The Rules Are 7% of the Program

Josh · Secure Sovereign

Published October 30, 2026 · Updated November 1, 2026

A line count of Bitcoin Core 32.1

*Release **32.1**, branch `32.x`, commit `f0de87f`, 26 October 2026. Current-tree figures are tokei 14 code lines. History figures are git numstat on non-merge commits.*

The consensus rules are **10,122** code lines. The implementation they sit in is **151,004**. The code that applies those rules, stores the result, or chooses what to relay and mine is **20,366**, twice the rules. The other **120,516** lines are the wallet, the GUI, RPC, P2P transport, and the rest of the node.

A wrong rule forks the network. A policy disagreement does not. The two live in the same program, and the rules are the thin slice. That is the measurement behind *The Vertical Layer Problem* and *Why Bitcoin Needs a Specification That Humans Can Read*.

Contents

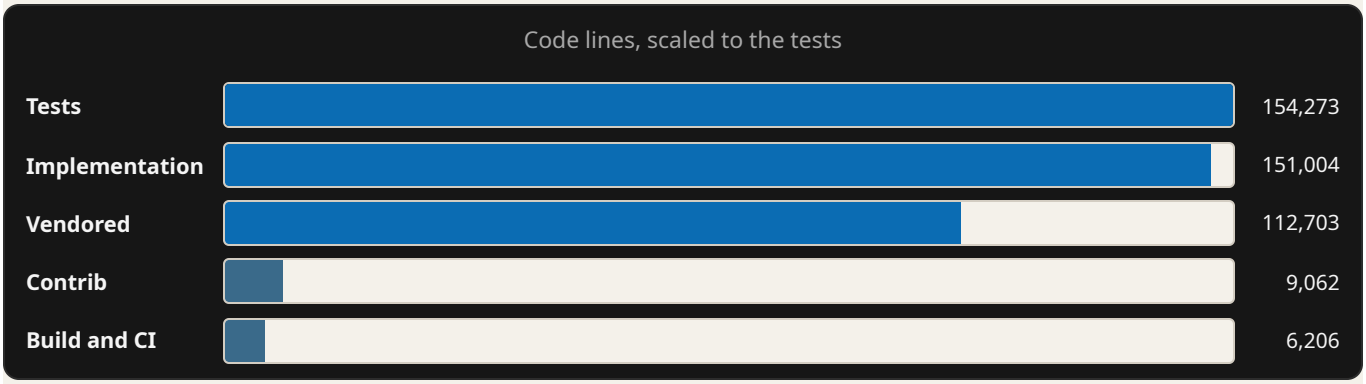
- I. What was counted
- II. The tree
- III. The implementation
- IV. The rules
- V. What sits beside the rules
- VI. Tests
- VII. Bundled code
- VIII. Who added the lines
- Where the work went
- What changes in the same commit
- IX. Who last touched what remains
- Who still edits a system
- The size of the contract
- X. Patch debt
- XI. What the count is for
- XII. Method

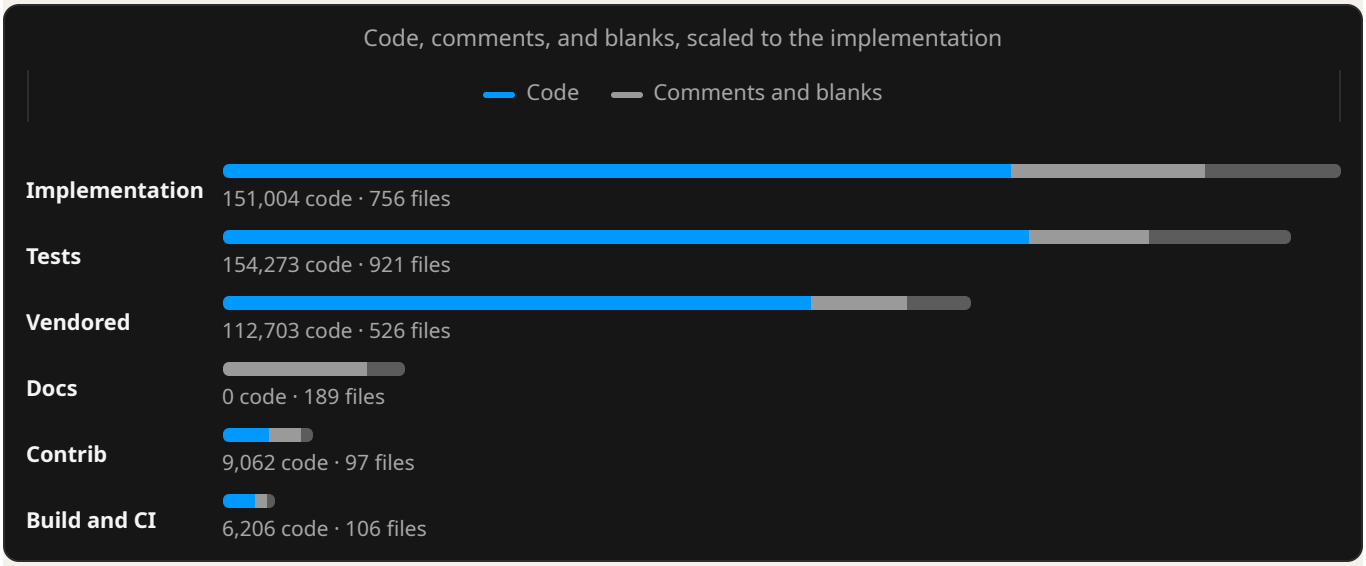
I. What was counted

Tokei parsed **2,698** files and **719,989** code lines. Qt locale files are 103 of those files and **286,741** of those lines. Tokei counts them as TypeScript. They are translations, and the charts below leave them off. Every other parsed file is in exactly one bucket.

The headline is the implementation. Vendored libraries are another 112,703. First-party code, implementation plus tests plus contrib, build, and docs, is **320,545**. Those three figures, 286,741 plus 112,703 plus 320,545, are the 719,989.

