Who Governs Bitcoin's Code, and Why Diligence Doesn't Ask

Secure Sovereign

Published October 30, 2026

Disclosure. The author builds Bitcoin Commons, an alternative Bitcoin implementation discussed near the end of this piece. The quantitative record comes from the author's Bitcoin Governance Research, a public repository with its methodology and datasets, and its definitions and limits are summarized in the methodology section.

Key findings

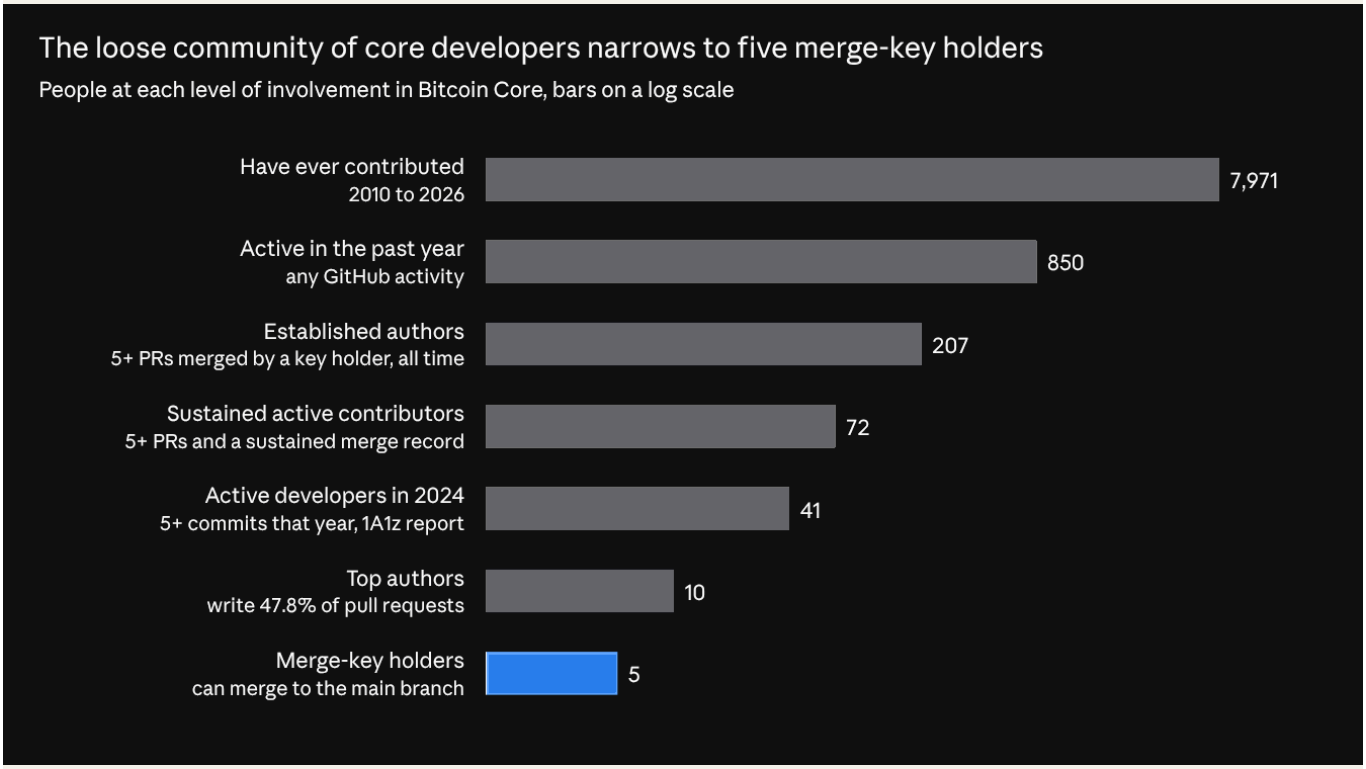
- Bitcoin ETF prospectuses disclose that a "loose community" of "core developers" informally manages the protocol's source code. The public record shows that community is a few dozen sustained contributors, and five people hold the keys that merge code.
- Bitcoin Core's written process sets no review minimum. Maintainers merged their own pull requests in about a quarter of their merges over the project's history, and in one in five during 2025.
- Measured against controls regulated firms already apply, SOX change management, SLSA two-person review, and DORA exit planning, Core meets none as written policy, though its practice has improved sharply since 2016.
- The supply cap has failed once and nearly failed again. A 2010 bug created 184 billion bitcoin, and a 2018 bug would have let a miner inflate supply. A small group of developers fixed both within a day.
- Every bitcoin ETF, custodian, and treasury company depends on the same software lineage, so switching products does not diversify the risk.
- Diligence teams can price this today as a governance modifier on position size, using the seven questions at the end.

Funds disclose the risk and stop there

US spot bitcoin ETFs held $111.09 billion in net assets as of October 8, 2026. Their prospectuses say who maintains the software that defines what they hold. The iShares Bitcoin Trust prospectus states that "a loose community known as the core developers has evolved to informally manage the source code," that membership evolves "largely based on self-determined participation," and that the core developers "are generally not compensated." The disclosure names a dependency, and the diligence that follows rarely asks who those people are, what controls govern their changes, or what happens when the controls fail.

