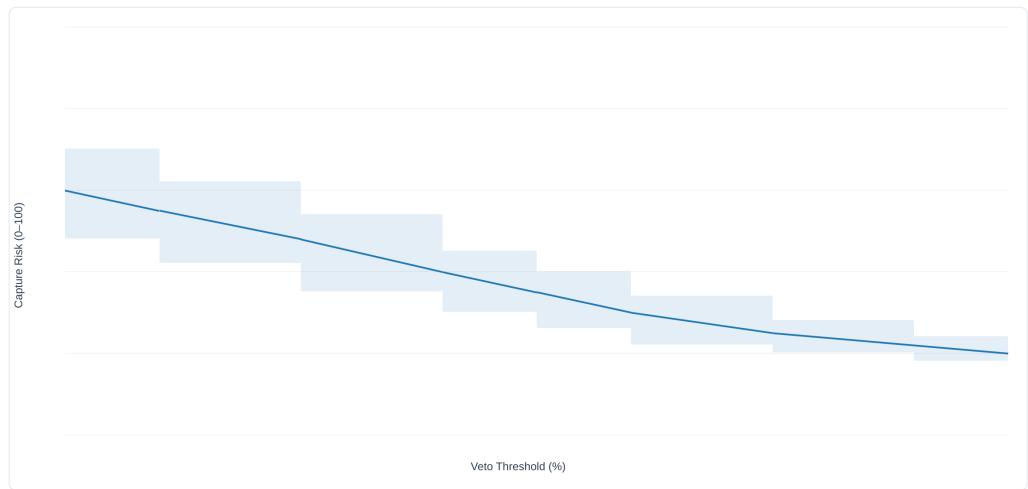


Figure: Economic veto thresholds and aligned incentives. Revenue allocation enables graduated sanctions without consensus changes.

## Why Secondary Chains Choose Commons

Secondary chains need merge mining infrastructure. Commons value proposition:

- **Reduced Integration Cost:** 1% fee cheaper than building infrastructure
- **Access to Bitcoin's Hash Power:** Leverage Bitcoin network effects
- **Governance Transparency:** Cryptographic audit trails
- **Proven Infrastructure:** Lower risk than building from scratch

**Target Adoption Strategy:** Target existing merge-mined chains (RSK, Namecoin, DATUM) with migration tools. Demonstrate economic benefits: reduced costs, improved governance, better security.

**Fallback if Secondary Chains Don't Adopt:** Phase 1 can proceed without full revenue. Alternatives include module fees, grants, donations. Long-term network effects accelerate adoption.

## Success Metrics

- **Level 1 (Sustainability):** 1000+ nodes, 20+ miners, revenue-positive operation
- **Level 2 (Ecosystem Health):** 3+ implementations with >15% combined node share

Success Level 1 proves sustainability. Success Level 2 proves the mission: implementation diversity becomes normal. We succeed when others copy the approach, not when we dominate the market.

## 8. Failure Modes & Mitigations

### Governance Capture

**Risk:** Keyholder collusion or compromise **Mitigation:** Multi-jurisdictional keyholders, transparent operation, fork-ready design **Comparison:** Current system easier to capture (target individuals privately, invisible control)

### Regulatory Pressure

**Risk:** Authorities pressure keyholders to implement backdoors **Mitigation:** Distributed keyholders across jurisdictions, visible capture attempts, modular containment **Protection:** No single jurisdiction can compel 3-of-5 threshold

### Technical Risks

**Risk:** Module consensus bugs, complexity explosion **Mitigation:** Module isolation, formal verification, security audits **Guarantee:** Module failures cannot affect consensus

### Social Risks

**Risk:** Community rejection, fork wars, reputation attacks **Mitigation:** Not asking permission, let code speak, coalition provides proof **Strategy:** Focus on substance, build alternatives, let market decide

### Ultimate Protection

Governance forks provide the ultimate accountability mechanism: users can fork governance rules while keeping the same Bitcoin consensus, preserving network unity while enabling governance competition.

## 9. Implementation Status

### Seven Repositories

All repositories are public and active at <https://github.com/BTCDecoded>:

1. **Orange Paper:** Mathematical specification of Bitcoin consensus
2. **Protocol Engine:** Core protocol logic and state management
3. **Consensus Proof:** Formal verification of consensus rules
4. **Reference Node:** Complete Bitcoin implementation
5. **Developer SDK:** Governance primitives and composition framework
6. **Governance:** Configuration repository for governance rules
7. **Governance App:** GitHub App that enforces governance rules

## Current State

Phase 1 infrastructure provides substantial code implementing core capabilities. The system includes mathematical foundation and clean architecture. Governance infrastructure enables cryptographic enforcement.

