2023 Brink Annual Report

Our generous sponsors enabled Brink to fund seven Bitcoin Core developers as of the end of 2023, including allowing many of them to work together in a shared office in London. They work in public on open source software, so anyone at any time can see all that they've accomplished, but in the following pages we summarize their major achievements for 2023 and share what they're excited about working on for 2024.

We focus on the quantity and quality of each engineer's code review and testing before we look at any other contributions they made. This is a deliberate decision. We frequently hear that many open source contributors worry that they will only receive offers if they pursue headline-grabbing projects, rather than quietly performing the high-quality code reviews and testing that are essential to keeping Bitcoin Core secure and getting useful changes merged quickly. As an organization, Brink always puts review and testing first and foremost, and we've tried to reflect that part of our internal culture in this external report.

The end of this report also includes financial information about Brink, information about the work performed by our executive director and our operations manager during 2023, and details about the make up of our board and our grant committee—everything you'd typically expect to find in a charitable organization's annual report. But Brink's primary mission is to find amazing Bitcoin developers and fund them on behalf of our sponsors, and we couldn't be more excited to start this report by showcasing all that they've accomplished in 2023.

2023 developer highlights

These are the highlights. For details about each developer's work, see later in this report.

Sebastian Falbesoner left over 300 review comments as a part-time engineer, many of them focused on version 2 encrypted peer-to-peer transport. He also began contributing the the libsecp256k1 cryptographic library used by Bitcoin Core and several other security-focused projects.

Michael Ford continued his role as project maintainer, leaving almost 1,600 review comments, merging an average of 11 pull requests a week, and releasing every 2023 version of Bitcoin Core. He helped lead several efforts to modernize Bitcoin Core's build toolchain.

Niklas Gögge left over 500 review comments, publicized several responsible disclosures he had made of serious vulnerabilities, found multiple new bugs in Bitcoin Core pull requests before they were merged, significantly extended Bitcoin Core's testing infrastructure, and made several safety-focused changes to Bitcoin Core's code.

Fabian Jahr left over 200 review comments after joining Brink mid-2023, led major improvements to distributed generation of ASMap files, wrote proof-of-concept code for batch validation of schnorr signatures, researched cross-input signature aggregation, and added significant resiliency to Bitcoin Core's code hosting.

Hennadii Stepanov left almost 1,300 review comments, contributing greatly to the project's effort to modernize its build tool chain, and continued his role as the project's GUI maintainer. He also continued working with the Bitcoin Design Community on a reference GUI wallet design compatible with Bitcoin Core.

Stéphan Vuylsteke left almost 600 review comments, earning special commendation from his peers for his diligence and follow up. He continued contributing to multiple education and mentorship efforts, including Qala, London BitDevs, and the Bitcoin Core Pull Request Review Club.

Gloria Zhao continued her role as P2P system maintainer, leaving 900 review comments and leading the work on package relay. She was also a leading contributor to TRUC (v3) transactions and cluster mempool. She gave multiple talks at Bitcoin conferences, co-maintains the Bitcoin Core Pull Request Review Club, and helps mentor several new Bitcoin Core contributors.

Sebastian Falbesoner

Brink has been funding Sebastian since 2021 for his part time review-focused work. During the year, he left over 300 review comments, most of them focused on pull requests for BIP324 version 2 encrypted peer-to-peer transport—a protocol improvement that greatly increases user privacy and can also improve security.

On behalf or our sponsors, Brink is pleased to be able to continue funding Sebastian for a third year (all of 2024) and to provide him an open-ended offer to upgrade to a full-time engineer role either this year or the beginning of 2025.

Review and testing

> *Fortunately, the BIP324 project progressed very smoothly last year (faster than I personally expected) and v2 transport protocol support is available since release v26.0, as default-off option. It will be default-on with the next release v27.0.*
>
> *I helped reach this goal by providing thorough code reviews and testing both for the PRs implementing the cryptographic primitives (#27985, #27993, #28008), the actual changes in the network layer (#28165, #28196) and the final signalling/integration (#28331). In parallel, I reviewed the corresponding changes in the functional test framework (#24005, #26222, #28374, #24748).*

Adding support for encrypted peer connections has long been on the wishlist of many developers, and contributors to the Bitcoin Core project voted it as a high priority for the 26.0 release. Although the original Bitcoin protocol uses cryptographic functions such as signatures and hash digests, it did not use encryption, so an encryption algorithm needed to be selected, added to Bitcoin Core's cryptographic library (libsecp256k1), and then integrated into Bitcoin Core. Further, the peer-to-peer (P2P) protocol needed to be upgraded to allow negotiating encrypted connections in a way that was backwards-compatible with older unencrypted connections. Several optimizations were also made for upgraded connections at the same time.

All of that required careful review. In most software, connection problems lead, at worst, to frustrated users. But Bitcoin nodes that can't reliably connect to peers may be vulnerable to problems such as eclipse attacks. Additionally, a failure to correctly implement encryption may put users at risk of an unexpected privacy loss.

Sebastian carefully reviewed pull requests that affected Bitcoin Core's P2P code, its tests, and its libsecp256k1 dependency. Other contributors told us how much they appreciated his fast reviews and re-reviews, and how that helped the project to maintain momentum and achieve the goal of being included in 26.0. As we're writing this report, we are only days away from BIP324 encryption being enabled by default in the upcoming 27.0 release—a testament to the work of many different Bitcoin Core contributors, including Sebastian.

Press:

- Brink Renews Grant for Sebastian Falbesoner's Review of BIP324 to Enhance Bitcoin's Censorship-Resistance - **Bitcoin Magazine ¶**
- Bitcoin Core adds optional support for v2 encrypted P2P connections - **Bitcoin Optech Newsletter #272 ¶**
- Bitcoin Core #29347 enables v2 P2P transport by default - **Bitcoin Optech Newsletter #288 ¶**

Libsecp256k1

> *In the course of reviewing the EllligatorSwift [key negotiation] part of BIP324, I dug deeper into libsecp2561 and started contributing there. Several PRs were opened in the categories of improving documentation (#1340, #1341), various refactoring and cleanups (#1339, #1357, #1393), but also adding exhaustive test coverage for secp256k1-ellswift (#1371), and some mild per-*

formance improvements by tightening group magnitude limits (#1344, #1348).

The libsecp256k1 library is the dependency Bitcoin Core and many other Bitcoin-focused programs rely on for multiple cryptographic operations. Many developers find it to be a very intimidating piece of software. Sebastian not only took on the challenge of reviewing code there this year in order to help advance the BIP324 project, but he continued contributing there afterwards. We at Brink are very excited to see this: it's hard to find contributors to libsecp256k1 and it's essential to the long-term safety of Bitcoin that we have a number of experienced contributors.

Press:

- Libsecp256k1 #1129 implements the ElligatorSwift technique for establishing v2 P2P connections - **Bitcoin Optech Newsletter #257** ¶

Plans for 2024

> “*Silent Payments is the project where I want to invest most of my energy in the next grant period; the only change is that the focus shifts exclusively on the libsecp256k1 side of it right now, as any work on Bitcoin Core directly depends on that.*

Silent payments, originally proposed by Ruben Somsen and co-championed by Josie Baker, is a new type of Bitcoin address that can be reused for multiple payments without creating any link between those payments onchain. This is a significant improvement over most existing Bitcoin addresses where sending multiple payments to the same address creates a strong link between them that reduces the privacy of the spender, the receiver, and the people who later receive payments from the receiver.

Silent payments requires wallets perform more work than is required for most address types, but it's mostly work that full nodes already perform, so it makes particular sense for an initial silent payment implementation to be made for Bitcoin Core's wallet. If it's successful and people begin using it, silent payments could easily spread to other wallets just as other recent innovations have spread from Bitcoin Core's wallet to other wallets (such as PSBTs, miniscript, and output script descriptors).

Silent payments does require a different type of cryptography than is used in Bitcoin Core today, so some of the work to add support for it is being done in Bitcoin Core's cryptographic library, libsecp256k1. As noted in a previous section, Sebastian has already built a strong familiarity with this library, so he's well poised to help make and review further improvements.

> “
>
> *My other priorities for the year will be package relay and legacy wallet removal.*

Package relay, in its initial planned version, will allow two related transactions to be considered as a package rather than as two independent transactions. This enhances the child-pays-for-parent (CPFP) fee bumping mechanism already used on Bitcoin by allowing even very-low-feerate parent transactions to get confirmed alongside a high-feerate child.

CPFP fee bumping is critical to the security of several time-sensitive contract protocols, such as Lightning Network. Certain Lightning transactions have to be created and signed long before they're broadcast. If they're created with a high feerate but feerates are low when they're broadcast, the user will lose money by overpaying fees. If they're created with a low feerate but feerates are high when they're broadcast, nodes will discard them in order to prevent excess use of their memory, making it impossible to fee bump them with CPFP. Package relay avoids that problem and makes Lighting and other contract protocols more secure by allowing a low-feerate transaction to be bundled with a high-feerate child transaction, evaluating them as a group under the CPFP rules.