II. The tree

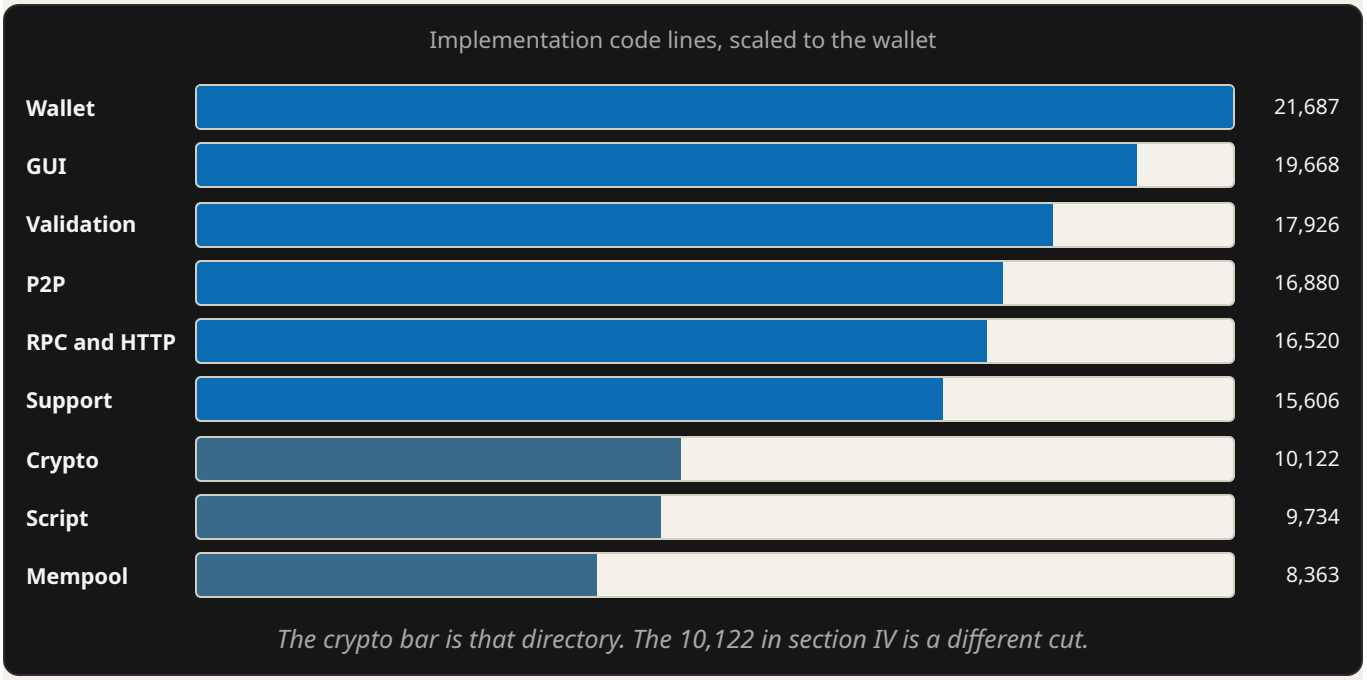




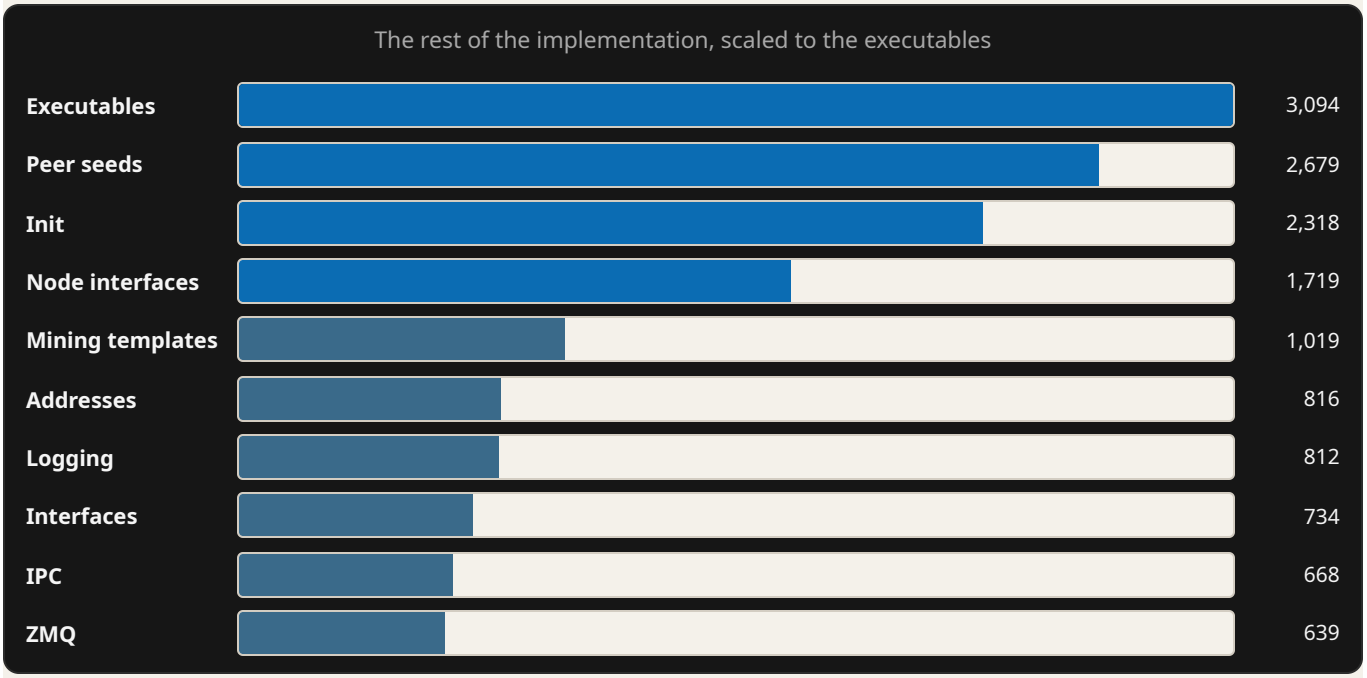
The program carries about as much test code as implementation code. On this release the test pile is the larger of the two. Implementation bodies are 73.8% C++ and 25.6% headers.

III. The implementation

Seven hundred fifty-six files. The median file is 66 code lines. Thirty-three files of 1,000 lines or more hold 38.9% of the implementation. Fifty-four files hold half of it.



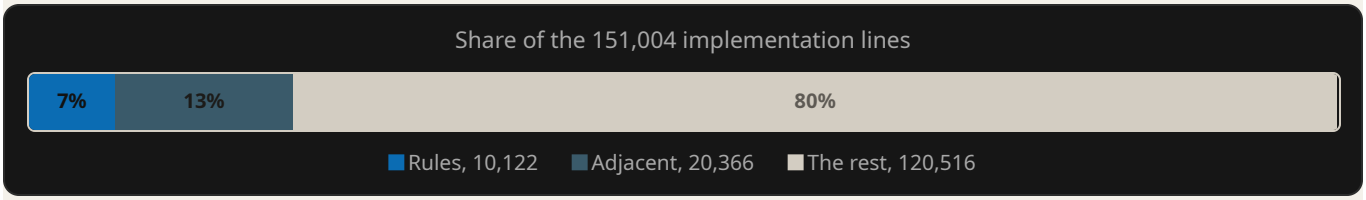
The crypto bar is that directory. The 10,122 in section IV is a different cut.

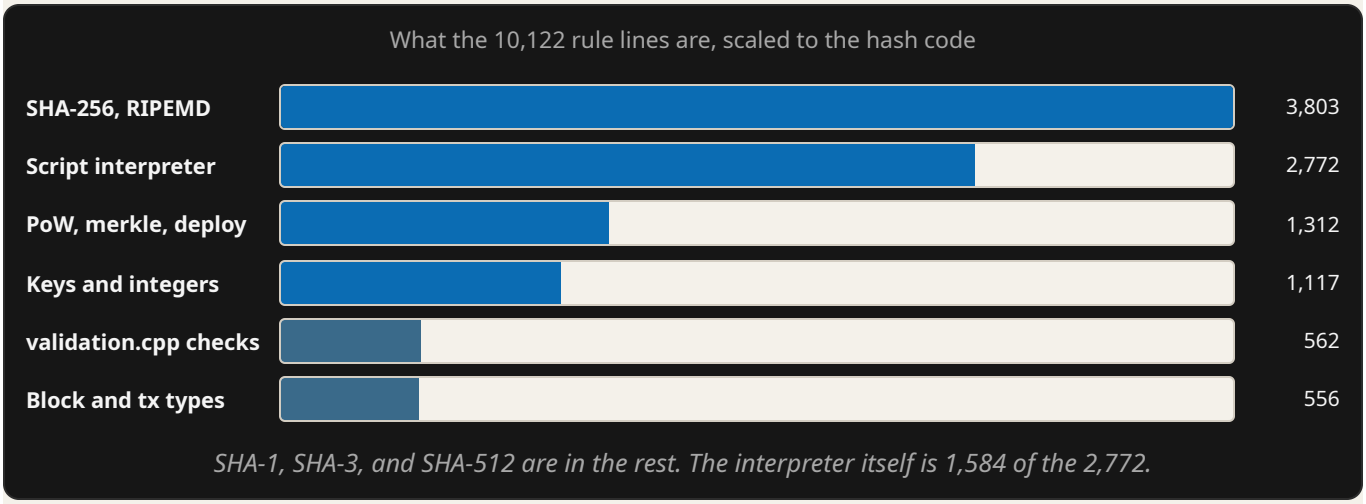


`src/node` is not all P2P. Block storage and chainstate in that directory are in the validation bar. P2P is the network files only.

IV. The rules

Consensus here means the code that decides whether a block or a transaction is valid. The bundled signature library is outside this split. It is counted in section VII.

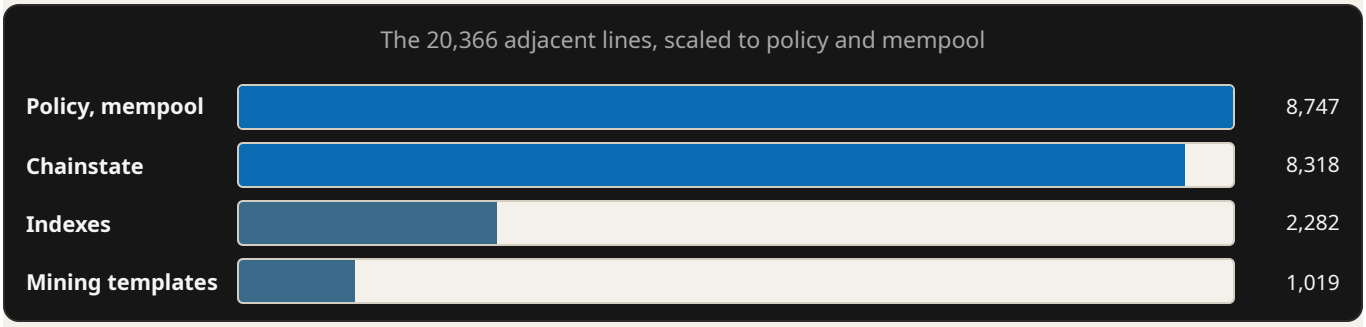




Core's exported `libbitcoinconsensus` is **4,757** of these lines. The kernel target is a different list, and it still compiles policy and mempool.

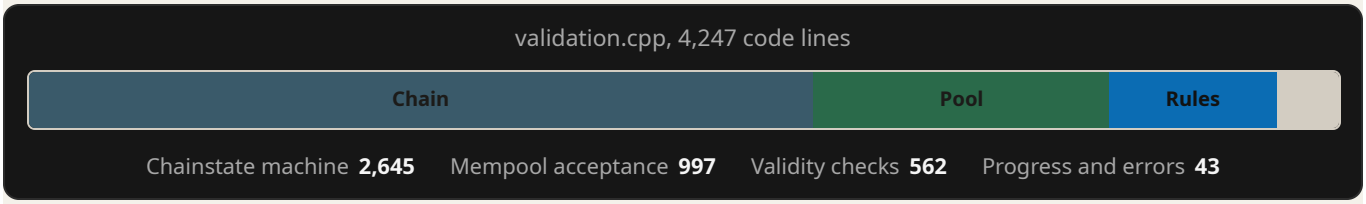
V. What sits beside the rules

Adjacent means the code that applies the rules, stores the result, or chooses what to relay and mine. An index can be wrong without making a block invalid. A policy limit can change without forking.