## Development Roadmap

### Bitcoin Commons Development Trajectory

Phase 1 → Phase 2 → Phase 3 progression (illustrative).



Figure: Development trajectory across phases showing progression from foundation to maturity.

### Upgrade Safety Checklist Coverage

Phase 3 estimates. Deploy safety controls across implementations.

	Bitcoin Core	Bitcoin Knots	btcd	Libbitcoin	Bitcoin Commons
Rollback Plan Tested	Partial	Partial	Partial	Partial	Complete
Canary Deploy / Staged Rollout	Partial	Partial	Partial	Partial	Complete
Deterministic Builds	Partial	Partial	Partial	Partial	Complete
Signed Releases	Complete	Complete	Partial	Partial	Complete
CI Gates Block on Failure	Partial	Partial	Partial	Partial	Complete
Audit Log Complete	Partial	Partial	Partial	Partial	Complete

Figure: Upgrade safety checklist before activation. Prerequisites must be met before governance enforcement begins.

Phase 1 (Infrastructure Building) code exists in working form. The roadmap below describes Phase 2 (Governance Activation) and Phase 3 (Full Operation). Phase 2 activation is milestone-based: prerequisites must be met before governance enforcement begins. Timeline depends on completing these requirements, not arbitrary deadlines.

Success metrics operate on two levels: Level 1 (project sustainability) and Level 2 (ecosystem health through diverse implementations). The goal isn't to replace Bitcoin Core, it's to create the foundation that makes competing implementations possible.

### Phase 1: Foundation

Phase 1 infrastructure includes substantial code providing core capabilities:

#### Seven Repositories:

1. **Orange Paper:** Mathematical specification of Bitcoin consensus

2. **Protocol Engine:** Core protocol logic and state management
3. **Consensus Proof:** Formal verification infrastructure
4. **Reference Node:** Complete Bitcoin implementation
5. **Developer SDK:** Governance primitives and composition framework
6. **Governance:** Configuration repository for governance rules
7. **Governance App:** GitHub App that enforces governance rules

**Current Capabilities:** - Mathematical foundation (Orange Paper) - Clean architecture (Protocol Engine) - Cryptographic enforcement (Governance App) - Network compatibility testing - Module system architecture designed

## Phase 2: Governance Activation

**Prerequisites (Must be met before activation):** - Comprehensive security audit by independent firm - Public community validation period completed - Production key management procedures operational - Formal verification of critical consensus paths complete - Legal review across multiple jurisdictions - Miner commitment threshold reached (at least one major miner) - No critical issues outstanding from Phase 1 review

### Phase 2 Milestones:

**Working Base Node:** - Complete Reference Node implementation - Ensure network compatibility with Bitcoin mainnet, testnet, and regtest - Milestone: At least one major miner committed to merge mining model

**Module System Architecture:** - Design and implement module API - Create module loading and management system - Build basic module examples - Milestone: Module system is functional and documented

**Cryptographic Governance:** - Implement multisig signing infrastructure - Set up distributed keyholder system - Create transparent decision-making processes - Deploy Governance App for automated enforcement - Milestone: Governance system is operational with full three-layer verification

**Lightning Integration Module:** - Build Lightning Network integration module - Demonstrate module system capabilities - Show how conflicts can be resolved through architecture - Milestone: Lightning module is working and adopted

**Merge Mining Support:** - Implement merge mining infrastructure - Target RSK, DATUM, Namecoin as initial secondary chains - Set up revenue collection system - Milestone: First revenue collection from merge mining fees

**Module Marketplace:** - Build infrastructure for module distribution - Create quality control and security audit processes - Implement module adoption metrics - Milestone: Module marketplace is operational

**Revenue-Positive Operation:** - Achieve sustainable funding through merge mining - Demonstrate economic model viability - Show self-sustaining development - Milestone: 1000+ node operators, revenue-positive operation (Level 1 success)

## Sustainability & Ecosystem Health

Targets: L1 Sustainability and L2 Ecosystem Health

### Phase 1 — Foundation

Targets focus on proving infrastructure readiness and first revenue.