Bitcoin Core's legacy wallet dates (in part) back to the original Bitcoin 0.1 code release in 2009. Wallets since then have evolved in many ways, including the adoption of BIP32 HD wallets and Bitcoin Core's recent switch to descriptor-based wallets. Currently, Bitcoin Core supports both old (legacy) wallets and newer (descriptor) wallets, with a set of tools that will help a user convert from legacy to descriptors. The plan is to drop legacy wallet support in a future release. Anyone who still has a legacy wallet will still be able to run an old release (offline if desired), convert to a descriptor wallet, and use that descriptor wallet with a current Bitcoin Core release.

Adding support for package relay and carefully removing legacy wallet code will require diligent reviews, which Sebastian has proven time and again is something he can provide.

Michael Ford

A Bitcoin Core contributor since 2012 and project maintainer since 2019, Michael is one of Brink's most senior developers. In 2023, he provided almost 1,600 review comments and helped lead the effort to modernize Bitcoin Core's build system. He was also the release manager for every Bitcoin Core release in 2023. He's often the first to comment on a new pull request, and he's been praised by colleagues for his "encyclopedic knowledge of everything that's happening in the project," allowing him to help coordinate disparate efforts across the project.

On behalf or our sponsors, Brink is pleased to be able to continue funding Michael's essential engineering work.

Review

> *A lot of my reviews are just churning through uninteresting day-today PRs/changes. For example, minor code refactors, CI fixes, documentation changes etc.*

A major software project consists of a million moving parts that each occasionally wear down, fall out of alignment, or need to be replaced with a newer version. Almost nobody ever writes or talks about that low-level of background maintenance, but if it ever stops being performed, the whole machine will quickly start to tear itself apart and no further progress will be made.

150 years ago, the person responsible for maintaining all of the parts of a moving train engine was called an engineer, and we don't think anyone earns the title of *software engineer* more than people like Michael who get their hands dirty every day by quietly taking care of all the live maintenance that a major open source project needs to be successful. The public will almost never hear about an engineer who does their job well, but engineers like Michael are an essential part of what makes possible every major accomplishment in the Bitcoin Core project that you do hear about.

Maintainer and release manager

> *I merged 60% (586/962) of the PRs to Bitcoin Core last year. This involves having a birds eye view of the entire project day-to-day. Keeping tabs of what has been reviewed, and by who, what is ready for merge (and what it may conflict with), and what the merge order of changes should be.*
>
> *All Bitcoin Core releases were put together by me last year (23.2, 24.1, 25, 25.1, 24.2, 26.0). I spend about 20-30% of my time making sure the projects continues to release software to Bitcoin Core users.*

The Bitcoin Core project has a rule that each pull request needs to be receive several "ACKs" from qualified reviewers before it gets merged and the code is changed for the next release. It's up to maintainers to review the reviews of pull requests and decide if a pull request has received high-quality review and if it's in line with the project's philosophy and goals.

This can be an especially thankless job. Advocates for a pull request will complain that maintainers are taking too long to merge, while critics of the same pull request will complain if they think it was merged prematurely. Many arguments between well-meaning advocates and critics come down entirely to difficult-to-compare sets of tradeoffs, often in cases where not merging a pull request is just as much of an expressed choice as choosing to merge it.

But even for unexciting pull requests, being a maintainer takes a significant amount of time away from a developer's own projects. Being a maintainer is a commitment to working on other people's projects, even people who are sometimes unappreciative of the maintainer's time and effort.

Through these incredible stats—over 11 careful merges per week, every week, for all of 2023, plus release management for two major and four minor releases—we again see Michael's commitment to helping other developers achieve their goals.

> *The project got numerous bug/security issue reports (multiple emails a day) and fixed those bugs in major and minor releases. Generally I consider this a success, because:*
>
> - *There were no major incidents involving the network, throughout the year.*
> - *As far as I'm aware, none of those bugs have leaked/been discovered.*
> - *No issues remained untriaged, although some have not yet been patched in a release.*

In addition to his regular maintainer duties, Michael serves on Bitcoin Core's security subproject where he helps triage and quietly resolve bugs that affect user safety and security. As quoted above, he describes this as a "general success", although we think Bitcoin Core's security track record can currently be qualified as an *extraordinary success*.

Toolchain modernization

> “ *Good progress to modernise our toolchains, but still work to do. LLD work is almost done, now mostly just stalled on Qt (the GUI). GCC release compiler upgrade is blocked on determinism issues. This just needs more time spent. C++20 was nice, and we are already seeing the benefits of that migration. CMake work has progressed to a point where it's usable, but is still blocked on final architecture decisions.*

Bitcoin Core is not only compiled software but software that aims to be compiled deterministically—meaning everyone who compiles a release version should be able to obtain identical executables. This is essential to minimizing trust in the release manager or any particular developer. Anyone who compiles the software themselves can verify that the executables being served from BitcoinCore.org are identical, ensuring users are receiving the actual code that's been so painstakingly reviewed and tested.

However, Bitcoin Core's determinism and need to run on a wide range of platforms brings a variety of challenges. The build system is complicated and changes that seem like simple improvements in one area can cause confusing problems in other areas. Additionally, Bitcoin Core needs to continue to support some code that was originally designed 15 years ago, such as code for the legacy wallet.

Bitcoin Core's build system only works well because of continued contributions by Michael and a handful of colleagues who quietly work on this area of the code that almost never receives any publicity.

Plans for 2024

> “
>
> *Linux release compiler upgraded to GCC 12 or 13, finish the macOS LLD migration (completing a 3-year project that allows us to drop Apple's semi-open source tools entirely, simplifying our own toolchain, and removing reliance on poorly maintained upstream projects), move Windows builds from GCC to Clang, get fully static release builds done, CMake migration completed (probably for Bitcoin Core 29.x), and initial Rust integration into Bitcoin Core (somewhat of a pipedream, but the migration of the project towards Rust, would seem inevitable).*

In additional to continuing as a maintainer and frequent release manager, Michael has multiple ambitious for further modernizing and improving Bitcoin Core's build tools. Several of these changes improve safety by replacing custom code and tools with standard libraries and tools.

Want to work with Michael? Brink is looking to hire an additional full-time build-system engineer in 2024, preferably someone to work directly with Michael in our London office. Compensation amount is based on experience and qualifications. We will pay for any required work visa. Apply at https://brink.dev/programs

Niklas Gögge

Niklas began contributing to the Utreexo proposal before graduating in March 2022 and joining Brink full time. Since then, he's focused on quality assurance: reviewing, writing tests, and refactoring failure-prone code. In 2023, he left over 500 comments on pull requests and, in early 2024, he disclosed multiple vulnerabilities in the btcd full node and the LND Lightning node, all of which he had previously discovered through testing, responsibly disclosed, and kept private until fixes were widely deployed, keeping users safe and minimizing disruption.

On behalf of our sponsors, Brink is pleased to be able to continue funding Niklas for all of 2024.

Review

> “ *In a lot of instances, some review is completely automatable. Specifically, the PRs for which no new harness needs to be written can be fuzzed automatically. I created a tool that can automatically build a given branch, fuzz a given harness, minimize the corpus, run the corpus through all sanitizers, and finally create a coverage report. I am using this for PR review and continuous fuzzing of the master branch. I am notified about any crashes on a private GitHub repository. At the moment, this infrastructure is limited to two machines (one beefy one for fuzzing and one small one for hosting the coverage reports), which Brink is paying for. This has freed up a lot of my time to focus on other non-automatable work.*

Reviews from Niklas covered almost all areas of Bitcoin Core. Many of his reviews were test-based: he examined the pull request to see if the proposed code change came with sufficient testing and whether the tests were being run. If new tests were needed, Niklas often contributed them. Reviewers frequently thanked Niklas publicly and several developers we spoke to privately indicated that positive reviews from Niklas were a strong signal that a pull request was high quality and had a good approach, encouraging other developers to contribute reviews and accelerating the progress of those pull requests towards getting merged.

In at least four pull requests (#28948, #28685, #28578, and #29242), Niklas used tests to find bugs before they were merged. In particular, #28685 is a fix for what could have become a serious bug that was caught by fellow Brink engineer Fabian Jahr (with diagnosis of the underlying issue by fellow Brink engineer Sebastian Falbesoner). After discovery of the bug, Niklas wrote a fuzz testing harness that found two additional serious bugs and allowed them to be fixed at the same time.

Press:

- Bitcoin UTXO set summary hash replacement - **Bitcoin Optech Newsletter #274** ¶
- Bitcoin Core #28685 fixes a bug in the calculation of the hash of a UTXO set - **Bitcoin Optech Newsletter #275** ¶

Fuzz testing net processing

> “ *In my opinion, improving our testing is the most important and impactful thing to work on. It would reduce our review bottleneck, it would make our code easier to change, and it would cause less frustration among developers.*

Fuzz testing takes code that was designed in the head of a programmer who probably expected an ideal situation and exposes it to variety of semi-randomly-generated input to see what happens. In a poorly designed program, exposing it to unexpected input can lead to crashes, memory leaks, wrong states, disclosure of private state, and many other problems. In mission-critical software like Bitcoin Core, any of those problems can quickly become a major vulnerability: a crash of a full node can allow theft of funds from a related Lightning Network wallet; a memory leak can escalate to a crash; wrong states can lead to accepting invalid transactions; and disclosure of private state can lead to theft.