The wording matters, and it varies by issuer in ways that tell a diligence team something. The iShares prospectus never names Bitcoin Core and writes "core developers" in lowercase, as whoever manages "the source code for the protocol." The ARK 21Shares prospectus uses nearly the same sentence, that "a loose community of core developers has evolved to informally manage the source code for the protocol," and elsewhere in the same filing says the network's infrastructure "is collectively maintained by a decentralized user base." The Grayscale Bitcoin Trust 10-K says bitcoin has "no official developer or group of developers that controls the Bitcoin Network," and then that "the Bitcoin Network's development is overseen by a core group of developers." Fidelity's 10-K is the only one of the four to name the project. It describes the developers of "a specific distribution of Bitcoin network software" known as "Bitcoin Core," calls it "the de facto standard," and says the core developers are "responsible for quasi-official releases." Read together, the filings treat the maintainers of one implementation as the stewards of the protocol, and the most specific of them says so outright.

The loose community is small enough to count



BTCDecoded contributor analysis of Bitcoin Core GitHub history, 2010 to 2026, and the 1A1z Funding Bitcoin report (October 2024). Established authors are counted over all time.

The public record lets the "loose community" be counted, and each step down the chart is a narrower definition of who does the work. Nearly 8,000 people have contributed at some point, 42% of them exactly once, and the set that sustains the project numbers in the dozens. Five of them hold the keys.

Core's own contributing guide disowns the category the prospectus relies on. It says "there is no particular concept of 'Bitcoin Core developers' in the sense of privileged people," and then names the exception, "repository maintainers who are responsible for merging pull requests." The two documents describe the same structure from opposite ends. The issuer discloses a loose community with authority over the protocol's source code, the project says the only people with special authority are its maintainers, and the data says those maintainers number five.

The same funds hold their coins under custody arrangements examined down to the signing policy. Diligence asks who holds the keys to the coins and leaves unasked who holds the keys to the code.

The asset is the rules, and the rules are code

A bitcoin is a balance that nodes accept as valid under rules written in software. The properties an allocator underwrites, a fixed supply, exclusive control of coins, and confidentiality of the holder's activity, hold only while that software enforces them correctly, so a change to the software is a change to the asset.

Bitcoin Core's record has a defect touching each of those properties. In versions 0.15 through 0.16.2, a removed check meant a miner could have inflated the supply by spending the same output twice. It was reported privately and fixed before anyone exploited it. In January 2026 the project pulled the binaries for versions 30.0 and 30.1 after finding that a failed wallet migration could delete every file in the wallet directory, "potentially resulting in a loss of funds." In version 31.0, the new private broadcast feature sometimes made connections over clearnet instead of the privacy network it was configured to use, which exposed the sender's IP address until version 31.1 fixed it. The consensus rules are a small part of the program, about 7% of Core's implementation, or 10,122 of 151,004 lines in version 32.1. The other 93%, the wallet, networking, interface, and the code that stores and relays what the rules accept, can reach a holder just as directly, and review has to cover all of it.

What a software failure does to a fund

The cost of a defect reaches a fund through its own operating terms. Take the case the record shows is possible, a consensus bug that splits the chain, as the 2013 version 0.8 release did when upgraded and older nodes followed different chains. For as long as a split lasts, two ledgers each claim to be bitcoin, and a custodian has to decide which one its client balances live on.

The iShares prospectus anticipates this. It lets the trustee suspend redemptions when delivering bitcoin is not reasonably practicable, including after a "significant technical failure, bug, error, disruption or fork of the Bitcoin network," and it commits the trust to "permanently and irrevocably abandon" any rights to assets a fork creates. A shareholder in that position cannot

redeem for coins, holds a claim on whichever chain the sponsor and custodian treat as bitcoin, and forfeits the other. With redemptions closed, the arbitrage that keeps an ETF's price near its net asset value stops working, so the share price can drift from the value of the coins behind it.

Wallet and privacy defects reach a fund another way. Custodians and issuers that run node software for validation or broadcasting inherit its defects, so a wallet bug can touch the funds they control and a privacy flaw can expose transaction flow that their clients expect to stay confidential.

The exposure is also correlated in a way allocators rarely see. Every US spot bitcoin ETF, every custodian, and every corporate treasury holding bitcoin depends on the same chain, and almost all of them validate it with software from one lineage. A failure in that software reaches every product at once, so moving between issuers or custodians diversifies counterparty risk and leaves this one untouched.

The supply cap has already failed once. In August 2010 a bug let a single transaction create more than 184 billion bitcoin. A patched client shipped within five hours, and the corrected chain overtook the bad one the next day. The 2018 bug came close to a repeat and was caught only because a developer of a competing implementation reported it privately. Both times, a supply that cannot inflate, the asset's defining property, was restored by a small group of developers acting fast, which is the arrangement this piece asks diligence to examine.

The 2018 response also shows who decides what holders learn. The project's own notice says the team chose "to immediately patch and disclose the less serious Denial of Service vulnerability" and held back the inflation flaw for three days "to give times for systems to upgrade." That was a defensible security call. It was also a decision about material information, made by a handful of people under no written procedure, with no one in the room for holders.