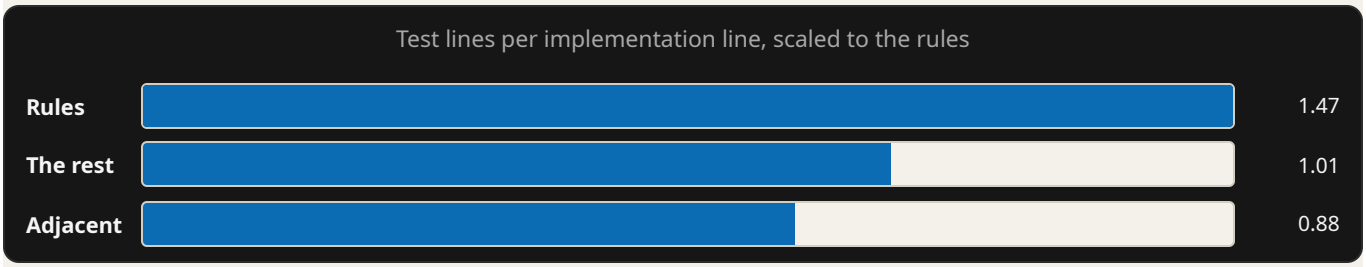
validation.cpp is mostly the machine around the checks

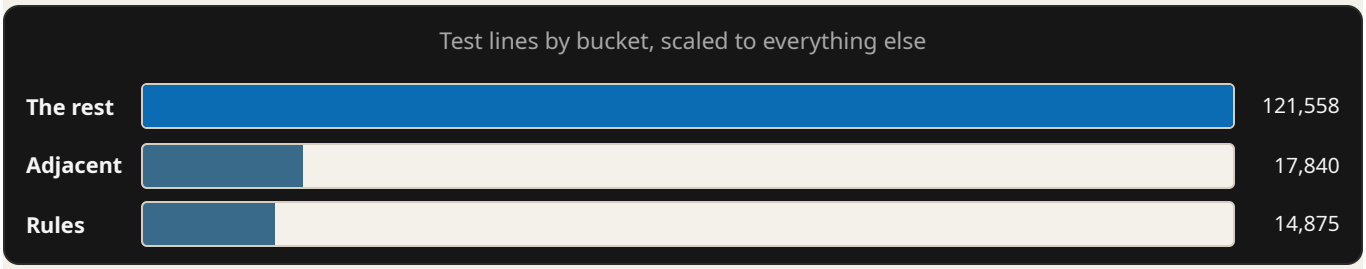
The file is 4,247 code lines. CheckBlock, ConnectBlock's checks, the subsidy, and BIP30 stay with the rules. The undo write stays with chainstate. MemPoolAccept stays with policy.

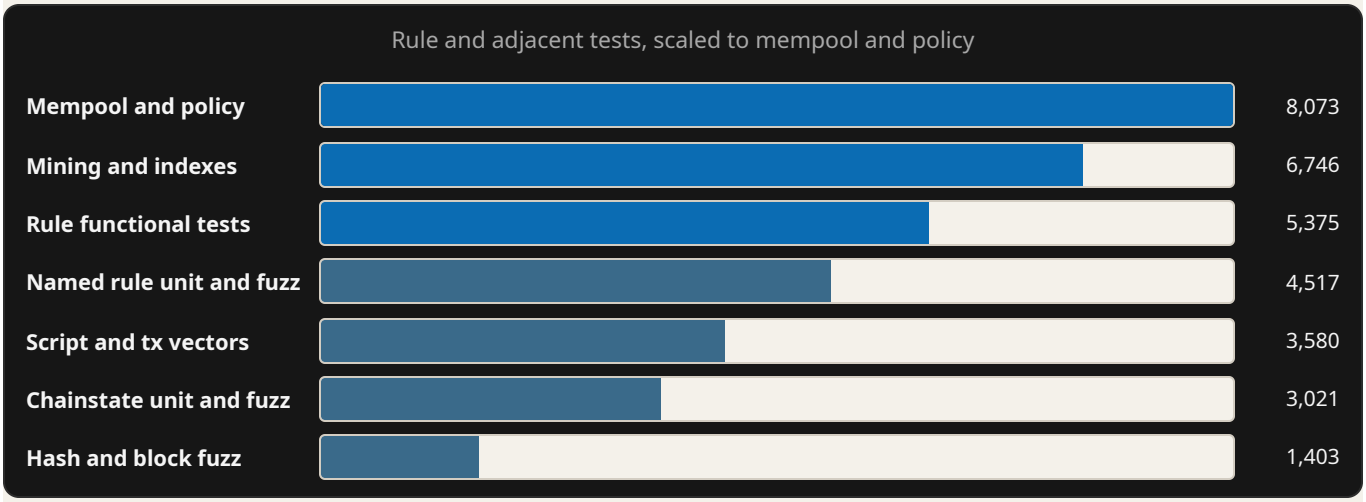


VI. Tests

Tests are 154,273 code lines, about as many as the implementation.



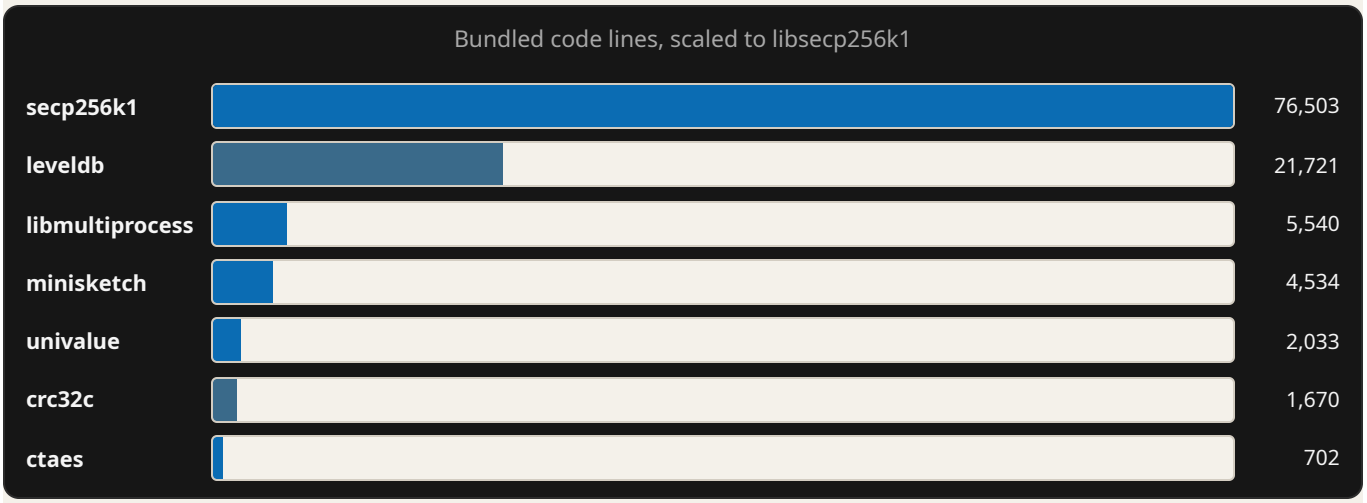




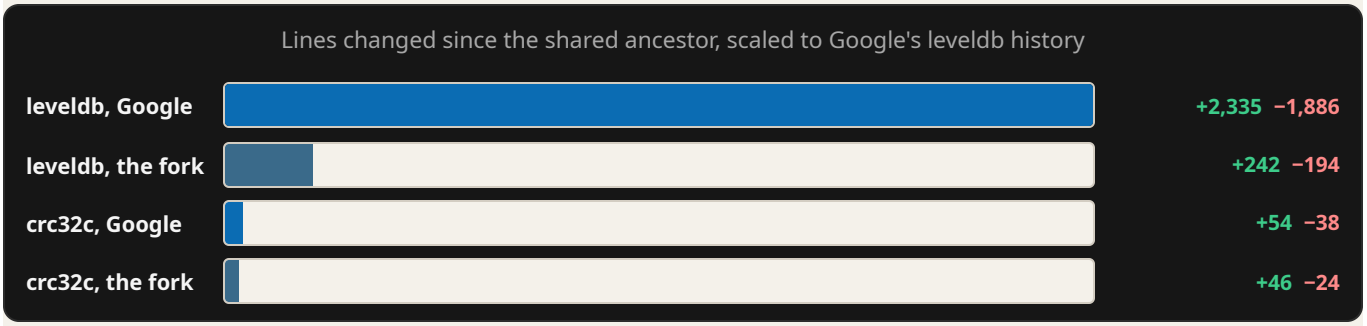
The long bar is wallet, GUI, P2P transport, lint, and CI. A test that calls the interpreter under another name is in that bar.

VII. Bundled code

Vendored trees sum to 112,703 code lines. **89,312** (79%) are maintained as Bitcoin Core projects. **23,391** (21%) are forks of Google libraries. secp256k1 is most of the first group, and it is 7.6 times the first-party rules.



Since the projects split, Google has changed leveldb about ten times as much as this fork has. crc32c has barely moved on either side.



Qt, Boost, SQLite, ZeroMQ, Cap'n Proto, qrencode, and systemtap are fetched when the node is built. They are not in this checkout, so they have no line count here.

Public-key checks in `src/pubkey.cpp` call into secp256k1. That library is a Core project, used by other software, and it sits outside the node's own source.

VIII. Who added the lines

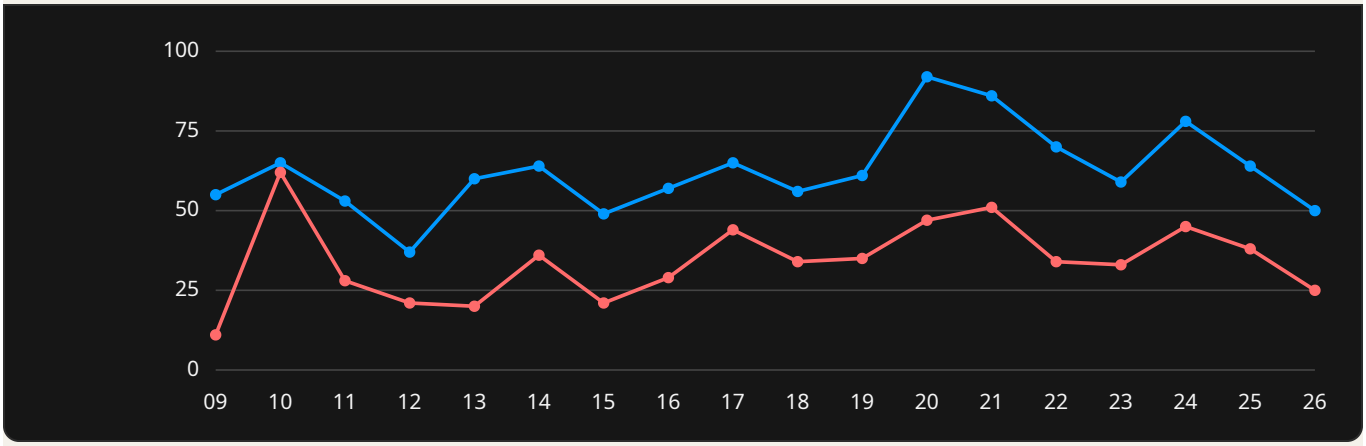
1,122,543 lines were added and **614,627** were deleted. **55%** of what was added was later deleted. Net is added minus deleted. It is not the lines still in the tree. That count is section IX. The filter is code files only. Comments and blanks count. Vendored trees, Qt translations, and build scripts do not.