Before fuzz testing was used with Bitcoin Core, developers were already on the cutting edge of using defensive programming techniques to avoid the worst problems, but many problems have still found their way into the code (and more probably exist). Fuzz testing allows developers to leverage cheap and abundant CPU and memory to test functions with massive numbers of random inputs to see if something fails.

Niklas has successfully used fuzz testing to discover multiple bugs that could have led to the loss of bitcoins if an attacker had discovered them first. Most Bitcoin Core developers are supportive of increased fuzz testing, however only about 67% of Bitcoin Core's net processing code is currently covered by fuzz tests. Niklas worked tirelessly during 2023 to try to improve that percentage, including maintaining multiple pull requests through multiple annoying rebases.

Press:

- Transcript of fuzzing presentation by Niklas Gögge - **Bitcoin Transcripts ¶**

Security engineering

> *Every now and then security focused work pays a dividend by turning up security relevant bugs in production code. Disclosure highlights the importance of security focused work and those who fund it: Brink.*

Niklas had 16 pull requests merged in 2023 (and several more in early 2024) that helped improve the safety of Bitcoin Core. Some notable pull requests include:

- #29412 drops mutated blocks received over the network as early as possible. This class of problems stems from a weird choice for the original consensus-enforced design for Bitcoin's merkle tree and has led to multiple serious vulnerabilities in the past (e.g. CVE-2012-2459 and 2019-merkle). Addressing the issue proactively lowers the risk of future vulnerabilities.

- #28956 removes the use of *adjusted time* from consensus code (without requiring any actual consensus changes). Adjusted time is another weird feature that was included in very early versions of Bitcoin and potentially creates more problems than it solves.

- #28460 significantly sped up many fuzz tests and #28480 fixed an issue where some fuzz tests weren't actually being run.

- #29064, #29219, #29031, #28558 increased the stability of fuzzing harnesses, which has already resulted in the discovery of undefined behavior.

Bitcoin Core's fuzzing corpora repository is maintained by Niklas. The corpora is used by the project's continuous integration tests and regularly finds bugs before they get released to users.

> *New fuzzing techniques and research is constantly published. We should explore the use of new tools and techniques to ensure our bug finding capabilities are as good as they can be. As of now, none of the [new] techniques have uncovered any bugs but they all widen the space of bugs we could find. I keep some of this work in private repos until I am reasonably confident it can't be used to trivially find new bugs.*

Niklas experimented with multiple new fuzzing tools, looking for ways to speed up fuzzing, increase coverage, and automatically find discrepancies between different code paths that are supposed to be identical (e.g., code running on different computer architectures). He's also used fuzz testing on Bitcoin projects beyond Bitcoin Core, resulting in the discovery of multiple serious vulnerabilities that were all responsibly disclosed. After being fixed, multiple vulnerabilities were able to be publicly disclosed in 2023 and early 2024:

- Two LND gossip handling vulnerabilities: an attacker could exploit the first bug to crash LND, leading to it being unable to send time-sensitive transactions, potentially allowing the attacker to steal from the user. The second bug could be exploited by an attacker to prevent an LND node from learning about certain channels, potentially forcing the user to forward payments partially or entirely across the attacker's channels, reducing the user's privacy and allowing the attacker to collect additional routing fees from the user. After Niklas's private disclosure, both bugs were quietly fixed; they weren't revealed until after users already had other security-related reasons to upgrade.

- Btcd consensus vulnerability: an attacker could make the btcd full node accept a block with a transaction that every Bitcoin Core full node would consider invalid. As Bitcoin Core nodes are predominate on the network, btcd users would mistakenly believe the block was valid and that the transactions in it were confirmed. This could lead to miners who use btcd losing money creating invalid blocks, Lightning Network users losing funds due to a false view of the blockchain, and onchain users believing transactions were confirmed when they weren't. After Niklas's private disclosure, the bug was quietly fixed. Later, shortly after Niklas's public disclosure, the bug was exploited on testnet, with the fixed version of btcd able to handle the bug.

- Core Lightning invoice parsing vulnerabilities: after discovering several vulnerabilities in CLN in 2022, Niklas submitted the fuzzing harness he had used to discover the bugs in 2023. Matt Morehouse realized the fuzz tests had actually discovered three additional vulnerabilities and further expanded them to find two more vulnerabilities, for a total of five vulnerabilities, all related to how CLN handled invoices. Three of the the vulnerabilities could be used to crash CLN, which can be a serious problem for an automated Lightning Network node due to it needing to be able to respond quickly in some cases to protect user funds. One of the bugs allowed the use of uninitialized memory which, in theory, could have allowed an attacker to extract private information from the node—although CLN keeps most of its private data in a separate process to help minimize such issues, so an actual attack would have been quite difficult. All of the vulnerabilities were fixed and none are known to have been exploited.

Press:

- Disclosure of past LND vulnerabilities - **Bitcoin Optech Newsletter #283** ¶
- Discussion with Niklas Gögge, discovered of LND vulnerabilities - **Bitcoin Optech Podcast #283** ¶
- Disclosure of fixed consensus failure in btcd - **Bitcoin Optech Newsletter #286** ¶
- Discussion with Niklas Gögge, discoverer of btcd vulnerability - **Bitcoin Optech Podcast #286** ¶

Plans for 2024

> *I want to continue building out the differential fuzzing engine. I want it to be a general tool for differential fuzzing, not just for fuzzing across different architectures. For example, we could continuously differentially fuzz the latest version of the script interpreter against old versions of itself (as a hard-fork sanitizer).*

Comparing Bitcoin Core against versions of itself, whether for different platforms or different versions, has the potential to automatically discover the types of vulnerabilities that have been among the worst yet seen in Bitcoin. For example, CVE-2018-17144 was accidentally introduced in Bitcoin Core versions 0.14 and 15.0, allowing the same bitcoins to be spent more than once; it's possible that differential fuzzing between version 0.13 and 0.15 could have discovered that bug. Another example would be the vulnerability fixed by the BIP66 soft fork: the old signature verification library used by Bitcoin Core (OpenSSL) would accept some signatures as valid on some platforms while considering them invalid on other platforms, which could have been used to split the network; differential fuzzing between those platforms could have discovered that vulnerability.

> *I want to research/experiment with Bitcoin Core specific fuzzing feedback. For example: our current mempool harnesses are great at achieving high coverage in the mempool code but they do not (for example) manage to successfully submit large transaction clusters to the pool. A custom mempool feedback, focusing on maximising "cluster size" (or other graph complexity metrics) could guide the fuzzer to bugs arising from complex mempool structures. The idea is to speculate on where bugs might be and try to guide the fuzzer to them through application specific feedback.*

Bitcoin Core currently depends on very talented and careful developers thinking through all possible ways a particular piece of code might be used. Fuzz testing can't entirely replace that, but it can allow us to throw money (in the form of CPU cycles) at checking that a developer's logic about what will happen matches the reality in millions or billions of different permutations, increasing our confidence that the analysis is correct. This is especially useful given the limited amount of time available from high-quality reviewers. It's much easier to scale up testing than it is to find, train, and retain new developers.

> *Something I want to get working next year is smarter scheduling for continuous fuzzing. It currently uses simple round-robin scheduling over all harnesses, i.e. each harness gets N hours of CPU hours every day. A smarter scheduling algorithm could take code changes (between revisions of Bitcoin Core) and coverage reached by each harness*

into account, to prioritise scheduling harnesses that reach the changed code. This will also be helpful to efficiently automate fuzzing of PRs, since it avoids fuzzing harnesses that are unrelated to the code changes being tested. A second goal is to allow scheduling of fuzzing jobs across multiple machines (it currently only supports one machine). This will allow us to scale up the infrastructure as needed.

Again, Niklas is working on scaling up Bitcoin Core's testing capabilities. This has the potential to find existing bugs in the code, allowing them to be fixed quietly. It also increases the chance that new bugs will be found before they can be exploited. Improved automatic testing of pull requests will speed up development as it allows developers to address many issues in their code before it's ever seen by reviewers. This makes the work more meaningful for reviewers and makes pull request authors happy by reducing the number of re-review cycles necessary to get good code merged.

> “ *More bug disclosures. There is one Bitcoin Core bug that I found that will be up for disclosure this year (pending discussion with the other devs). There are also a few disclosures I have written up for older Bitcoin Core bugs that were never disclosed (publishing these will be discussed at the next CoreDev meeting).*

Publicly disclosing vulnerabilities that have been fixed makes it easier for new developers to learn from those old vulnerabilities. It also demonstrates the work that Bitcoin Core developers are quietly performing in order to keep users safe, and helps motivate users to upgrade to new versions (even if just the oldest maintained version of Bitcoin Core).

Niklas has also asked to attend conferences and training related to cutting-edge fuzzing techniques, which Brink will be proudly sponsoring him to attend.