Core fails the controls a firm applies to every other critical system

A regulated firm already holds its critical software to three tests. Bitcoin Core meets none of them as written policy, though its current practice meets parts of them.

CONTROL THE FIRM ALREADY APPLIES	WHAT IT REQUIRES	BITCOIN CORE
Change management under SOX IT general controls	No single person can develop, approve, and deploy a change	Maintainers merged their own pull requests in 2,501 of 9,793 maintainer merges, and no written rule bars it
SLSA source integrity, Level 3	Every change approved by two trusted persons before submission	No written review minimum, though zero-review self-merges fell from 83.9% of self-merges in 2016 to 4.0% in 2026

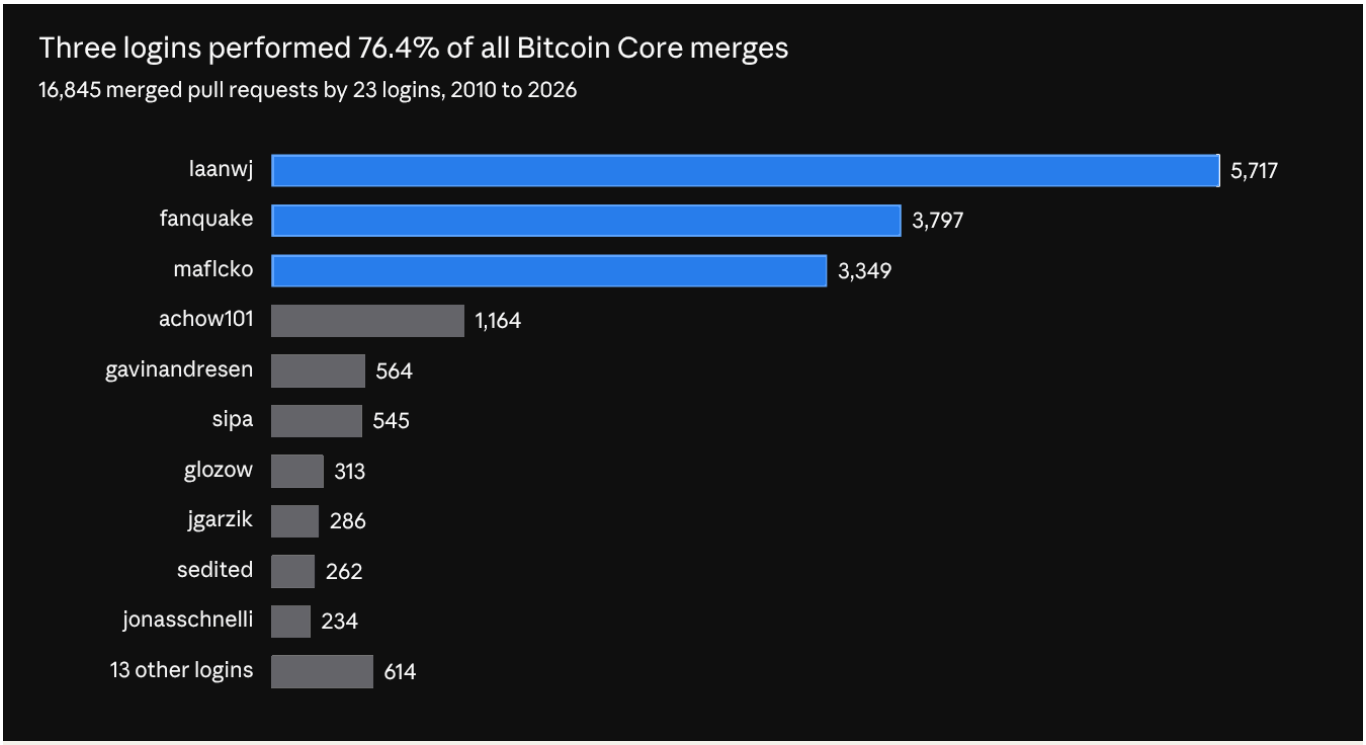
CONTROL THE FIRM ALREADY APPLIES	WHAT IT REQUIRES	BITCOIN CORE
ICT third-party risk under EU DORA	An assessment of concentration risk and a documented exit strategy	Core or code derived from it runs on most nodes, with no independent specification to exit against

The third test shows the deepest gap. DORA's Article 28 says that "for ICT services supporting critical or important functions, financial entities shall put in place exit strategies," and third-party risk programs enforce that through contracts, with audit rights, service levels, and termination clauses. Bitcoin Core has no CEO or foundation, no contract, and no counterparty, so the most critical dependency in a bitcoin position sits outside the program built to catch it.

Bank capital rules already treat the network as something to assess. Under the Basel Committee's cryptoasset standard, SCO60, an asset qualifies for the lightest treatment only if its network is "designed and operated to sufficiently mitigate and manage any material risks." Bitcoin falls into the harsher Group 2 on other grounds, since it has no stabilization mechanism, so the network question never gets asked of it. The standard names the question, and no bank has to answer it for bitcoin.