Of the lines added to code files, through 26 October 2026

55% deleted | 45% net

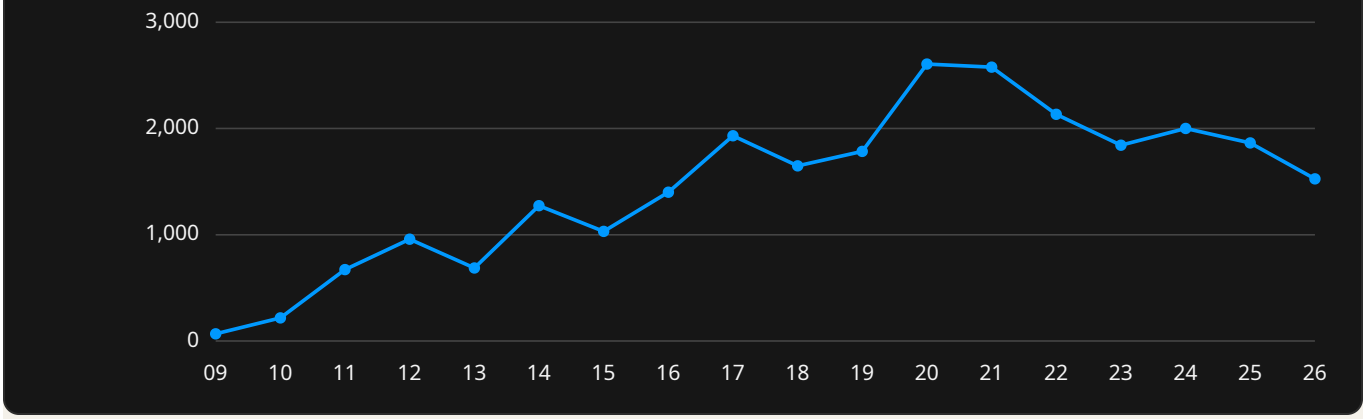
Added **+1,122,543** Deleted **−614,627** Net **+507,916**

Lines added and deleted, by year



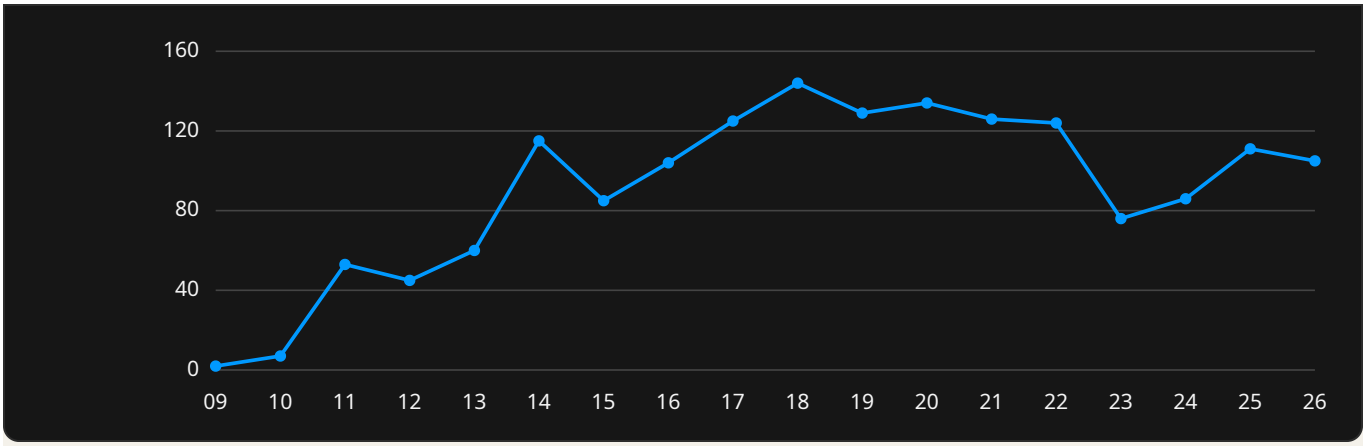
Thousands of lines, rounded. The largest net is 2020, at 44,625. 2010's net is 3,185.

Code commits by year

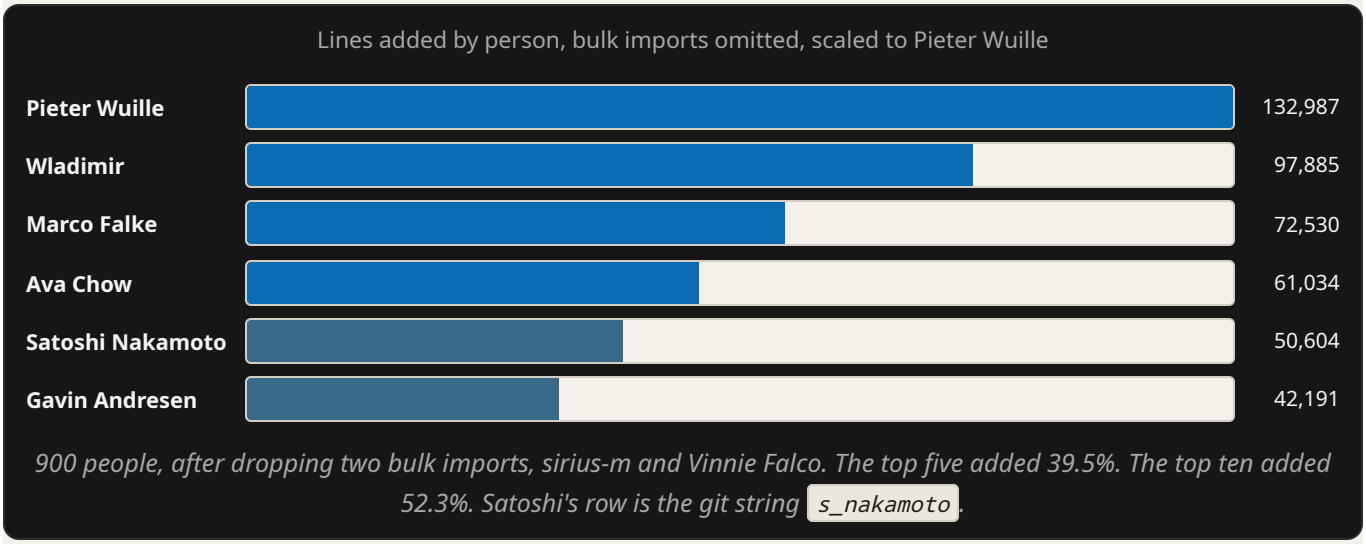


The peak is 2,606 commits in 2020.