Want to work with Niklas? Brink is looking to hire an additional full-time test engineer in 2024, preferably someone to work directly with Niklas in our London office. Compensation amount is based on experience and qualifications. We will pay for any required work visa. Apply at https://brink.dev/programs

Fabian Jahr

After a long history of volunteer part-time contribution to Bitcoin, Brink began funding Fabian full time in April of 2023. During the remainder of the year, he left over 200 review comments on PRs related to assumeUTXO, multiprocess, and package relay. He also led major improvements to distributed generation of ASMap files to improve Bitcoin Core's denial-of-service protection, wrote proof-of-concept code for batch validation of schnorr signatures to speed up block verification while simultaneously researching cross-input signature aggregation (CISA), and automated the process of moving Bitcoin Core's repository from third-party GitHub to a self-hosted GitLab instance.

On behalf of our sponsors, Brink is pleased to be able to continue funding Fabian full time for all of 2024.

Reviews

> “ *I focused my review on larger projects, and particularly I think my review helped get AssumeUTXO merged into Bitcoin Core. This can be seen in my review statistics above and also in this bug, which I discovered during testing post-merge.*

Nearly a quarter of Fabian's review comments in 2023 were left on Bitcoin Core pull request #27596, “assumeutxo (2)”. This code contribution by James O'Beirne “finishe[d] the first phase of the assumeutxo project. It [made] UTXO snapshots loadable via RPC.” AssumeUTXO has been a top priority for the Bitcoin Core project as it makes a new node on fast hardware fully functional for everyday users within a matter of minutes, rather than hours. This massively lowers the barrier of entry to a user validating their own transactions with their own node—a fundamental requirement if we want Bitcoin's consensus rules to remain in control of everyday people.

Over the five months the AssumeUTXO pull request was being reviewed by developers, it received 448 comments—over 10% of which came from Fabian, who reviewed it and re-reviewed it multiple times. Fabian also found the above-mentioned bug, #28685, which could have led to security problems, and he helped review multiple AssumeUTXO follow-up pull requests, helping the project progress.

Press:

- Bug found in UTXO set summary hash - **Bitcoin Optech Newsletter #274 ¶**
- Podcast with Fabian Jahr, discoverer of UTXO set summary hash bug - **Bitcoin Optech Podcast #274 ¶**
- Bitcoin Core #27596 finishes the first phase of the assumeutxo project - **Bitcoin Optech Newsletter #272 ¶**

ASMap

> *Amazingly, what we have now actually shouldn't even be working according to experts I talked to (network engineers from ISPs, CDNs, IRRs, etc.), so in some regards what we have now is even better than what I had hoped for a year ago.*

ASMap is a project started within Bitcoin Core in 2019 to improve node resistance against cheap denial-of-service attacks, including dangerous eclipse attacks. A full node is only fully secure if it connected to at least one honest peer. If a malicious entity controls a large number of IP address across a wide range of the major subnets, it's possible that some nodes could make all of their connections to that malicious entity, allowing it to censor blocks from the most-proof-of-work blockchain and instead provide blocks that well-connected nodes would mostly ignore.

Entities that directly control blocks of IP addresses are called autonomous services (ASes) and a map to all of them (ASmap) could allow Bitcoin Core to ensure it connected to a variety of different ASes. This wouldn't prevent an eclipse attack based on collusion between different entities, but it would make eclipse attacks and other denial-of-service attacks more difficult than they are today.

Creating an ASMap is challenging for an individual, but even doing that isn't enough for Bitcoin Core. The project rightfully doesn't want to trust information from a single individual. Because IP addresses are frequently traded from one AS to another, there's no reliable way for Bitcoin Core contributors to verify an ASMap that a single contributor created in the past. Instead, Fabian worked to develop tools for a mapping process that could be run by multiple contributors independently in parallel. If they all obtained the same result, that would allow the project to use the resulting map without trusting any individual person.

Professional network engineers who frequently work with ASes and routing tables didn't think a distributed parallel process could work well enough for multiple people to all obtain the same results—but Fabian made it work through the development of new tools and processes. This may allow Bitcoin Core to begin including a default ASMap file in future releases, improving safety and reliability for everyone operating a full node.

Press:

- Improved reproducible ASMap creation process - **Bitcoin Optech Newsletter #290 ¶**
- Discussion with ASMap contributor Fabian Jahr - **Bitcoin Optech Podcast #290 ¶**

Schnorr batch verification and CISA research

> *While simultaneously getting started learning more about CISA I reviewed and rebased the secp256k1 code for schnorr batch verification, had some discussions, and experimented with the integration into Core. The proof of concept code is available here and some conceptual discussion has happened based off of it there and in libsep256k1.*

One of many advantages of schnorr-style signature verification discussed prior to the activation of taproot is the ability to batch verify multiple signatures at the same time. For example, if a typical block consisted entirely of taproot transactions using schnorr signatures, it would be possible to verify all of those signatures simultaneously about twice as fast as verifying each of them independently.

Fabian started his work this year by reviewing a previous experimental module for libsecp256k1. He later rebased the code and, in early 2024, opened a draft PR to Bitcoin Core that begins performing batch validation. His work has already received several review comments.

> *I realized that CISA was just very far off from batch validation in*

> *terms of conceptual work that was still needed.*

Validating multiple signatures together, described above, seems similar to combining multiple signatures together, which in Bitcoin is called cross-input signature aggregation (CISA). Two types of aggregation are known: half aggregation would allow a transaction with multiple inputs to only include a single full signature (about 16 vbytes) and half a signature (about 8 vbytes) for each additional input, reducing the size of transactions generated by a single user or a group of cooperating users (such as in a coinjoin) by about 8% in a typical case (or about 14% in the best case). Full aggregation would allow a transaction to contain only a single full signature no matter how many inputs it had, reducing its size by about 16% in a typical case (or about 40% in the best case).

Either type of CISA would require a soft fork and both types may conflict with other proposed soft fork upgrades. Much more research on the topic is required, some of which Fabian conducted this year. He began maintaining CISAResearch.org, which contains a collection of education about the topic, and created a proof of concept implementation in Python for half aggregation.

GitHub alternative

> “*Based on my initial testing on my own GitLab server before my grant application, I thought this would be a matter of a few days. Unfortunately, the sync between GitHub and GitLab was unstable and several approaches did not work as expected. Paid support from GitLab was not very helpful and their open source community also didn't help in a meaningful way. Our main contact person there also left the company in the meantime.*
>
> *It was almost a strike of luck that I started trying to use one of their professional service tools for the migration task and started contributing there with PRs and issues. Through a conversation in one issue I was able to find a configuration that allowed the sync to finally work.*

Although it's easy to obtain a complete copy of every revision to Bitcoin Core's code using Git, the discussions behind those code changes are all stored on a single centralized platform: GitHub. In the past, GitHub has ceased hosting popular open source projects due to government requests and their own policies. They've also had persistent bugs on their platform that have slowed development of Bitcoin Core, including website optimizations that increased the risk that important code feedback might not be seen by reviewing developers before safety-critical code was merged.

For many years, several contributors have been hosting backups of Bitcoin Core's GitHub issues and PRs in case there was a problem. That ensures critical context isn't lost. But if GitHub were to suddenly delist the Bitcoin Core project, it could potentially take weeks or months until the project was able to restart on a different platform and import all of the context needed to continue development at the same pace as before.

One of Fabian's projects this year was figuring how to minimize that gap. He's created a set of scripts that frequently backs up the Bitcoin Core repository and creates a self-hosted version using the open source GitLab software. In the event of a problem with GitHub, the project can begin using the GitLab version in a matter of days (at most)—with every existing user being automatically issued their own GitLab account and every issue and pull request being updated to its latest state as of the backup. He achieved this goal early in 2024 and is continuing to maintain it in case it is ever needed.

Press:

- GitLab backup for Bitcoin Core GitHub project - **Bitcoin Optech Newsletter #292 ¶**
- Discussion with GitLab backup contributor Fabian Jahr - **Bitcoin Optech Podcast #292 ¶**

Other activities

> “*I regularly get good feedback from participants who started to contribute to Bitcoin projects because of it.*

Fabian is an organizer for the Berlin Bitdevs meetup, which can have up to 50 attendees. He's been giving presentations at conferences, such as BTC23 in Innsbruck, and helped organize the Bitcoin Core developer meeting in early 2024.

Plans for 2024

> “*I feel like I have only recently gotten into a position where I can help drive faster development of CISA, given what I have described above. I want to continue the work on CISA in 2024 with more focus.*

He hopes to continue working with cryptographers such as Jonas Nick and Tim Ruffing as they develop a scheme for full signature aggregation and develop a signature proof for it. Even if full aggregation remains elusive, he hopes to create a BIP proposal for half aggregation that can be extended to full aggregation if that later becomes available.

> “ *I want to continue to support the implementation of Silent Payments because I think it is rare that a non-softfork proposal has seen so much excitement and only very little push-back. I also think that my experience is helpful, given it interacts with secp256k1 and looks to leverage indices as well.*

Fabian joins fellow Brink engineer Sebastian Falbesoner in helping to support the implementation of silent payments for Bitcoin Core. As we mentioned in Sebastian's section of this annual report, silent payments provides a privacy-enhanced reusable address format that works especially well for users of full nodes, making it a natural fit to be implemented first in Bitcoin Core before spreading to other wallets.