Five people hold the merge gate, and their duties to holders are unsettled

Five trusted-key holders can merge code into Core's main branch, and the existing maintainers decide who joins them. Over the full record, 23 GitHub logins have used merge keys, and three of them performed three-quarters of all merges. Node operators install releases without auditing them and rely on the maintainers' review, which makes that review the last control before code reaches the network.



BTCDecoded cross-repository comparison, merged_by field on 16,845 merged Bitcoin Core pull requests, 2010 to 2026.

The all-time share understates the gate as it works today, because the top three changed over the years. In every year since 2013, the three most active mergers performed between 88% and 97% of that year's merges, including 90.3% in 2025 and 91.2% so far in 2026. Since 2022 a single maintainer has merged half of all pull requests. In any given year, about nine in ten changes that reach Bitcoin Core pass through three people's hands, and the same holds at the last step before installation, where three people signed 90.1% of the releases whose signers are recorded.

No written procedure governs who joins the group. When a contributor proposed documenting the maintainer list and selection process in 2023, a long-time contributor described the process as "someone puts their name forward, it is discussed at a IRC meeting, they open a pull request," and objected to shoehorning the project "into a formal company structure" (#26868).

Counsel will recognize the shape as key-person dependence with no employment agreement, no succession plan, and no settled duty running to holders. In 2023 the English Court of Appeal held in Tulip Trading v van der Laan that whether Bitcoin developers owe fiduciary duties to coin owners was a serious issue to be tried. The claimant discontinued the case in 2024, so the question remains open, and a holder has no established legal claim on the people who change the rules of the asset.

Legal scholarship reached this point years ago. Angela Walch argued that "the core developers and significant miners of public blockchains function as fiduciaries" (In Code(rs) We Trust, 2019), and that the word "decentralized" can work as "a liability shield for those operating the systems

(developers and miners)" (Deconstructing 'Decentralization', 2019). She also warned that misreading where power sits produces inaccurate risk assessments of the assets built on these systems, which is the error this piece asks diligence teams to correct.

The law is moving toward fewer obligations for developers. The CLARITY Act, which passed the House in July 2025, carries the Blockchain Regulatory Certainty Act in its Senate text and would shield non-custodial developers from money-transmitter and Bank Secrecy Act obligations. The bill failed a Senate cloture vote 49 to 50 on September 15, 2026, and no new vote is scheduled. The provision concerns licensing, not fiduciary duty, and its direction is still clear. Policy treats developers as publishers of software, so any control over how they change it will have to come from the project or from the institutions that rely on it.

How rule changes get decided

Merge statistics show who approves code. Changes to the rules show how contested decisions get made, and they follow no written procedure either. Taproot, the most recent consensus upgrade, locked in in June 2021 under a "Speedy Trial" mechanism that required 90% of blocks in a signaling period to support it, after months of public argument over how activation should work. The method was chosen for that upgrade and binds none that come after it.

Policy changes, which alter what a node relays without changing consensus, need less. Version 30, released in October 2025, removed the default 80-byte limit on OP_RETURN data. Critics, among them developer Luke Dashjr and Nick Szabo, warned that it invited spam and permanent storage of illicit data, and one lawyer argued that a change this contested should not ship bundled with security fixes. The maintainers shipped it. Over roughly four months, Bitcoin Knots, a Core derivative that kept the limit, rose from about 4% to about 21% of the nodes counted by coin.dance. The episode shows the decision process as it actually runs. The maintainers judge rough consensus, and an operator who disagrees can switch to a different build of the same code.