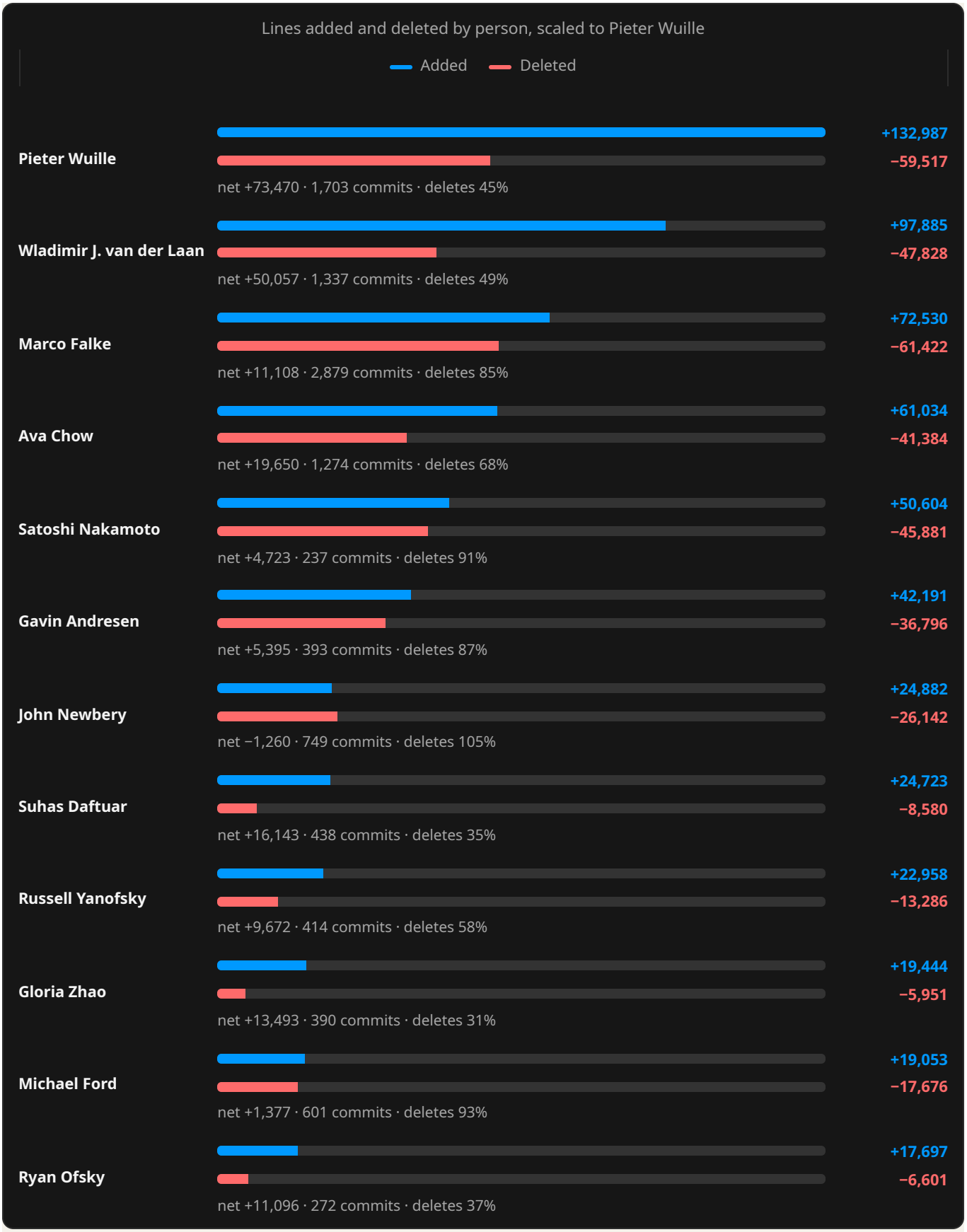
Distinct people by year



The peak is 144 people in 2018. 2023 falls to 76.



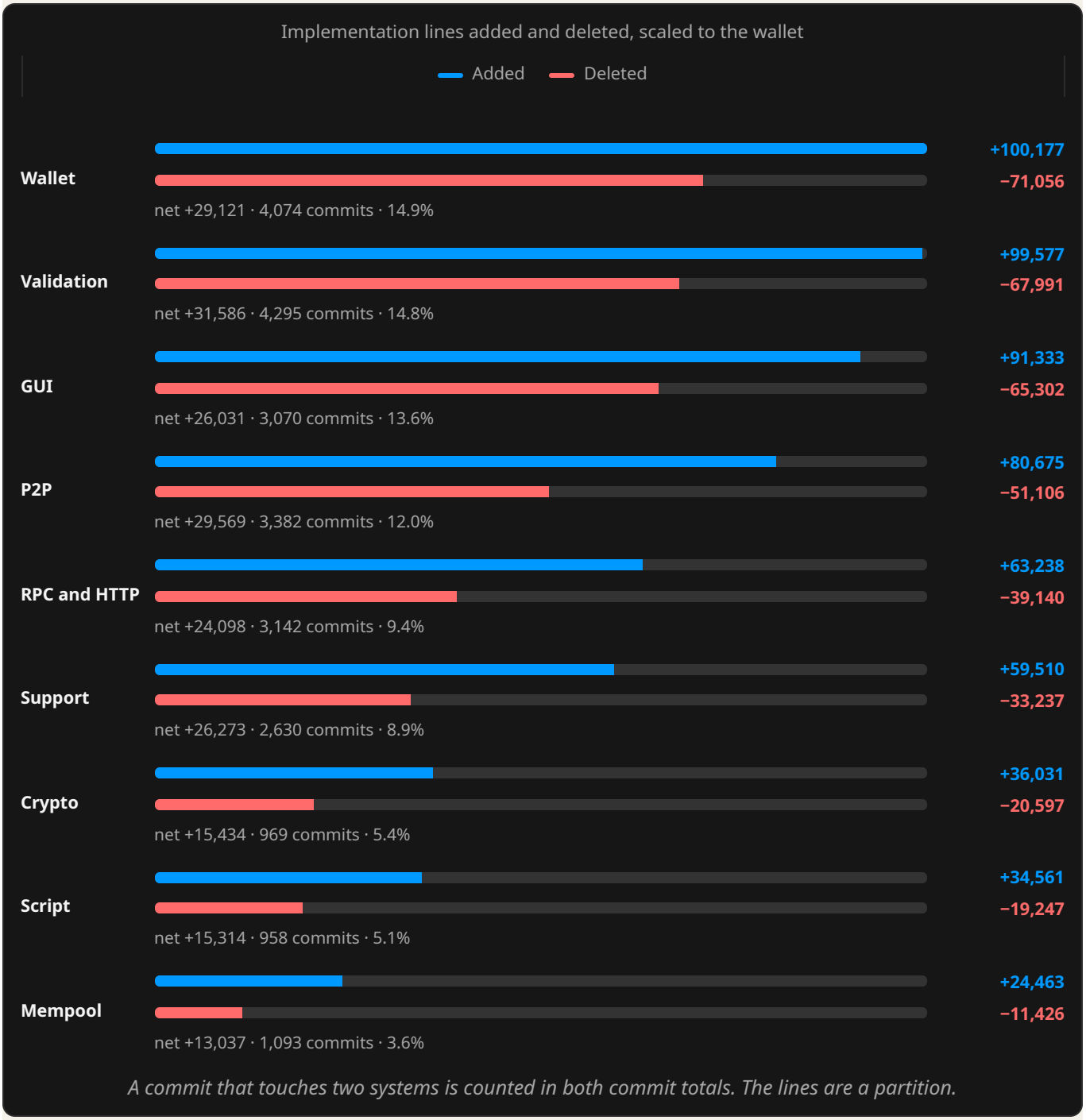
900 people, after dropping two bulk imports, sirius-m and Vinnie Falco. The top five added 39.5%. The top ten added 52.3%. Satoshi's row is the git string `s_nakamoto`.



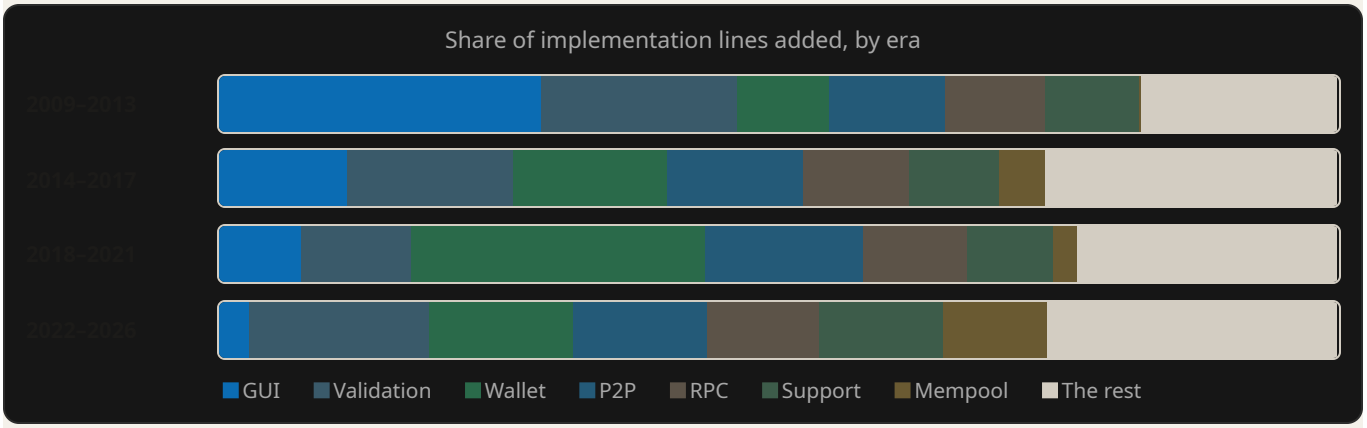
Michael Ford's small net skips build scripts. His merge share is a different ranking, in *Who Controls Bitcoin*.

Where the work went

The 55% figure above counts every filtered code file, including tests. Restricted to implementation files, and split with the same system lists as section III, **672,059** lines were added and **431,312** were deleted. **64.2%** of what was added was later deleted. No system has a majority. Wallet is the largest share of lines added, at **14.9%**. Validation is **14.8%**. The top four, wallet, validation, the GUI, and P2P, are **55.3%**.