### Phase 2 — Expansion

Targets emphasize nodes, miner adoption, and ecosystem activation.



### Phase 3 — Maturity

Targets aim at self-sustainability and ecosystem health.



*Figure: Sustainability and ecosystem health indicators across phases. Tracks node adoption, miner participation, and revenue generation.*

## Phase 3: Maturity

**Advanced Modules:** - Build privacy enhancement modules - Create alternative mempool policy modules - Develop smart contract integration modules - Milestone: 50+ available modules

**Interoperability:** - Fedimint integration demonstrating infrastructure positioning - Shared Iroh networking and LDK Lightning components enable natural interoperability - Commons as infrastructure layer enabling other projects

**Self-Sustaining Development:** - Achieve complete independence from external funding - Demonstrate sustainable economic model - Show governance system can operate without founder - Milestone: Self-sustaining without external funding

**Economic Leverage:** - Demonstrate economic leverage over contained ecosystems - Show how rules can be enforced through economic pressure - Prove governance system effectiveness - Economic leverage over secondary chain ecosystem

**Production Deployment:** - Full mainnet governance infrastructure - First multisig merge completed - First OpenTimestamps anchor created - Public monitoring operational - Key rotation completed - Milestone: 10,000+ node operators, recognized as viable alternative

**Recognition as Viable Alternative:** - Gain recognition from Bitcoin community - Demonstrate technical superiority - Show governance advantages - Milestone: Accepted as legitimate Bitcoin implementation

## Phase 4: Ecosystem Normalization

**Reference Implementation:** - Become reference implementation for modular architecture - Set standards for Bitcoin implementations - Influence Bitcoin development ecosystem - Reference for HOW to build modular Bitcoin implementations

**Multiple Implementations:** - Enable multiple implementations using Commons SDK - Show multiple implementations are possible - Demonstrate governance system scalability

**Implementation Diversity Normalized:** - Make multiple implementations normal in Bitcoin - Show Core is one option among many - Milestone: Implementation diversity normalized (Level 2 success)

**Governance Model Adoption:** - Have governance model adopted by other projects - Show governance principles are universal - Milestone: Governance model adopted by other projects

## Strategic Positioning

**Infrastructure, Not Competitor:** - Position as infrastructure for multiple implementations - Not trying to replace Core, providing foundation for alternatives - Success measured by ecosystem health, not market share

**BitMEX Precedent:** - BitMEX validated Type 3 software forks - Commons adds specification, governance, and economics

**Success When Others Build on Your Foundation:** - Measured by ecosystem adoption - Not about market share dominance - About enabling ecosystem health

## Key Metrics

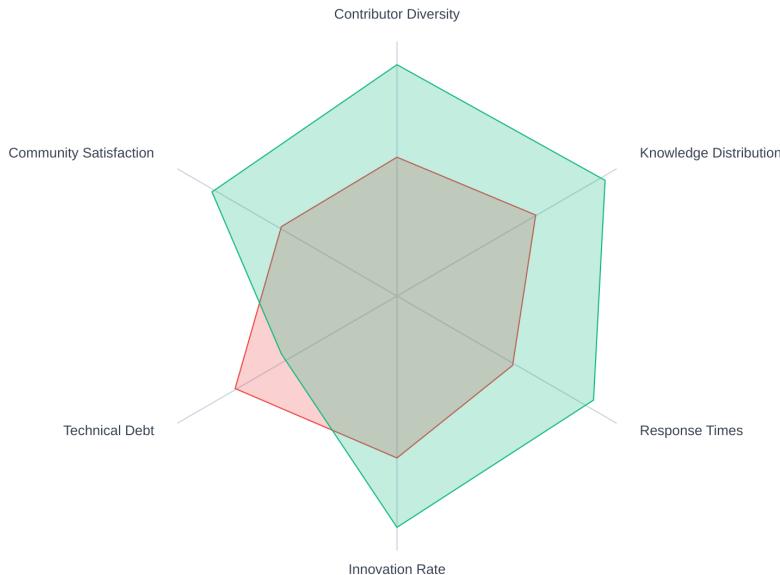
**Technical Metrics:** - Network compatibility: Full Bitcoin network compatibility - Module adoption: Number of modules in use - Revenue generation: Merge mining revenue collected - User adoption: Number of node operators

**Governance Metrics:** - Decision transparency: All decisions cryptographically signed - Economic alignment: Revenue flows are auditable - Anti-capture: Distributed keyholders, fork-ready design - Sustainability: Self-funding without external dependencies

**Ecosystem Metrics:** - Diverse implementations: Number of independent implementations - Module marketplace: Number of available modules - Developer adoption: Number of developers building on platform - Community recognition: Acceptance by Bitcoin community

Community Health Radar

Multi-dimensional health comparison



*Figure: Community health radar tracks breadth of participation, contributor retention, and review responsiveness across releases.*

These metrics measure the health of the ecosystem, not just the success of Commons itself. The goal is to create a thriving ecosystem of implementations and modules.