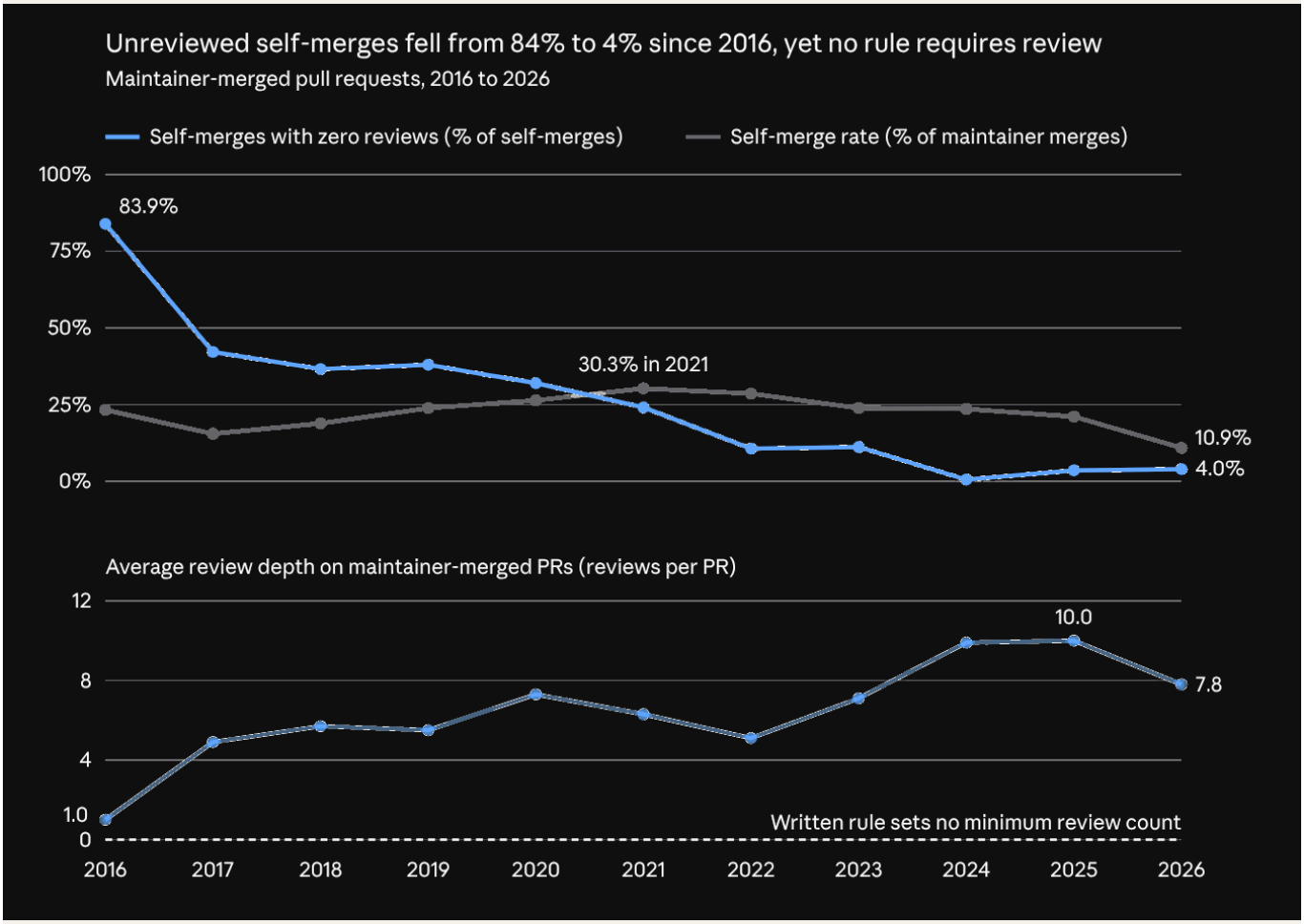
The next contested change is already in the filings. BlackRock's July 2025 iShares prospectus warns that quantum-resistant changes "could require the achievement of broad consensus within the Bitcoin network community and a fork (or multiple forks), and there can be no assurance that such consensus would be achieved or the changes implemented successfully." A migration of that size will run through the same unwritten process that took Taproot years, and the largest issuer has already told its investors the result may be a split chain.

Who pays for the work

The prospectus line that developers are "generally not compensated" understates how the work is funded. A 2024 report by the research group 1A1z, Funding Bitcoin, counted $8.4 million spent on Core developers in 2023, spread across 13 sponsoring organizations, and found 41 developers with at least five commits in 2024. Two grant-makers, OpenSats and Spiral, provided 62% of grant funding, and Chaincode Labs accounted for 46% of employment spending. Eight sponsors depend entirely on donations, and the top three of those receive 62.25% of their funding from a single source.

Diligence asks about funding concentration because it is how influence enters an organization without a vote. The people who control the merge gate depend on a small number of discretionary grants, many renewed annually, with no disclosure owed to holders. A 2023 budget of $8.4 million maintains software that now secures $111 billion in US ETF assets alone.

Review practice has improved, and it rests on custom



BTCDecoded Bitcoin Core governance research, TEMPORAL_ANALYSIS_REPORT, 2016 to 2026 to date (2026 is a partial year). GitHub review records exist from 2016.

Since GitHub review records begin in 2016, average review depth on maintainer-merged pull requests has risen from 1.0 to 7.8, and zero-review self-merges have fallen from 83.9% of self-merges to 4.0%. Active reviewers grew from 22 to 94 by 2021, and pull requests per reviewer fell from 71 to 21. A 2025 Quarkslab audit called Core the most mature and well tested codebase its engineers had assessed and found no high-impact issues in its peer-to-peer scope.

Incident handling has matured too. Since 2024 the project has published security advisories under a written disclosure policy that ranks bugs from low to critical and sets disclosure timelines for each grade. The list now holds 28 entries, one of them critical, the 2018 inflation bug. The policy also says critical bugs fall outside its standard timelines and would need an ad hoc procedure, so the most serious class of defect still has no written process.

Diligence should credit all of it, and should also notice what holds it in place. Core's contributing guide sets no numeric review minimum and leaves merge decisions to the maintainers' judgment of contributor consensus. Review stays concentrated, with a Gini coefficient of about 0.93 and the top ten reviewers producing 39% of reviews since 2021. A control that exists as the habit of a small group lasts as long as that group does, and nothing in the written process obliges the next group to keep it. Set beside sixteen other Bitcoin repositories on a shared health index, Core's rank moves with how size is handled, so the finding concerns missing written rules and not measured underperformance.

The gap between the rule as described and the rule as written shows up in the project's own funders. Brink, one of the largest funders of Core developers, told supporters in its 2025 engineering report that "every change to Bitcoin Core must be carefully reviewed by multiple qualified developers before it can be merged." That describes a two-person rule. The contributing guide contains no such requirement, and in 2025 one in five maintainer merges was the maintainer's own pull request.