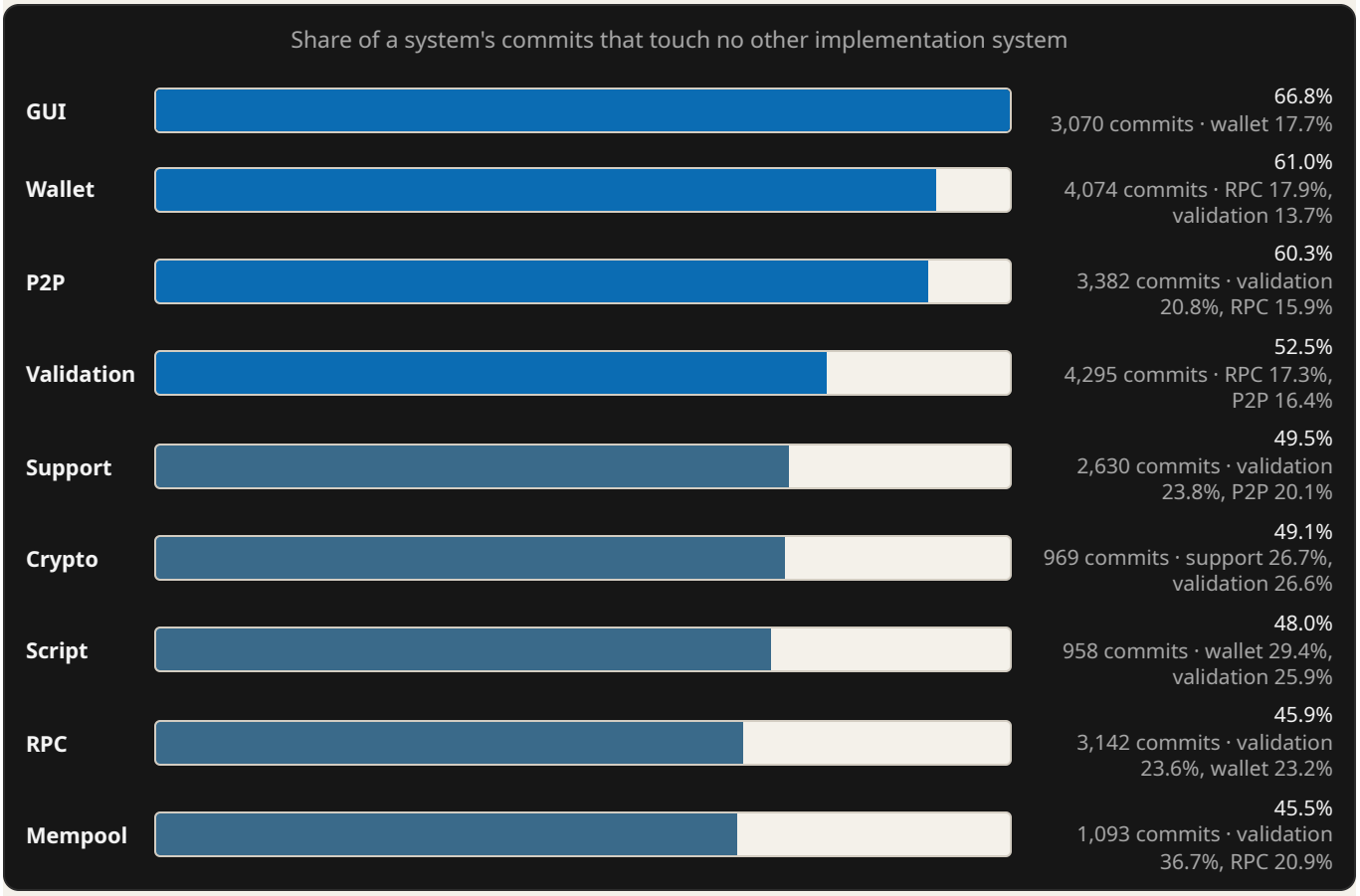


The early years are the GUI. In 2009–2013 the GUI is **28.7%** of implementation lines added. Wallet is the plurality in 2018–2021, at **26.3%**. From 2022 through 26 October, validation leads, at **16.2%**, and the GUI's net is **+261**. Wallet is the largest tree today because older lines survived.



What changes in the same commit

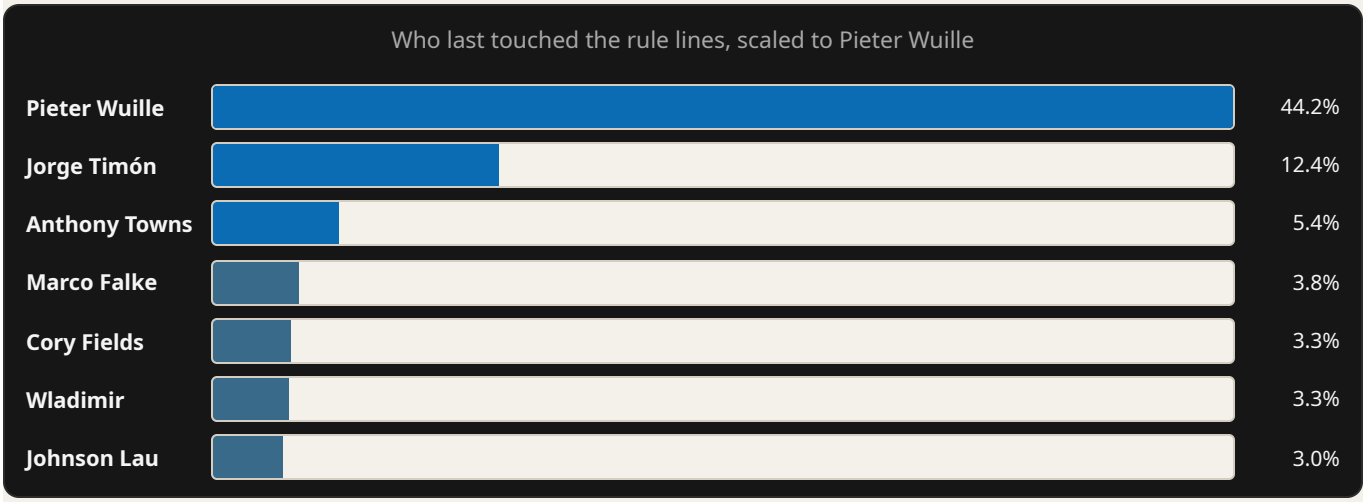
61.0% of wallet commits touch no other implementation system, and **13.7%** of them also touch validation. The tight pair is mempool and validation: **36.7%** of mempool commits also edit validation. RPC is the least separate. It is alone in **45.9%** of its commits, and those commits also touch validation **23.6%** of the time and wallet **23.2%**.

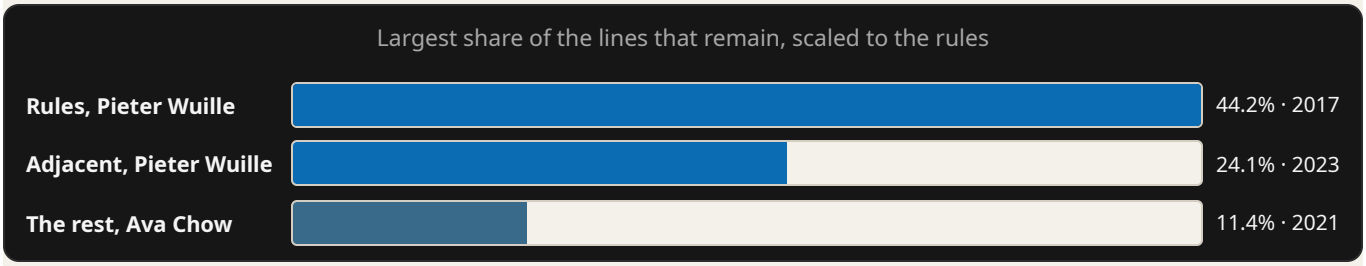


In the last three years the GUI fell to 47% alone. Wallet stayed near 68%, and validation near 62%.

IX. Who last touched what remains

`git blame -w` on the 756 implementation files. Each non-blank line goes to the last commit that changed it. `validation.cpp` is split on the same function boundaries as section IV. Comments count, so these totals sit above the tokei code lines: 12,734 non-blank rule lines against 10,122 code lines.