> “ *I will put together a new tracking issue for AssumeUTXO mainnet params deployment and will champion for more focused review so that we can finally let users take advantage of this feature.*

Fabian plans to continue his work on AssumeUTXO, helping to bring it to 100% support and allowing new users of full nodes to start using them for receiving transactions potentially within minutes of installation. This will arguably be the largest single improvement in the usability of a full node in Bitcoin's history. We're excited that Fabian is working with other developers to help get it across the finish line.

> “ *I would like to continue maintaining/championing these projects: batch validation, default ASMap, GitLab code hosting, and BitDevs and other educational events. None of them are completely done but many of them have now entered a different phase that allows me to focus more time on other topics, such as dedicating more time to review overall.*

We've already mentioned the importance of the projects that Fabian worked on last year, and we're happy to see him continue to maintain them and move the ones that can be completed a little closer to eventually being included in Bitcoin Core. Given the high quality of the reviews he's been able to provide, we're especially excited to see him able to devote more time to reviews in 2024.

Hennadii Stepanov

Hennadii has been contributing to Bitcoin Core since 2018 and became the project's GUI maintainer in 2021. Brink began funding him in 2021 and helped relocate him and his family from Ukraine to the United Kingdom in 2022. During 2023, he left almost 1,300 review comments and helped spearhead the initiative to modernize Bitcoin Core's build system from autotools to CMake. He also works closely with the Bitcoin Design community in the creation of a QML-based reference GUI that's compatible with Bitcoin Core and demonstrates user-interface best practices.

On behalf or our sponsors, Brink is pleased to be able to continue funding Hennadii's work on both the high-level GUI and various low-level systems.

Reviews

> “ *As a code reviewer, I was mostly focused on the following topics: the build system (including depends and Guix), CI tests, the libbitcoinkernel project, and the GUI.*

Nearly every review from Hennadii starts with him actually running the code, something other reviewers sometimes skip but which occasionally reveals bugs that are obvious to a human but hidden from the automated testing infrastructure. This is especially useful when testing GUI changes and changes to the build system, which are difficult to completely automatically test.

GUI maintainer

> *As a maintainer, I was responsible for: Bitcoin Core GUI repository, Bitcoin Core translation project on Transifex, Translations-related steps in the release process for every release, and Bitcoin Core QML GUI repository.*

The Bitcoin Core project has experimented with the *monotree* development model used by the Linux Kernel project. The Bitcoin Core GUI repository is maintained separately from the project's main repository. Periodically, Hennadii pulls changes from the main repository into the the GUI repository. Similarly, when changes are made in the GUI repository, they are synced back into the main repository. This allows developers focused on GUI development to subscribe to the GUI repository and ignore changes to the main repository, or vice versa for developers interested in the main project but not the GUI.

In addition to performing these periodic syncs, Hennadii is responsible for triaging all new issues and pull requests to the GUI repository, and he often ends up not just triaging new issues but also fixing them.

As the only one of Bitcoin Core's maintainers that isn't a native speaker of English, he also generously agreed several years ago to manage the translation process that allows Bitcoin Core to be available in dozens of languages.

Bitcoin Core's current Qt-based GUI is a 12 year old re-implementation of the original wxWindows GUI used in Bitcoin 0.1. It's been upgraded many times to support new features, but Hennadii and several members of the Bitcoin Design community have been working a new version based on the QML design language. The goal is not only to upgrade Bitcoin Core's GUI but also to offer a reference design for other Bitcoin software that provides a wallet and can operate a node.

CMake

> *The CMake reviewing process is still happening in a dedicated staging branch with an extensive set of CI tests. Its current progress is approximately 85% (57 reviewed commits, 80 reviewed pull requests) for now. The collaboration involves seven active reviewers: Cory Fields, Sebastian Kung, Michael Ford, Vasil Dimov, Aaron Clauson, Max Edwards, and Pablo Martin [and Hennadii makes eight]. We have weekly Google Meet calls.*

Hennadii has been working with multiple other contributors on a long-term project to convert Bitcoin Core from GNU autotools to the modern CMake build system. This will not only significantly simplify the build system, it will also unlock additional benefits, such as the ability to upgrade Bitcoin Core's GUI to the latest version of Qt, the cross-platform widget library it uses.

In addition to his work directly on the Bitcoin Core side of the build system, Hennadii also contributed CMake code to the libsecp256k1 project, which allowed it to immediately access some of the improvements available with CMake and will simplify integrating it with a CMake version of Bitcoin Core in the future.

Plans for 2024

> *This next year I plan to work on migration to the CMake-based build system; reviewing pull requests that interest me, namely: build system, libbitcoinkernel, cluster-based mempool (linearization and feerate diagrams); getting rid of Boost.Process and re-enabling external signer support on Windows; hardware-accelerated SHA256 implementations for native Windows builds; and removing recursive mutexes from the codebase.*

Improvements to the build system will continue allowing Bitcoin Core to run on a variety of platforms, ensuring that almost anyone running any variety of modern hardware and operating system will continue to be able to run a full node. In particular, Hennadii is one of the Bitcoin Core developers most focused on ensuring Bitcoin Core remains fully functional on Windows, a platform with a large number of less-technical users who still want to use Bitcoin without needing to trust any third party and who can still contribute to enforcing Bitcoin's consensus rules by validating their own transactions with their own node.

Stéphan Vuylsteke

Stéphan joined Brink part-time in 2022 while continuing to provide education and mentorship through Qala. He now works almost full time for Brink, providing reviews of Bitcoin Core pull requests. He continues to help new developers through office hours with Qala, hosting the London BitDevs, and co-maintaining the Bitcoin Core PR Review Club.

On behalf or our sponsors, Brink is pleased to be able to continue funding Stéphan's careful reviews and ongoing educational initiatives for another year.

Reviews

> *I try to spend a good chunk of my time on PRs (on various parts of the codebase) that are quite self contained, important to the project, and have good enough momentum to reasonably be able to be merged. I try to focus on going in-depth while providing quick re-review to prevent the PR from losing momentum.*

Almost 600 review comments were left by Stéphan in 2023. One of his peers remarked, "something I really appreciate about Stéphan, which is somewhat rare in Bitcoin Core: if he reviews something, he does re-reviews very quickly, and he sticks with it until it's merged. Afterward, he'll open followups to fix remaining issues, write tests, etc. If you tag him for review on areas of the code he is familiar with, he is very reliable."

> *For the release of v26, I have focused mostly on the libbitcoinkernel project. This was quite a new area for me to explore, requiring me to dig quite deep for each PR.*

Bitcoin Core's libbitcoinkernel sub-project, originally championed by Carl Dong and extensively worked on by several other developers, has been carefully refactoring Bitcoin Core's code to separate consensus logic from any other code. In an idealized form of the Bitcoin system, each full node verifies every block of transactions using exactly the same consensus rules, all of them coming to exactly the same conclusion about whether the block is valid or not. This allows each node, acting independently from every other node, to come to consensus without depending on voting or any other process that's vulnerable to manipulation.

This idealized process only works if every node actually does use *exactly* equivalent consensus code. When they don't, serious bugs can result—for an example, see the section of this report that describes Niklas Gögge's discovery of a vulnerability in the btcd alternative full node. Ensuring that different versions of Bitcoin Core remain compatible with each other makes it imperative to ensure that any change to consensus code is obvious to developers (and to users who audit the developers' work), and that the changes are well reviewed. The libbitcoinkernel sub-project makes this easier by working to clearly separate consensus code from other code used for other purposes.

The separation of consensus code from other code also requires the creation of a clean interface between the two, which may eventually allow other programs to interface with the exact consensus code Bitcoin Core uses. For example, it could be possible for future versions of btcd to validate new blocks both with their own code and with libbitcoinkernel, allowing them to warn their users when an incompatibility was discovered, rather than risking falling out of consensus.

Despite the long-term importance of the libbitcoinkernel project, it's always a challenge to attract and retain attention on refactoring-type projects that don't deliver any immediate benefits for users and rarely receive headlines. That makes it especially useful to have reviewers like Stéphan whose fast reviews and re-reviews help pull requests maintain momentum and get merged without becoming stale.

Educational projects

> *Since my previous grant application in May 2023, my educational efforts have mostly revolved around: hosting the monthly London BitDevs, with an attendance of usually*

> *at least 15-20 people every month; become a project co-maintainer of the Bitcoin Core PR Review Club alongside Gloria [Zhao]; built VS Core to help new developers or experienced "drive-by" contributors quickly get up and running with a Bitcoin Core development environment; contributed to various discussions about how to improve open source education curricula, mostly stemming from my experience with and focusing on Qala; helped Max Edwards with the v26 RC Testing guide.*

When Stéphan first came to Brink, part of his time was spent working on Bitcoin developer education with Qala. He's maintained a strong commitment on helping educate new developers, which has always been a great fit with Brink's focus on increasing the capabilities of open source contributors.