The developers describe the shortage in their own words. Pieter Wuille wrote in 2024 that "review is the biggest bottleneck for our project" (#29473). Anthony Towns warned in 2021 that the shortage creates "an incentive to get in the habit of merging code that's not thoroughly reviewed in order to avoid delaying things because we've run out of review resources" (#20721). When maintainers debated self-merging in 2022, Marco Falke estimated that "about half of my changes are merged by myself" and wrote that "in the absence of several active maintainers per file/module, I fail to see how self merges could be rare" (#25524). The self-merge rate is what review scarcity looks like from the merge button.

The project has also considered writing the rule down. A 2022 pull request to document maintainer responsibilities (#25560) met pushback from contributors and maintainers who wanted it cut back to minor fixes, in part, as one contributor summarized, to avoid "a restriction that a maintainer can't merge their own PR no matter how much review it has."

Core's release process shows the project can write a multi-party rule when it chooses to. Release binaries go out only after "6 or more people have guix-built and their results match" (release process), so no single person can ship a binary that differs from the source. Pieter Wuille has been clear about what that control leaves out. A build attestation "isn't and shouldn't be interpreted as the builder/attester vouching for the code they are building," he wrote, because "we have the review process for that" (#21462). The project writes a numeric rule for the step that checks binaries against source and leaves the step that checks the source itself to custom.

Other critical projects write the rule down

A written review rule is ordinary for security-critical open source. OpenSSL, the cryptography library that drew global scrutiny after the 2014 Heartbleed flaw, states in its committer policy that "all submissions must be reviewed and approved by at least two committers" and that "neither of the reviewers can be the author of the submission." That is the SLSA two-person rule in policy form, adopted by an open-source project under the same review pressure Core faces. Core's practice now often exceeds that bar, and its written process still does not state it.

Bitcoin's code is a commons, and a commons without rules runs down

Core's codebase is a commons that many depend on and no one owns. Its upkeep draws on two scarce inputs, reviewer expertise and the consistent application of the design principles that keep the software safe. Elinor Ostrom's study of commons that last found that durable ones share a set of design principles, and a review of 91 later studies found strong support for them. Two bear directly on Core. The people affected by the rules take part in changing them, and the people who monitor use are accountable to the users. Core has a self-selecting maintainer group and no written review rule, and the holders who bear the losses, institutions now among them, have no part in setting either.

When principles live in reviewers' heads, they get applied unevenly. The 2013 split came from a database limit that old nodes enforced and new nodes did not, and the 2018 bug from a check a later optimization judged redundant. In both cases no build step enforced a rule the network depended on.

The concentration is documented in the academic literature. Azouvi, Maller, and Meiklejohn found in 2018 that "there is usually a handful of people that accounts for most of the discussion" in Bitcoin's development. De Filippi and Loveluck concluded that "the governance of the Bitcoin project relies almost exclusively on its leaders," and Yeung and Galindo argued in 2019 that public blockchains without formal internal governance are unlikely to be taken up at scale.

The strongest scholarly counterview comes from Nabilou, who argues that Bitcoin "achieves decentralized governance by putting users at the center," because node operators can refuse to run code they reject. That veto is real for intended rule changes that users can see and debate. It

does not reach a defect nobody has found, and exercising it requires an alternative implementation, which brings the argument back to the cost of exit.

The exit option costs more than it appears

The standard answer is that operators can refuse an upgrade or switch software. Exercising that option is expensive, because the alternatives share Core's code. Bitcoin Knots, the largest, is a patch set maintained on top of Core, so the operators who left Core over version 30 still run Core's code. The 2018 bug hit Core and Knots as well as Bitcoin ABC, Litecoin, and Dash, because a flaw travels with the lineage that carries it. Core has never published a formal specification of its consensus rules, so an independent implementation has to reproduce Core's behavior from Core itself, and reimplementations have diverged in edge cases, including script validation in btcd. In the Quarkslab audit, cross-implementation fuzzing found most of its issues in the alternatives. In third-party risk terms, that is a concentration with no tested exit.

The objections, answered

The first objection is that the system has worked for seventeen years. The record is real, and it belongs to particular people. The improvement since 2016 came from the habits of one group, and nothing written obliges its successors to keep them. Diligence treats a track record with no control behind it as reliance on management and prices it as key-person risk.

The second is that the network is decentralized. Validation is, since thousands of independent operators each check the chain. Code approval is not. Five key holders control what goes into the software most of those operators install, and the operators upgrade without auditing the diff.

The third is that anyone can review open source. Anyone can, and few do. Review has a Gini coefficient of about 0.93, and ten people produce 39% of it. The right to review is open, and the work is concentrated.

The fourth is that consensus changes need miners and users to agree. That holds for intended changes to the rules, which miners and economic nodes do gate, and it does nothing about bugs. The 2018 inflation flaw, the version 30 wallet bug, and the 31.0 privacy flaw all shipped in ordinary releases with no activation vote, because nobody intended them.

The fifth is that regulating developers would be worse than the problem. Nothing here asks a regulator to act. The questions are for allocators and counsel deciding how much exposure to take, and for issuers deciding what to disclose. A project can adopt written controls on its own, as OpenSSL did.