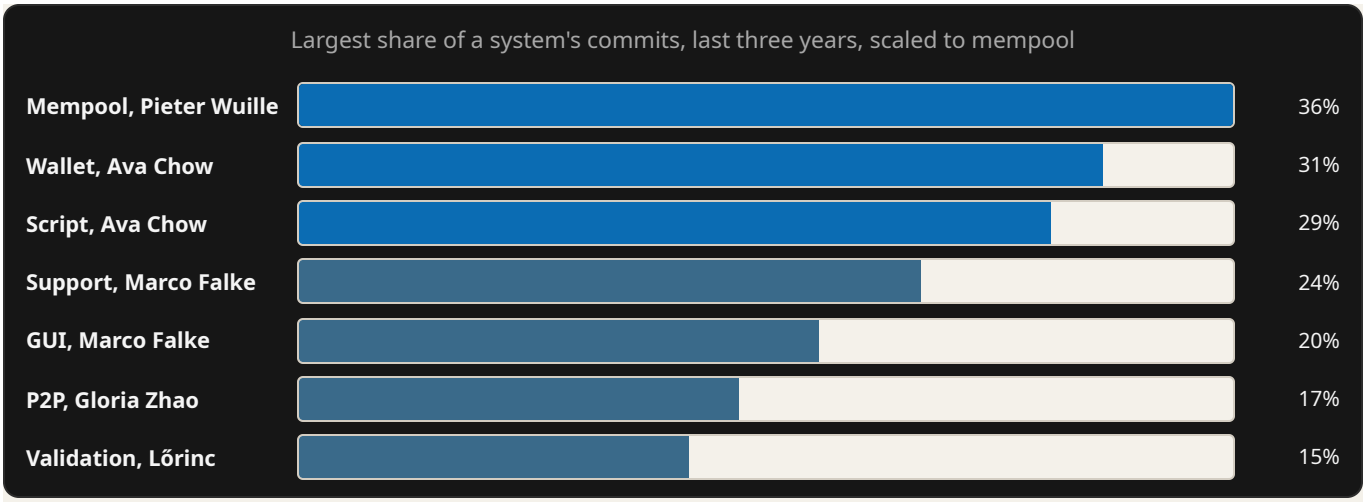


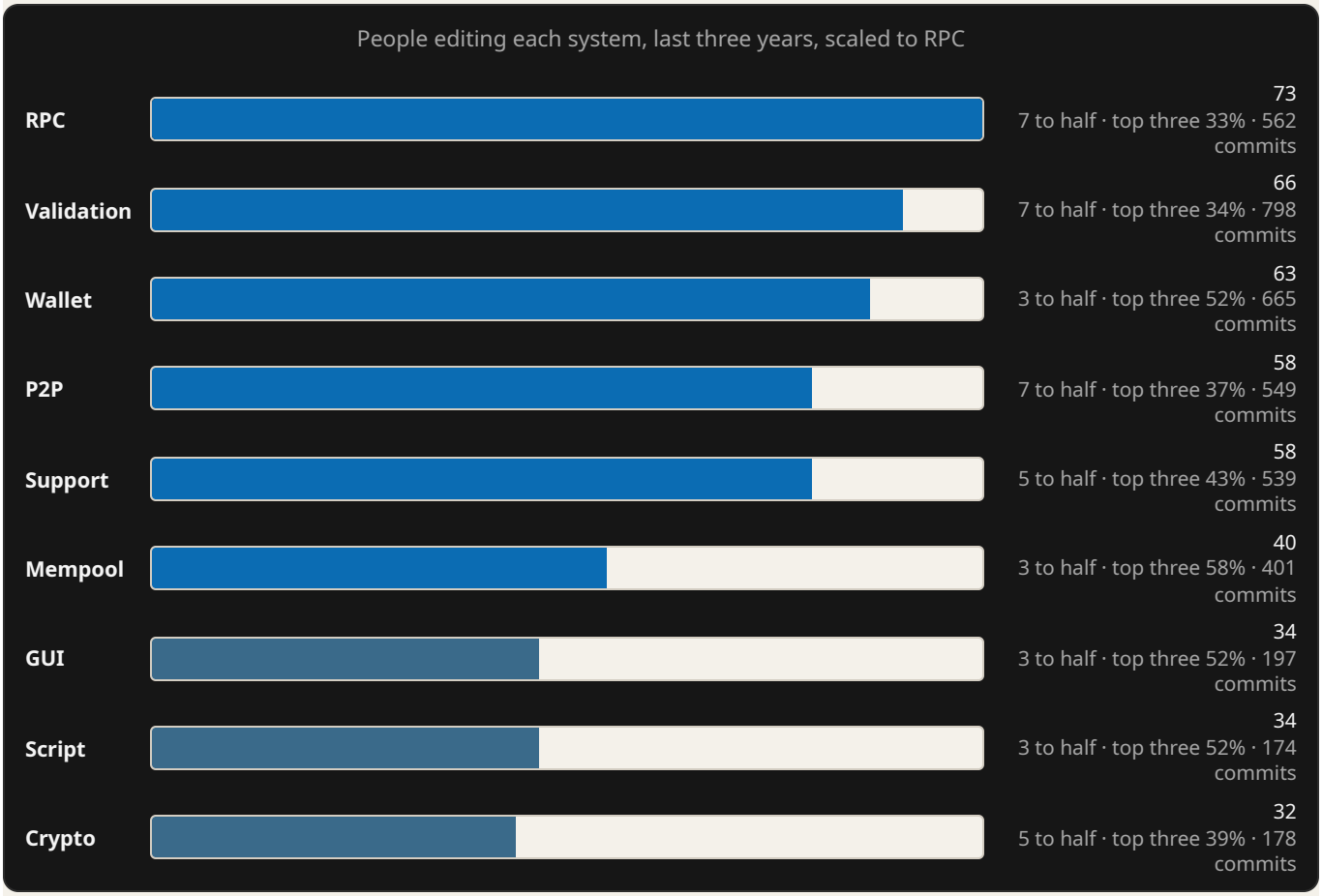


Pieter Wuille last touched 13.9% of all 187,516 non-blank implementation lines, against 44.2% of the rules. Satoshi's author string leads a large share of historical additions and is absent from the lines that remain. Who would make the next change is the next section.

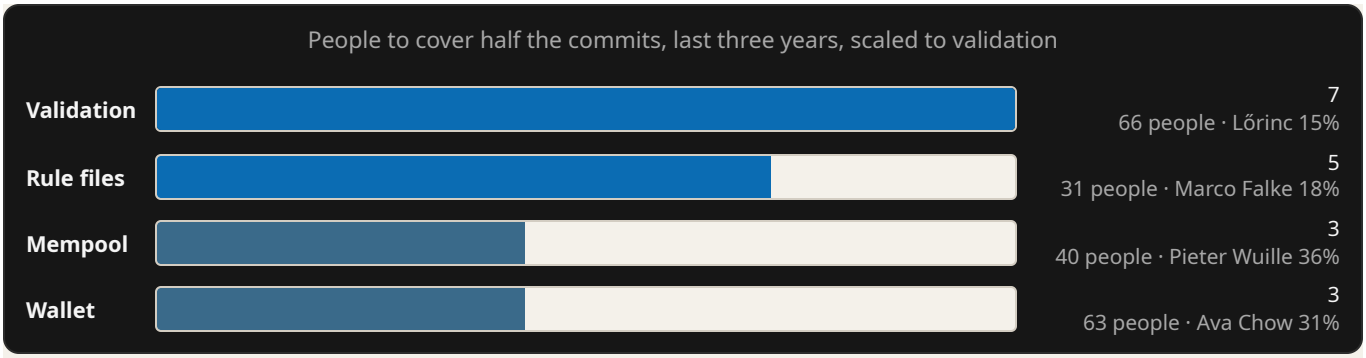
Who still edits a system

Over the three years ending 26 October 2026, validation's 798 commits come from 66 people, and seven of them cover half. Lőrinc is 15%. Marco Falke is 10%. TheCharlatan, including commits signed sedited, is 9%. Mempool is the concentrated system: Pieter Wuille is 36% and Suhas Daftuar is 12%, just under half together, and three people cover half. Wallet takes three people to reach half. Ava Chow, including commits signed Andrew Chow, is 31% of them.





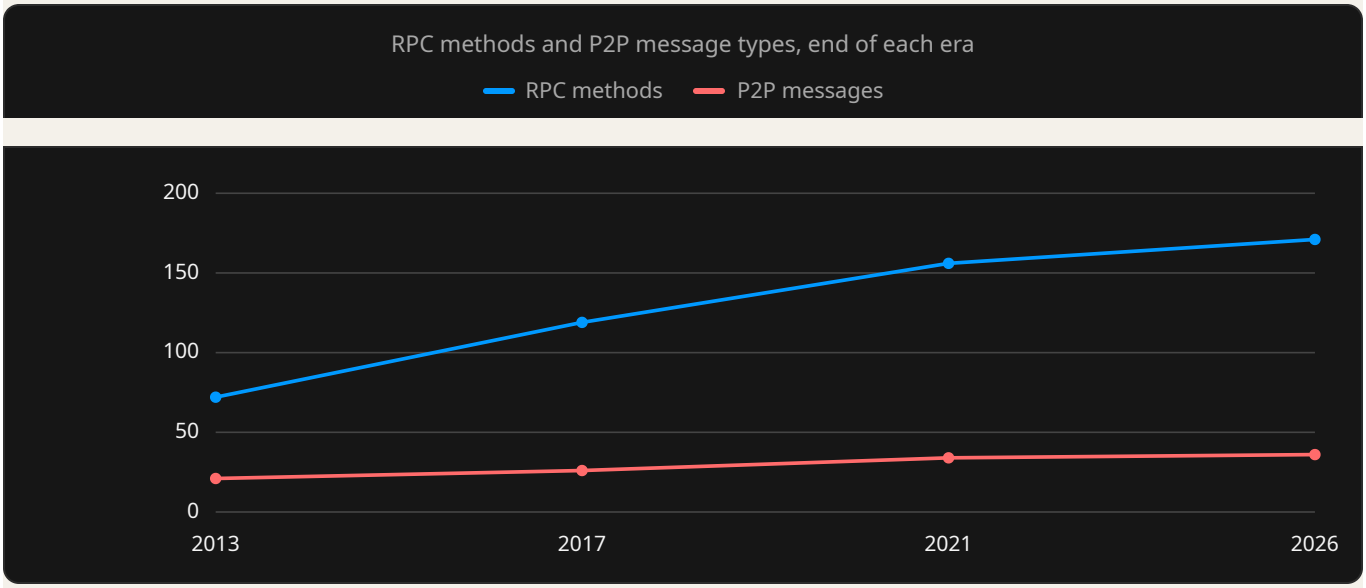
The rule files are their own count. In the same three years, **169** commits touch them, from **31** people, and **five** cover half. Marco Falke is 18%. Anthony Towns and Lőrinc are the next two, and the three together hold 42%. Pieter Wuille's 44.2% is the text that remains, and the median of those lines is 2017. These recent commits are a different list. The 562 check lines inside `validation.cpp` are left out of this count. That file is mostly chainstate and mempool.



Two git names that share an email are one person. RPC has the most people, 73. The GUI is the quiet tree, 197 commits.

The size of the contract

At this release the tree exposes **171** RPC methods, **36** P2P message types, and **5** named deployment slots. Named slots went 1, 5, 6, then 5. Taproot left the named list. The rule is still in the interpreter.

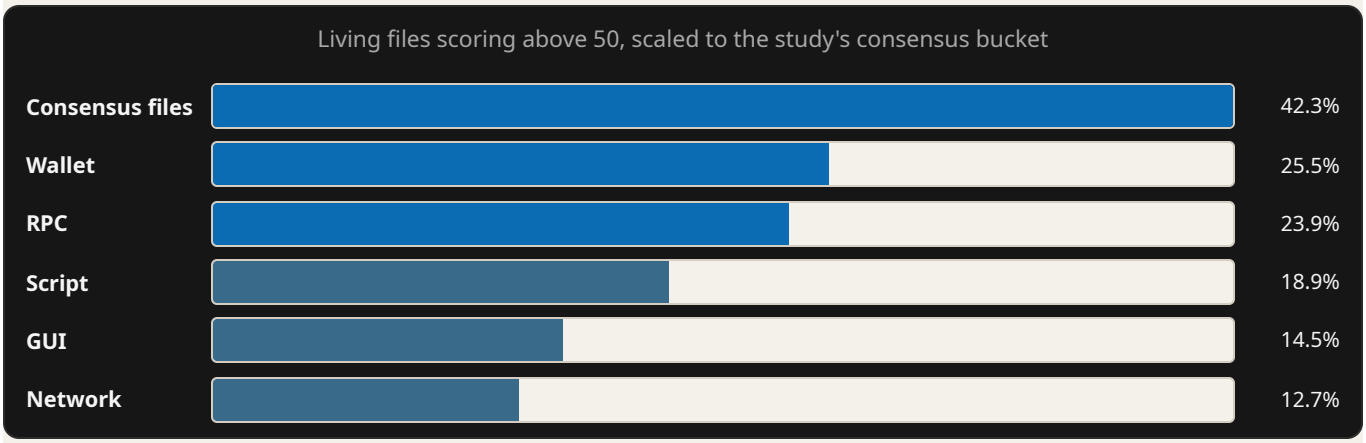


X. Patch debt

The January 2026 debt study scores a file from how often it is edited in small diffs and how long it has gone without a large rewrite. This is the same formula, rerun on **17,072** merged pull requests through 12 October 2026. It is a churn score. It is not a bug count.