The Bitcoin Core Pull Request (PR) Review Club meets online regularly to help less experienced developers gain experience reviewing a current or recently-merged Bitcoin Core pull request. More experienced developers such as Stéphan are available to answer questions and suggest how reviewers can make the most of their effort.

Stéphan also created a project this year that allows anyone who uses the VS Code integrated development environment to automatically set up a build environment for working on Bitcoin Core in an isolated container. Normally, creating a build environment requires carefully following a detailed setup guide. Stéphan's VS Code extension makes it much easier for new contributors to get a little experience with Bitcoin Core, or for already experienced open source contributors to simply fix an issue themselves (and hopefully submit the fix as a pull request) rather than opening an issue that some other developer will need to reproduce and fix.

Plans for 2024

> "*I plan to largely continue contributing like I have in 2023, focusing on one or two priority projects that require deeper knowledge and continued commitment. Current projects that I'm interested in are multiprocess, package relay, and cluster mempool, but most importantly I try to focus on where my help is best used—which is difficult to predict in advance*

Stéphan's ability provide fast and comprehensive review will remain in high demand in 2024. We will describe in detail the importance of work on package relay and cluster mempool in the section of this report about Brink engineer Gloria Zhao, but we will briefly note here that those projects that Stéphan plans to focus on are also the top two projects voted on by Bitcoin Core contributors for the version 28 release planned for late 2024. They will definitely benefit from as much review as possible.

Multiprocess is a project championed by Russell Yanofsky for many years now, and it's one that many contributors continue to strongly support. The goal is to clearly separate the operation of Bitcoin Core as a node, as a wallet, and as a GUI. In other words, each of those parts of Bitcoin Core should be able to run in a separate process, with them communicating with each other over a clearly defined interface (API).

If such an API becomes stable, it should be possible for external software to substitute for any part. For example, instead of using the default GUI, you could use an alternative GUI (for example, the QML GUI being worked on by Brink engineer Hennadii Stepanov), allowing users of that alternative GUI to continue benefiting from Bitcoin Core's predominant node implementation and highly-reviewed wallet. Alternatively, an experimental new wallet could be substituted for Bitcoin Core's default wallet.

Multiprocess may eventually allow the node, the wallet, and the GUI to be split into completely separate repositories to be maintained by different sets of developers, each with their own development priorities. This is similar to how other aspects of Bitcoin that were originally handled directly by Bitcoin Core (e.g. proof-of-work hashing) are now developed separately and managed using external tools. This separation of concerns can help node developers focus on providing the best possible node—a node that's as easy as possible for everyone in Bitcoin to run.

As a project that's important long term, but never urgent, multiprocess is always in need of additional reviewers that can help give pull requests momentum towards being merged. Whether on multiprocess, package relay, or cluster mempool, we're excited to see how much critical review Stéphan will be able to perform in 2024.

Gloria Zhao

Gloria joined Brink in January 2021 as the first member of our one-year Fellowship program, just a few months after she started contributing to Bitcoin Core. She became a Brink engineer in 2022 and Bitcoin Core maintainer for the P2P subsystem in the same year. In 2023, she provided 900 review comments and was the driving force within the Bitcoin Core project for *package relay*, an upgrade to the P2P protocol that will significantly enhance the security of contract protocols such as Lightning Network. She's also been one of the lead developers on TRUC (also called *version 3 transaction relay*) and cluster mempool, which aim to solve additional security-related problems for contract protocols.

On behalf or our sponsors, Brink is pleased to be able to continue funding Gloria's ongoing work to improve the Bitcoin P2P network.

Review and maintainer duties

> “*When reviewing other peoples' PRs, sometimes I found bugs, wrote tests, or provided some insight from a bigger picture understanding. I don't always find bugs, but I think my reviews are helpful—I try to focus on making progress on high priority things, even if they require understanding complex areas of the codebase.*

As maintainer of the P2P subsystem, Gloria commented on almost every substantial pull request affecting that part of the code. She also left comments on many pull requests outside of P2P, including multiple pull requests by promising new Bitcoin Core contributor Abubakar Sadiq Ismail, who has been working on improving fee estimation.

Many of Gloria's review comments in 2023 were on her own pull requests, which are some of the most high-impact pull requests that were opened to Bitcoin Core in 2023. She replied to hundreds of comments from other reviewers, quickly acknowledging and addressing their concerns. Without her commitment to timely and professional replies, we doubt she would have been able to get pull requests with such large scope merged.

Press:

- The Role of Bitcoin Core Maintainers & the Path Forward - **Bitcoin 2023 Conference (Bitcoin Magazine)** ¶

Package validation and P2P package relay

> “*My original plan last year was to get package validation done first, and implement the p2p changes. There was some thought to do them in parallel, but my main collaborator (Greg Sanders) and I quickly realized it was much more productive to focus on one track at a time.*
>
> *Championing a large project also involves convincing others to prioritize and review it. Bitcoin Core contributors have continuously voted package relay as the highest priority project for every release cycle, so I think I have been somewhat successful in that effort.*

When Alice spends bitcoins to Bob and then Bob spends those bitcoins to Carol, Bitcoin's consensus rules requires Alice's transaction appear earlier in the blockchain than Bob's transaction. This ensures Bob can only spend bitcoins he owns; he can't spend on credit. Alice's transaction is the *parent* of Bob's *child* transaction. If both of their transactions are currently unconfirmed, then a miner can only include Bob's transaction in a block if they also include Alice's transaction in the same block. This dependency created by Bitcoin's consensus rules provides a mechanism for fee bumping, called child-pays-for-parent: if Alice's transaction pays a low feerate—too low for a miner to want to include it in a block—then Bob can choose a high feerate for his transaction, and the weighted average of both Alice's parent transaction and Bob's child transaction can be high enough for a miner to profitably mine both transactions in the same block.

CPFP fee bumping is a powerful tool for contract protocols, such as Lightning Network, where certain transactions are created and signed long before they are broadcast, preventing the signers from knowing what feerate to use. With CPFP, the pre-signed transaction can use a low feerate and whoever receives money from the transaction (e.g. Bob) can simply create a child transaction with an appropriate feerate to get both the pre-signed parent and the high-feerate child included in the next block.

However, Bitcoin Core full nodes have a limited amount of memory to use for storing unconfirmed transactions, so they don't store transactions with a feerate below the amount necessary to get confirmed within the next day (approximately). That means a low-feerate parent transaction by Alice might not be stored in a node's mempool, resulting in the node rejecting Bob's high-feerate child transaction because its parent is unavailable.

The two-part solution for this is allowing a *package* of related transactions to be both relayed and validated together. In *package validation*, Alice's parent transaction and Bob's child transaction are validated together and it's their weighted average feerate that determines whether they are able to enter the mempool. For nodes to effectively use package validation on the network, they need to know when receiving Alice's parent transaction not to validate it as an independent transaction—that they should wait for Bob's child transaction to arrive and that they should run package validation on the both transactions together; this ultimate objective is called package relay.

Although these changes are relatively easy to describe, they represent a significant difference from the way Bitcoin Core has been independently validating each transaction since its initial release in 2009. Because package relay will be used in a security sensitive way by contract protocols such as Lightning Network, it's critical that it be carefully evaluated for any potential problems.

Zhao's continuing work on package relay has included not just writing code for Bitcoin Core but also writing a draft BIP and engaging developers working on Lightning Network to ensure that the ultimate project meets their needs and expectations.

Press:

- Gloria Zhao - v3 Transactions and Package Relay - **Stephan Livera Podcast ¶**

Addresing pinning and RBF problems with v3 and cluster mempool

> “*This is very much tied to package relay, as we're trying to answer a common set of questions. They also share some code, dependencies on one another, and the same small group of reviewers.*
>
> *In 2022, while I was gathering information about RBF and pinning, one major issue identified was "we don't have a good way of assessing or enforcing incentive-compatibility in mempool," which can lead to undesirable behavior in replacements and evictions. Fortunately, this problem nerd-sniped some very smart folks who have been working on a better way to assess incentive compatibility and rewriting the mempool to address these underlying issues.*
>
> *Now, this project takes the form of v3 and cluster mempool (mostly review). As I'm writing this, we've pretty much finished v3 and are starting to review/merge things for cluster mempool.*

Package relay solves one problem for contract protocol developers—how to pay fees on transactions signed long before they're broadcast. Another problem occurs when more than one party to a contract attempts to pay fees. To prevent wasting the resources of full node operators, such as memory and bandwidth, Bitcoin Core has to limit the number, size, and flexibility of the transactions it accepts, but a malicious party to a contract can sometimes abuse those rules to prevent honest parties from getting a transaction confirmed in a timely manner, called transaction pinning. Pinning is especially a problem for contract protocols such as Lightning Network where safety sometimes depends on time sensitive transactions.

Gloria researched and helped design one solution to many current transaction pinning attacks: *topologically restricted until confirmation* (TRUC) transactions, also known as *version 3* (v3) transactions. This allows the creators of transactions to opt-in to a set of restrictions that limits the severity of pinning attacks by 99% or more without creating any problems for honest users.