## Risk Mitigation

**Technical Risks:** - Consensus bugs: Formal verification and testing - Network splits: Consensus compatibility maintained - Module conflicts: Architecture prevents base layer conflicts - Maintenance burden: Economic incentives for quality

**Governance Risks:** - Capture: Distributed keyholders, transparent operation - Economic pressure: Self-funding, economic alignment - Social engineering attempts: Focus on substance, let code speak - Regulatory pressure: Multi-jurisdictional, fork-ready design

**Market Risks:** - Adoption: Focus on technical superiority and governance advantages - Competition: Build better system, let market decide - Sustainability: Economic model scales with usage - Recognition: Demonstrate value through results

## 10. Conclusion

Bitcoin's governance vacuum represents its greatest vulnerability at multi-trillion dollar scale. The technical architecture is bulletproof, but the social architecture runs on gentleman's agreements. Bitcoin Commons provides concrete, implementable solutions by synthesizing three proven frameworks: Ostrom's commons governance principles, Hayek's competitive discovery mechanisms, and Bitcoin's cryptographic enforcement tools.

This isn't speculation. It's applying battle-tested principles from economics, social science, and cryptography to governance. Each framework addresses weaknesses in the others: cryptography makes Ostrom enforceable at scale, infrastructure enables Hayek's competition, and modularity plus fork-ability creates competitive discovery.

The choice is clear: either we build these institutions or institutions get built for us. The question is not whether Bitcoin needs governance reform, but whether the community will proactively build institutions or reactively accept whatever gets imposed during the next crisis.

The choice: decentralize the builders, or watch them become kings.

# References

## Academic Sources

- Ostrom, E. (2009). *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press.
- Hayek, F. A. (1945). The Use of Knowledge in Society. *American Economic Review*, 35(4), 519-530.
- De Filippi, P., & Loveluck, B. (2016). The Invisible Politics of Bitcoin: Governance Crisis of a Decentralized Infrastructure. *Internet Policy Review*, 5(3).
- Walch, A. (2015). The Bitcoin Blockchain as Financial Market Infrastructure: A Consideration of Operational Risk. *Fordham Law Review*, 84(1), 1-58.
- Walch, A. (2017). The Path of the Blockchain Lexicon (and the Law). *Vermont Law Review*, 42(1), 1-30.
- Walch, A. (2019). Deconstructing ‘Decentralization’: Exploring the Core Claim of Crypto Systems. *Cryptoassets: Legal, Regulatory, and Monetary Perspectives*, 55-78.

## Historical Sources

- Andresen, G. (2014). Bitcoin: The Future of Money? Princeton University, March 27, 2014.
- Hearn, M. (2016). The Resolution of the Bitcoin Experiment. *Medium*, January 15, 2016.
- BitMEX Research (2020). Bitcoin Core’s Competition. *BitMEX Research*, January 2020.

## Technical Sources

- CVE-2018-17144 (2018). Bitcoin Core Denial of Service Vulnerability. *CVE Details*. <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-17144>
- Bitcoin Core Statistics. Public GitHub repository data.
- Bitcoin Optech Topics: High quality technical primers and references
  - Merged mining: <https://bitcoinops.org/en/topics/merged-mining/>
  - Stratum v2: <https://bitcoinops.org/en/topics/stratum-v2/>
- BOLT (Lightning) Specifications: <https://github.com/lightning/bolts>

## Repository Links

- <https://github.com/BTCDecoded/the-orange-paper>
- <https://github.com/BTCDecoded/protocol-engine>
- <https://github.com/BTCDecoded/consensus-proof>
- <https://github.com/BTCDecoded/reference-node>
- <https://github.com/BTCDecoded/developer-sdk>
- <https://github.com/BTCDecoded/governance>

- <https://github.com/BTCDecoded/governance-app>