Counting every path that history ever touched, including files that are gone, **28.2%** score above 50 and the patch-to-refactor ratio is **3.0**. RPC is 63% of that wider count.

Restricted to files still in this tree, **15.8%** of files score above 50, and those files hold **10.2%** of current code lines. High-scoring files are smaller than average. Many of the old high scores were paths that no longer exist (`main.cpp` , `ui.cpp`). Among living files the ratio is **3.6** patches per refactor. Wallet is **6.9** to 1. The study's own consensus bucket, 26 files, is patched once per refactor, and 42% of those files still score above 50. That bucket is not the 10,122 rule lines.



XI. What the count is for

The rules that have to be specified, audited, and matched by any other implementation are 10,122 code lines, plus a signature library that is larger than they are. The program operators download is the other 93% of the implementation, the tests, and the bundled trees. Treating a release as "the consensus rules" collapses those piles.

Don't Trust, Verify is about that collapse at the review layer. The first independent audit of Bitcoin Core, Quarkslab in September 2025, scoped P2P, mempool, block validation, and chain state. Those are mostly the adjacent lines in section V, plus some of the checks. The interpreter, the hash implementations, and the type of the transaction are the rest of the 10,122, and secp256k1 is another 76,503.

Who Controls Bitcoin measures who merges. This article measures who wrote lines and who last touched the lines that remain. The rankings differ. Merge concentration answers who can land a change. The blame in section IX answers whose text is still in the rule files. Both can be true, and neither is a substitute for the other. The later sections answer three further questions. Wallet commits also touch validation 13.7% of the time. Over the last three years, the rule files' 169 commits come from 31 people, and five cover half. Validation's wider tree comes from 66. A release exposes 171 RPC methods, 36 P2P message types, and 5 deployment slots.

Bitcoin Core: The Biggest Fallacies argues the client is a monolith because the wallet, the network, and the rules ship together. The line count is the size of that claim. Seven percent is the rules. Eighty percent is the rest of the same binary.

What Bitcoin's Stalled Proposals Tell You uses the authorship Gini from the governance research. That Gini is on pull-request authorship. The added-line shares in section VIII are a third cut. Top-five line additions at 39.5% and a historical authorship Gini near 0.85 are the same shape at two resolutions. They are not the same number.

The cost of storing the chain this program validates is a separate account, in *Full Cost of Running a Bitcoin Node*.

XII. Method

Snapshot. Bitcoin Core **32.1**, branch `32.x`, commit `f0de87f040621468b956f01f526a43a010310501`, 26 October 2026, "Merge bitcoin/bitcoin#36658: [32.x] Finalise 32.1". The tag matches the branch tip. Not shallow. 50,657 commits, 16,987 of them merges. Line counts are this release. Authorship is history through this commit.

Code lines. tokei 14.0.0. A file is in one bucket. Implementation is `src/` excluding vendored prefixes (`secp256k1`, `leveldb`, `crc32c`, `minisketch`, `univalue`, `libmultiprocess`, `ctaes`), Qt locale, tests, fuzz, bench, markdown, and `CMakeLists.txt`. The system split uses explicit file lists for `src/node`, because that directory is not all P2P. `src/node/mining_args.cpp`, `mining_args.h`, and `mining_types.h` are in mining templates.

Forks. leveldb in this release is `13da2d67` (11 August 2026). The shared ancestor with Google is `0c40829` (5 November 2019). The Google tip in the comparison is `bb74ef73` (8 October 2026). crc32c is `efb8ea04` (30 August 2024), ancestor `21fc8ef` (8 February 2022), Google tip `2bbb3be` (7 April 2025). The bars are git numstat and include comments and blanks.

The three-way cut. Rule files are the SHA-256 and RIPEMD-160 sources, `hash.cpp` and `hash.h`, the script interpreter and script data, `pubkey`, `arith_uint256`, `uint256`, `src/primitives`, `src/consensus`, proof of work, versionbits, deployments, and signet. `validation.cpp` is split by function start, comment-stripped, then scaled from 4,187 raw code-like lines to tokei's 4,247. Adjacent is chainstate and storage, policy and mempool, indexes, and mining templates, including the matching parts of `validation.cpp`. SHA-1, SHA-3, and SHA-512 stay in the rest. The three parts sum to 151,004.

Tests. The named rule unit and fuzz files are the earlier filename list (4,517) plus integer, hash, script, and block tests that name a rule type (1,403). Functional rule tests are an explicit basename list: `feature_block`, `feature_taproot`, `feature_segwit`, `feature_cltv`, `feature_csv_activation`, `feature_dersig`, `feature_nulldummy`, `feature_bip68_sequence`, `feature_signet`, `feature_versionbits_warning`, `p2p_segwit`, `p2p_invalid_block`, `p2p_invalid_tx`, `p2p_mutated_blocks`. Adjacent adds an explicit list of mining, assumeutxo,

pruning, index, fee, package, and chainstate tests to the mempool and policy filename prefix. A test outside those lists stays in the rest. That undercounts a functional test that exercises the interpreter under another name.

History. `git log HEAD --no-merges --numstat`. Renames of the form `{old => new}` are rewritten to the new path. Identity merges every pair of git names that share an email, and one name that used two addresses, plus the published aliases: Wladimir's three spellings, Ava Chow and Andrew Chow, Marco Falke and MarcoFalke, Gloria Zhao and glozow, Michael Ford and fanquake, jtimon and Jorge Timón. The tables use the release-note names. Satoshi Nakamoto's row is the `s_nakamoto` string. People who share only a first name stay separate. Ryan Ofsky and Russell Yanofsky are different people.

Systems over time. The same numstat history, implementation paths only, classified with the section III file lists, including the explicit `src/node` split. A path is classified by the name it had in that commit. Vendored trees and tests are out. The era chart uses that same partition in four spans: through 2013, through 2017, through 2021, and from 2022 through this release. Crypto, script, and the smaller systems are the remainder of each bar. A commit that touches two systems counts in both commit totals. "Only this system" means the commit touched no other classified implementation system. The three-year window is author dates from 26 October 2023 through this release. The people column merges git names that share an email, and the same published aliases as the history.

Rule-file editors. The same window and the same person merge. A commit counts once when it touches a file in the section IV rule list: the SHA-256 and RIPEMD-160 sources, `hash.cpp` and `hash.h`, the script interpreter and script data, `pubkey`, `arith_uint256`, `uint256`, `src/primitives`, `src/consensus`, proof of work, versionbits, deployments, and signet. `validation.cpp` is left out. Its 562 check lines share the file with chainstate and mempool.

The contract. RPC names are the command table when that table carries the method name (2013 and 2017) and `RPCHelpMan` / `RPCMethod` constructors when it does not (2021 and this release). Hidden methods are included. P2P names in 2013 are the message strings the network dispatcher in `src/main.cpp`, `src/net.cpp`, and `src/alert.cpp` receives or sends. From 2017 on they are the `NetMsgType` strings. Deployment slots are `DEPLOYMENT_*` enumerators other than testdummy, plus the block-version-2 height rule, with height-field aliases of the same fork collapsed to one name. Era tips are the last commit on this history at the end of 2013 (`aec55a07`), 2017 (`c9439e73`), 2021 (`1831d43e`), and this release.

Blame. `git blame -w --line-porcelain` on each implementation file. Whitespace-only changes do not move the line. Blank lines are dropped. The same alias list as the history.

Debt. `scripts/analysis/code_turnover_analysis.py` in the Bitcoin Governance Research repository, run on `data/github/prs_raw.jsonl` (25,506 pull requests, 17,072 merged with a date, last merge 12 October 2026). The score weights patch frequency, time since a large rewrite, age of files the script labels consensus, and modifications per estimated line. Change type is a size heuristic: a refactor is a change over 30% of the estimated file or over 200 lines. The script's subsystem labels are filename patterns. They are not the three-way cut in section IV. Living-file membership is the 32.1 tree.

Sources

- Bitcoin Core 32.1, commit `f0de87f040621468b956f01f526a43a010310501`, branch `32.x`, 26 October 2026
- tokei 14.0.0, file-level report on that tree
- Bitcoin Governance Research, technical debt analysis, formula rerun 30 October 2026 on the pull-request dump through 12 October 2026
- *Who Controls Bitcoin*, *The Vertical Layer Problem*, *Why Bitcoin Needs a Specification That Humans Can Read*, *Don't Trust, Verify*, *Bitcoin Core: The Biggest Fallacies*, *What Bitcoin's Stalled Proposals Tell You*, *Full Cost of Running a Bitcoin Node*

SECURESOVEREIGN
Markdown is the canonical source: https://secsov.com/articles/the-rules-are-7-percent/index.md
Formatted snapshot of https://secsov.com/articles/the-rules-are-7-percent
Josh · Secure Sovereign