Gloria has also worked with several other Bitcoin Core contributors, including Suhas Daftuar, Pieter Wuille, and Greg Sanders on a ground-up redesign of Bitcoin Core's mempool to make it easier to ensure each new transaction added to the mempool strictly improves the set of transactions available to be mined. This has long been the goal of Bitcoin Core's mempool policy, but Bitcoin Core previously attempted to achieve that using a variety of different heuristics. In the redesigned mempool, called cluster mempool, a fast sorting function is used to put every transaction in the mempool into a linear order. If a new transaction is received to a full mempool, it is only accepted if it sorts higher than at least one other transaction. If a replace-by-fee (RBF) replacement is received, it must rank higher than all of the transactions it replaces.

This conceptually simple new design makes it much easier to reason about mempool behavior and also provides new capabilities that can be used in the development of new policies for avoiding pinning and other problems that affect users of contract protocols. Gloria has been working closely with her colleagues to implement, review, and test cluster mempool, parts of which have already started being incorporated in Bitcoin Core as of early 2024.

Press:

- Sibling replace-by-fee - **Bitcoin Optech Newsletter #287 ¶**
- Discussion with Gloria Zhao, who proposed sibling replace by fee - **Bitcoin Optech Podcast #287 ¶**

Advocacy

> *The goal of this work is to get the broader Bitcoin community to understand and participate productively in protocol development. I think most would agree advocacy is important (decentralized open source development is really hard but it's what we signed up for!). Funding is less of a pain point now than before, but the number of total and new contributors to Bitcoin Core has been steadily decreasing.*
>
> *I limit this to 10-15% of my time, as there is an infinite supply of things to do. A framework I came up with last year was allocating my time across a "funnel." The top is spreading a message "the code needs work and runs on donations," and the bottom might be recruiting somebody to work on Bitcoin Core and looking at PRs together.*

Gloria came to Brink after being a participant in the Bitcoin Core Pull Request Review Club, a project she now co-maintains with fellow Brink engineer Stéphan Vuylsteke. She's also been teaching and mentoring new Bitcoin Core developers Abubakar Sadiq Ismail and Ishaanam, the later of whom is a high school student that already has 21 commits merged into Bitcoin Core.

She's also begun giving more public talks and presentations, including talks at Africa Bitcoin Conference, Bitcoin 2023, Oslo Freedom Forum, BTC Prague, and btc++. With Mark "Murch" Erhardt, she co-authored a 10-part series about Bitcoin Core's mempool for the weekly Optech Newsletter, and helped host 9 socratic seminars with London BitDevs.

Plans for 2024

> - *V3 policy: get it done and deployed*
> - *I think it'll be reasonable to get at least 1 of the package relay subprojects done*
> - *Review cluster mempool PRs when they are ready*
> - *More review, maintenance, and advocacy/mentoring/teaching work*

Gloria's plans for 2024 are very similar to her extraordinary accomplishments for 2023. The deployment of v3 policy will be a major boon to the security of Lightning Network and other contract protocols. Package relay will unlock the full set of benefits that v3 policy makes available. Cluster mempool will make Bitcoin Core's mempool policy much easier for developers and interested laypeople to analyze holistically, and will provide the foundation for multiple future improvements in speed and security. And we expect Gloria's excellent advocacy to continue to inspire both new and existing developers to work on Bitcoin Core.

Organizational Updates

We led with, and focused, this annual report on the engineering work accomplished by Brink-funded developers throughout the year, as their efforts directly furthers our mission of strengthening the Bitcoin software, protocol, and network. As a non-profit organization with no revenues, we rely entirely on community support and our generous sponsors to achieve our mission.

Thanks to the strong support of the community's contributions, we achieved significant growth across our organizational programs. While the bulk of this report has largely focused on our engineering output and achievements, we will now provide a summary of the financial aspects of our efforts.

Note: For clarity, the figures provided are approximations based on the cash method for 2023. Audited, accrual-based GAAP financials will be available in our public 2023 Form 990 filing.

Fundraising

Brink's sponsors donated approximately $2.4 million toward our mission in 2023. These contributions came from over 500 different donors and included approximately 2.5 BTC in donations.

Thanks to the early financial contributions from our Founding Sponsors–Wences Casares, John Pfeffer, and an anonymous donor–Brink's operating costs were fully covered in 2023, **allowing 100% of all new donations to go directly to our programs**.

Our major donors, who contributed over $5,000 each, played a crucial role in our success. We'd like to thank them publicly:

- Startsmall: $1,000,000
- Marathon Digital Holdings: $500,000
- The Draper Foundation: $250,000
- Samara Asset Group / Cryptology: $150,000
- Anonymous: $125,830
- Chun Wang: $100,000
- Stakwork / LND: $60,000
- Lightspark Group, Inc.: $50,000
- BitMEX/HDR Global Trading: $50,000
- Ledger: $15,945
- Van Eck Associates Corporation: $10,000 (plus a future pledge of 5% profits from VanEck Bitcoin Trust ETF for at least 10 years)
- CleanSpark Inc.: $10,000
- Anonymous: $9,725
- Anonymous: $5,000

Brink is thankful for our diverse donor base, which includes high-net-worth individuals, Bitcoin miners, Bitcoin mining pool operators, venture capitalists, Lightning businesses, hardware device manufacturers, family offices, exchanges, and hundreds of individual donors. In 2023, we not only diversified our donor base, but the majority of new donors were also first-time supporters of Brink–a testament to the quality and reputation of our engineers' work and our broader community outreach efforts. Thank you all for your support!

Bitcoin 2023 Matching Fundraising Event

We launched our first matching campaign in 2023, thanks to a generous $500,000 match from our partners at Marathon during the Bitcoin 2023 conference in Miami. The campaign ran from the start of the conference through the end of the year, offering a $2 match for every $1 donated during the conference and a $1 match thereafter. The community responded enthusiastically, and the full $500,000 match was met within the three days of the conference, raising over $800,000 for our developer funding initiatives. The successful collaboration between individual donors and a major Bitcoin business was a notable achievement, and we'll certainly consider it for future campaigns.

Recurring Pledges

With a strong track record of funding talented Bitcoin developers, Brink earned the trust of several donors in 2023 who made multi-year pledges to support our mission. We are deeply grateful to these donors, whose pledged commitments help make our mission and the careers of the engineers we support more sustainable:

- Startsmall: $5,000,000 ($1,000,000 annual commitment for 5 years)
- Samara Asset Group / Cryptology: $450,000 ($150,000 annual commitment for 3 years)
- Stakwork / LND: $180,000 ($60,000 annual commitment for 3 years)
- Lightspark Group, Inc: $150,000 ($50,000 annual commitment for 3 years)
- VanEck Associate Corporation: Pledged 5% of future profits from VanEck Bitcoin Trust ETF (HODL) for at least 10 years
- Bitwise Asset Management: Pledged 3.33% of future profits from Bitwise Bitcoin ETF (BITB) for at least 10 years

Fundraising for the Bitcoin Core Developer Meetings

Twice a year, Bitcoin Core developers gather for in-person meetings to share knowledge, review code, present ideas, and collaborate. Separate from Brink's fundraising for our grants and fellowship programs, Executive Director Mike Schmidt also fundraises specifically for these developer meetings.

Thanks to generous donations from Spiral and OpenSats, we helped to successfully organize two in-person Bitcoin Core developer meetings in 2023:

- Spiral/Block Inc: $25,000 for a three-day event in Dublin, Ireland
- OpenSats: $21,000 for a four-day event in Azores, Portugal

More information about Bitcoin Core developer meetings is available on the https://coredev.tech/ website. Here you will find summaries of both the Dublin and Azores meetings.

Expenses

In 2023, Brink's expenses were approximately $1,600,000. The breakdown of these expenses include:

- **Program: Developer Funding ($1,220,000)**
  - Developer Salaries & Grants ($1,100,000)
  - Travel ($35,000)
  - Office ($85,000)
- **Program: General Bitcoin Core Support ($16,000)**
- **Program: Bitcoin Core Developer Meetings ($41,000)**
- **Program: Bitcoin Optech ($14,000)**
- **Operations, Staff, and Fundraising ($309,000)**
  - Staff Compensation ($200,000)
  - Operational expenses ($92,000)
  - Fundraising ($17,000)

As an organization, we are proud to have 81% of our expenses going directly toward our programs. This high program expense ratio reflects our dedication to supporting Bitcoin development and ensuring that the majority of our funds are used for program-related activities. We thank our sponsors who have covered our operational expenses so that all new donations can 100% go toward our programs.

Developer Salaries & Grants

Brink's primary program is to fund Bitcoin engineers working on open source Bitcoin software. In 2023, the grants / wages for these engineers totaled $1,100,000.

This total includes various benefits and costs as part of their compensation package, including:

- Visa expenses
- Pension contributions
- Healthcare costs
- Computer hardware
- Taxes

Travel Expenses

In addition to compensation for their work, Brink provides a travel subsidy for each engineer. In 2023, Brink covered flights and hotel expenses for all of our Bitcoin Core engineers to attend the Core Developer meetings in Dublin and the Azores.