The sixth is that no fund diligences Linux or the internet's protocols either. A defect in an operating system can stop a firm's systems, and the firm controls that exposure through its own patching and change management. A defect in Bitcoin's consensus code changes what the firm owns, and no internal control can patch a ledger that has split.

The seventh, and the one Bitcoin developers press hardest, is that the informality is deliberate. A foundation, a board, or a charter gives courts and governments someone to sue or pressure, and diffuse authority protects the people who maintain the code. The concern is real, and investors should price the trade-off it describes, though informality has not stopped suits either, since the Tulip Trading claim named individual developers. It does not explain the missing review rule, though. A written review rule binds a process, creates no legal entity, and hands no one a new target, which is why OpenSSL can publish one.

The project's own history shows the pattern behind that defense. Bitcoin launched with an alert system whose messages, Satoshi wrote in 2010, were "signed with a private key that only I have" (2010 post). When it was removed in 2016, Wladimir van der Laan argued that it was "seen as a centralized point of control" and that a network-wide alert system "would also need a network-wide bureaucracy for managing it" (#7692). Removing it was a reasonable call. It also shows the project's settled preference, which is to remove a lever rather than write rules for it, and the merge gate is a lever no project can remove.

A governance-aware design can be built

Bitcoin Commons, the author's project, is a design for building and governing a Bitcoin implementation under written controls. Its status is stated row by row.

TEST	BITCOIN CORE TODAY	BITCOIN COMMONS DESIGN	STATUS
Who can merge	Five trusted-key holders who choose their successors	Signature thresholds by layer, from 2-of-3 to 6-of-7, checked by a merge gate that administrators cannot bypass	Designed, enforcement off
How much review	No written minimum	Review periods from 14 days at the extension layer to 180 days at constitutional layers, and 365 days for consensus-rule changes	Configured, enforcement off
What counts as correct	Core's own behavior	An independent specification of 159 consensus rules, with 251 production functions checked against it by the Z3 solver	Built
Proof of compatibility	None	Differential testing against Core across chain history, targeted at about	Program exists, completion record

TEST	BITCOIN CORE TODAY	BITCOIN COMMONS DESIGN	STATUS
		900,000 blocks	not published
Cost of exit	High, with no specification	Lower, because compatibility can be tested against a specification	Depends on the row above

The verification half is further along, and a node built from a written specification goes at the exit problem directly. The governance half is configured with enforcement switched off. Two public proofs would turn it from design into evidence, a completed full-history differential run and a fail-closed merge check on a real repository. What the design shows already is that written controls for a Bitcoin implementation can be specified.

Peter Todd raised the standard objection to specifications during the 2013 chain split. "Specifications aren't magic; they're just words on paper," he wrote, and writing one "doesn't magically make code actually follow the spec" (2013 thread). For a document that sits beside the code, he was right. The Commons design binds the two mechanically. A tool called spec-lock parses the formulas in the specification and the annotated Rust functions that implement them, ties each function to its rule, and uses the Z3 solver to check that they still agree, so drift between rule and code fails verification instead of waiting for a reviewer to notice. The binding covers the 251 locked functions, not the whole node, and a rule written wrong in the specification would be locked in wrong, which is what differential testing against Core is for.

What allocators and counsel can do now

1. Put the node software on the dependency register. Ask issuers and custodians which implementation and version they run, who approves upgrades, and whether that upgrade process falls inside their SOC 2 scope.
2. Ask who can merge changes to the software and who decides who can. For Core, five trusted-key holders who choose their successors.
3. Ask for the written review rule. For Core, none exists, and practice has improved since 2016.
4. Count independent implementations by code lineage, not by client name, and ask whether correctness is defined by a specification independent of the incumbent. For Core, most nodes run Core or its derivatives, and no formal specification has been published.
5. Ask who funds the people with merge authority and on what terms. For Core, a small number of annual grants, concentrated around one donor at the time of the 2024 report.
6. Ask how a fork would be handled. Read the issuer's redemption-suspension and forked-asset terms, and ask the custodian how it decides which chain is bitcoin.
7. Ask issuers to disclose the controls in their risk factors, beyond the existence of the developers.

S&P's corporate methodology applies governance as a modifier that moves a rating by notches, and Fisher's 15 points treat management integrity as a veto. The answers can be scored the same way, as a modifier on position size by default and as a gate when nothing independent can verify the software. They also need re-running on a schedule, because they depend on a small group of people.

The risk is disclosed, measurable, and unpriced

Every major issuer already tells investors that a small, informal group maintains the software their product depends on. The public record shows how small that group is, how it reviews and merges code, and where its written rules stop. The controls needed to price the exposure, change management, two-person review, exit planning, and governance modifiers, are ones a regulated firm already applies to every other critical system. Applying them to bitcoin needs no new framework, and the questionnaire below puts them in the form diligence teams already use.

Appendix. Questionnaire items

The seven questions, written for an operational due diligence questionnaire sent to an issuer, custodian, or treasury company.

TOPIC	QUESTION	WHAT A STRONG ANSWER SHOWS
Software dependency	Which node implementation and version do you run to validate client holdings, and who approves upgrades?	A named implementation and version, a written upgrade policy with a named approver, and the upgrade process inside SOC 2 or equivalent audit scope
Merge authority	Who can merge changes to that software, and how are they chosen and removed?	Named people or roles and a documented admission and removal process
Review rule	What written rule sets the minimum review before a change merges?	A published two-person or stricter rule that excludes the author as a reviewer
Implementation diversity	How many independent implementations validate the chain you hold, counted by code lineage?	At least one implementation not derived from the incumbent's code, and a plan to test against it
Specification	Is correctness defined by a specification independent of the incumbent implementation?	A published specification and a record of compatibility testing against it
Funding	Who funds the people with merge authority, and on what terms?	Named funders, multi-year terms, and no single source above a stated share

TOPIC	QUESTION	WHAT A STRONG ANSWER SHOWS
Fork handling	If the chain splits, how do you decide which chain is bitcoin, and what happens to redemptions?	A written fork policy naming the decision-maker, the criteria, and the timeline for telling clients

The last item maps to the redemption-suspension and forked-asset terms in each issuer's prospectus, so counsel can test the answer against the filing.

Methodology

The quantitative record comes from Bitcoin Governance Research, an analysis of Bitcoin Core's public history covering 25,122 pull requests and 339 releases from 2010 to 2026, with 8,890 issues, 51,062 mailing-list emails, 430,613 IRC messages, and forum archives used for context. The repository holds the scripts, datasets, and reports behind every figure here, starting from findings/EXECUTIVE_SUMMARY.md.

A self-merge is a merged pull request whose author and merger are the same person, and a pull request with no recorded merger is not counted as one. Maintainer figures use a roster of maintainer logins, under which 2,501 of 9,793 roster-authored merges are self-merges. A pull request counts as zero-review when it has no GitHub review object with a state of approved, changes requested, or commented. Core publishes no maintainers file, so the 23 merge-key holders are inferred from observed merge activity.

GitHub had no review functionality before 2016, so review figures start there, and the all-time zero-review share of self-merges, 45.1%, overstates the gap in early years for that reason. Review objects also miss review done in comments, on IRC, or offline, so the counts are a floor. Sustained active contributors are those with at least five pull requests and a minimum average quality score in the contributor timeline analysis. The 17-repository peer comparison uses a shared health index without review data, and its intervals cover zero. Figures for 2026 cover a partial year. The analysis found no measurable link between self-merging and later reverts, so the argument here rests on missing controls and not on measured harm.

Sources

Exposure and disclosure. US spot bitcoin ETF assets, October 2026 (edgeX), iShares Bitcoin Trust prospectus, August 2024 (SEC), iShares Bitcoin Trust prospectus, July 2025 (SEC), ARK 21Shares Bitcoin ETF prospectus (SEC), Grayscale Bitcoin Trust 10-K, FY2024 (SEC), Fidelity Wise Origin Bitcoin Fund 10-K, FY2025 (SEC)

Control frameworks. Program change controls (ITGC), SLSA source requirements v1.1, DORA Article 28 text, Basel Committee SCO60 cryptoasset standard, OpenSSL committer policy, S&P corporate methodology, Fisher's 15 points

Legal duties. Tulip Trading v van der Laan, EWCA Civ 83 (2023), Claim discontinued, CLARITY Act status, Senate Banking fact sheet on developer protections

Research. Walch, In Code(rs) We Trust (2019), Walch, Deconstructing 'Decentralization' (2019), Azouvi, Maller, and Meiklejohn, FC 2018, De Filippi and Loveluck, Internet Policy Review 2016, Yeung and Galindo, European Journal of Risk Regulation 2019, Nabilou, Stanford Journal of Blockchain Law and Policy 2020

Rule changes and funding. Taproot lock-in, Bitcoin Core v30 and OP_RETURN, 1A1z Funding Bitcoin report, Brink engineering report 2025

Bitcoin Core. Value overflow incident, CVE-2010-5139, CVE-2018-17144 disclosure, Wallet migration bug notice, January 2026, Bitcoin Core 31.1 release notes, Security advisories, CVE-2018-17144 affected versions, Bitcoin ABC incident report, Consensus libraries, Quarkslab audit, Bitcoin Programming Textbook, Governance model, CONTRIBUTING.md

Developer record. Wuille on review as the bottleneck, #29473, Towns on the review shortage, #20721, Self-merge debate, #25524, Maintainer responsibilities proposal, #25560, Maintainer selection discussion, #26868, Release signer concentration

Historical record. Todd on specifications, 2013, Satoshi on the alert system, 2010, Alert system removal, #7692, Release process (Guix attestations), Wuille on attestations and review, #21462

Code composition. The Rules Are 7% of the Program, Bitcoin Core 32.1 line counts

Bitcoin Commons. Whitepaper, Consensus specification, Differential testing guide

Quantitative record. Bitcoin Governance Research, covering the executive summary, research methodology, temporal analysis, merge pattern breakdown, cross-repository and size-adjusted comparisons, and contributor analysis.

SECURESOVEREIGN

Markdown is the canonical source: https://secsov.com/articles/who-governs-bitcoins-code/index.md

Formatted snapshot of https://secsov.com/articles/who-governs-bitcoins-code

Secure Sovereign