We were also pleased to fund other engineer travel to Bitcoin developer events throughout the year, depending on the engineers' interests or area of expertise. This included covering travel costs, lodging, and conference tickets for engineers who attended or presented at Bitcoin 2023 Miami, BTC Azores, Advancing Bitcoin, the Africa Bitcoin Conference in Ghana, TabConf, the Lightning Summit, and BTC Prague.

The total expenses for travel, lodging, and tickets for engineers in 2023 amounted to approximately $35,000.

Office Expenses

In 2023, Brink sponsored seven engineers, with five based in our London office and two working remotely. We are pleased to offer this flexibility for engineers who prefer either a collaborative office setting or the option to work remotely.

To facilitate their work in London, Brink covers various expenses, including office and meeting space costs, as well as necessary equipment. We also provide space for colleagues to visit for collaboration and idea sharing.

The total office-related expenses, including rent, insurance, software, and other supplies, in 2023 amounted to $85,000.

General Bitcoin Core Support

In 2023, Brink began financially sponsoring some of the infrastructure supporting the Bitcoin Core project. Our deep involvement in the project means we understand what specific ways we can add value and help developers. Previously, these compute and server resources were often paid out-of-pocket by individual contributors to support essential functions like continuous integration, the build system, and testing infrastructure. Sponsoring this infrastructure is both cost-effective and high leverage in that it helps find more bugs (in the case of fuzz testing) and results in a faster developer experience (in the case of continuous integration) but can represent a huge cost to an individual. Brink is happy to relieve Bitcoin Core contributors of this administrative and financial burden, whether they are Brink-funded engineers or not. In all cases, the contributors retain control of the servers and the autonomy to use them as they see fit for the project.

The total expenses for server and cloud hosting for Bitcoin Core engineers was $16,000.

Bitcoin Core Developer Meetings

The fundraising and management of funds for the Core Developer meetings are distinct from Brink's other programs. Each event incurs several types of expenses, including:

- Venue rental for 3-4 days
- Meals, snacks, coffee, and beverages during the meeting
- Travel expenses (flights, hotels) for attendees not sponsored by other means
- Swag (hats, shirts, hoodies, stickers)
- Meeting materials (electrical & AV, equipment rentals, office supplies, etc.)

The total spent for Core Developer meetings in 2023 was $41,000. Unspent funds are reserved for future Core Developer meetings.

Bitcoin Optech

Although Bitcoin Optech predates Brink, Brink handles the administration of Optech's expenses, which include web hosting fees for bitcoinops.org, email services for the weekly newsletter, and transcription services for the podcast.

The total expenses for Optech in 2023 were $14,000.

Brink Operations and Staff

In 2023, the Brink Operations team consisted of Mike Schmidt, Executive Director, and Emily Kee, Operations & UK Office Manager. This team is responsible for managing day-to-day activities and ensuring that the non-profit operates efficiently. Their responsibilities include administrative management, financial management, human resources, event coordination, fundraising and donor management, and communications and public relations.

Total Expenses for Staff Compensation, Operational Expenses, and Fundraising for 2023 total $309,000:

- Staff Compensation (Emily and Mike): $200,000
- Fundraising (travel, events, swag): $17,000
- Operational expenses (banking, accounting, auditing, legal, supplies, insurance): $92,000.

Brink conducts voluntary annual audits to ensure transparency and accountability. In 2023, we completed our 2022 audit with Rogers & Co., which was conducted in accordance with US generally accepted auditing standards. We were found to be compliant.

For a more detailed view of our finances as a non-profit organization, please view our public 990 filings from the IRS or UK filing history listed on Companies House.

Team

Board

The Brink Board oversees Brink Technology Inc., the 501(c)(3) non-profit organization, and is Brink's ultimate governing body. In 2023, we were pleased to announce that Jonathan Bier joined Brink's Board. With his role administering the BitMEX Open Source Developer Grant Program, long-time experience in the ecosystem, and industry connections, he is a valuable addition to the Board. Jonathan joins Executive Director Mike Schmidt and long-time Bitcoin and Lightning open source contributor Christian Decker on the Board.

In early 2023, Jerry Brito wrapped up his two-year term on the Board. Jerry brought his experience as Executive Director at Coin Center to help build the foundation of the organization in those formative first two years. We thank Jerry for his valuable contributions.

Grant Committee

After forming a separate, specialized Grant Committee in 2022, Brink was fortunate to add engineer Gloria Zhao to the committee in 2023. Gloria joined existing Committee members Mike Schmidt, Christian Decker, and long-time Bitcoin technologist David Harding.

With the Committee responsible for evaluating new engineering grant applications and assessing progress of existing engineers, Gloria's role as a maintainer of the Bitcoin Core project is a significant addition. Her perspectives and in-depth, day-to-day technical knowledge of the Bitcoin Core codebase, adds to an already robust set of Committee members.

Emily Kee

Emily Kee, Brink's Office & Operations Manager, made significant contributions in financial oversight, corporate compliance, event planning, and fostering employee morale.

As Operations Manager, Emily took a proactive role in financial oversight, thoroughly reviewing Brink's accounts in both the US and UK. Her efforts led to increased transparency, accuracy, and stronger financial stewardship. Additionally, she played a key role in reviewing and producing the necessary documentation for Brink's voluntary 2022 audit and 990 filings.

Emily streamlined the onboarding process for all grant recipients and employees at Brink, ensuring that everyone felt welcomed, supported, and equipped to be productive, while maintaining full compliance with US and UK regulations.

Her coordination was instrumental in the success of the CoreDev meetings in Dublin and The Azores, keeping both events under budget and earning positive feedback from participants.

From onboarding to addressing daily HR inquiries and organizing holiday parties, Emily's leadership as London's Office Manager brought stability, transparency, and trust, enabling the office to thrive.

In 2024, Emily will continue to focus on enhancing financial transparency through the voluntary 2023 audit, reviewing existing operational procedures to identify efficiency gaps, reduce unnecessary spending, and steward donor funds responsibly.

Mike Schmidt

As Executive Director, Mike Schmidt works with Brink's Board to lead the organization in fulfilling our mission "*to strengthen the Bitcoin protocol and network through fundamental research and development, and to support the Bitcoin developer community through funding, education, and mentoring*".

His most important responsibilities include:

- Building a Board, Grant Committee, and operational team that can efficiently execute on our mission
- Recruiting and retaining top engineering talent that is most beneficial for Bitcoin
- Fundraising to support and sustain Brink engineers and operations long into the future

Adding Jonathan Bier to the Brink's Board and Gloria Zhao to the Grant Committee enhanced two already strong groups governing Brink's operations.

We were fortunate to add Fabian Jahr as a full-time remote grantee in 2023, while retaining six of Brink's existing long-time engineers.

After a challenging fundraising year in 2022 during the Bitcoin bear market, Mike successfully drove the fundraising initiatives discussed above, resulting in an increase in our reserves by the end of 2023.

To be efficient with both engineers' and Brink's Grant Committee's time, Mike, with valuable HR assistance from Emily, revamped the yearly review process for engineers, resulting in a more streamlined and standardized approach. This new process, served as the foundation of the engineering summaries in this report.

Mike also led a new initiative in 2023 to publish relevant Brink technical engineering calls for public consumption. Recognizing the valuable content in our monthly internal video calls, Mike began recording, editing, and publishing these calls on Brink's new YouTube channel to share knowledge with the broader community.

While much of Mike's time is dedicated to fundraising and operational work, he also significantly contributes towards Brink's programs.

He contributes to, reviews, and publishes the weekly Bitcoin Optech newsletter, primarily authored by Brink Grant Committee member and Mastering Bitcoin author David Harding. Alongside Bitcoin Core contributor Mark Erhardt (Murch), Mike co-hosts the Bitcoin Optech Podcast, a weekly audio discussion of the technical Bitcoin and Lightning developments covered in the Optech newsletter. Including the translation reviews, this totals 51 newsletters published, 51 podcasts recorded with leading Bitcoin and Lightning engineers , coordination on 51 podcast transcriptions, and review of 204 newsletter language translations.

Having organized three Bitcoin Core Developer meetings previously, Mike, with strong support from Emily Kee, Adam Jonas from Chaincode, and sponsors Spiral and OpenSats, organized two more successful developer meetings in 2023.

In 2024, Mike will continue to prioritize Brink's fundraising initiatives while contributing to Bitcoin Optech, organizing Bitcoin Core developer meetings, and working with Emily to ensure the efficient operation of the organization.

Outlook

While 2023 was a strong year for Brink and the engineers we support, there is always more and better work to be done.

With engineers at the core of our mission, we aim not only to continue supporting our existing talented Bitcoin engineering team but also expand support, especially in areas like education, training, and networking opportunities.

By building out our reserves, we can extend the organization's runway and provide job security and sustainability for engineers, which we hope will result in longer-term retention of talented engineers. If financial conditions allow, we would also love to be able to onboard one or more new engineers to Brink.

We all want to see a stronger, more resilient Bitcoin network. As outlined in this report, Brink engineers made substantial progress in improving the foundational Bitcoin infrastructure in 2023 and have already made significant progress in 2024 towards that end. We are grateful for all of the Brink-sponsored engineers' work and thank all of Brink's sponsors for empowering us to achieve this mission!