FCAT

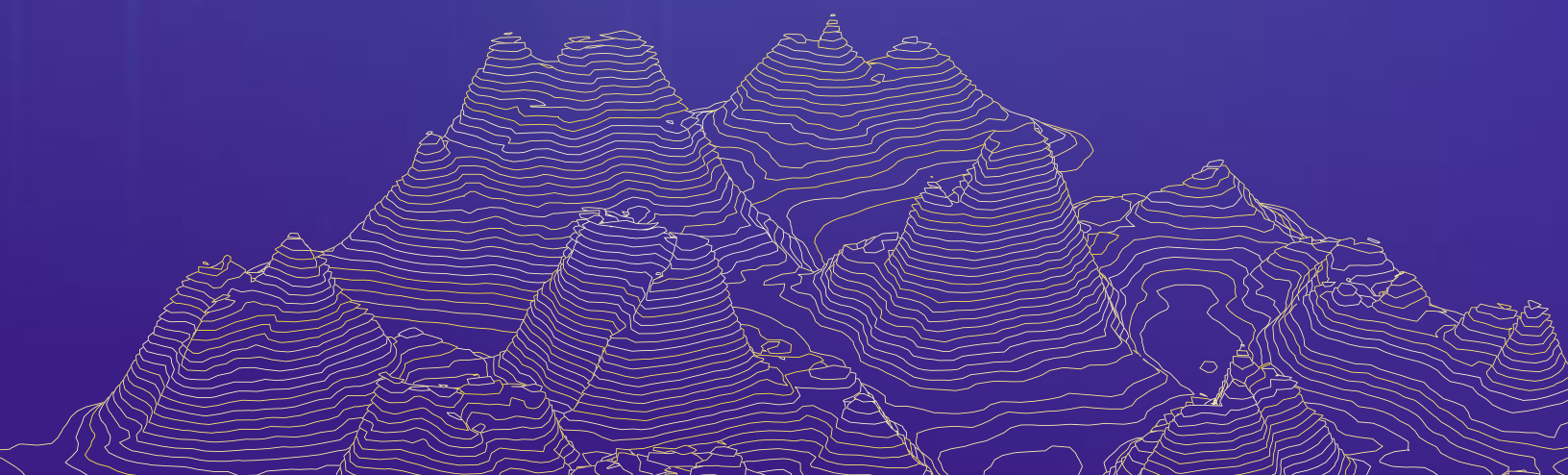
The Bitcoin Renaissance Series 2.0

A Technical Deep Dive

By Jack Blatchford & Matt Allen

August 28, 2025

In this second installment of our “Bitcoin Renaissance” series, FCAT’s blockchain analysts explore the surge in Bitcoin developer activity prompted by ordinals and new technologies — examining the emerging use cases and innovations that are enhancing the protocol’s functionality, scalability, and utility.



The Bitcoin Renaissance is well underway, driven by the introduction of ordinals and the development of new technologies like BitVM. The recent explosion in developer activity and interest in building on Bitcoin has sparked a renewed focus on enhancing the protocol's functionality and scalability. As the Bitcoin ecosystem continues to evolve, we've seen an increase in the demand for blockspace, periods of high and volatile transaction fees, as well as a proliferation of use cases set to enhance the expressivity and utility of the Bitcoin network.

The current Bitcoin Renaissance is not the first time developers have explored uses beyond peer-to-peer transactions. Colored Coins (2012)[1], co-authored by the "father of Ethereum," Vitalik Buterin, stored metadata on the Bitcoin blockchain but ultimately failed to gain significant traction. Scaling too has gone through many iterations and caused quite the confrontation within the bitcoin community, highlighting the challenges surrounding scaling.

From 2015 to 2017, the Blocksize Wars — a contentious dispute between developers, enthusiasts, and industry participants — raged over the path to scaling Bitcoin. The debate was divided between two camps: one that advocated to increase the blocksize limit to help scale Bitcoin, while the other sought to maintain backwards compatibility by modifying block structures instead. This ultimately led to the creation of Bitcoin Cash and the implementation of Segregated Witness (SegWit).

SegWit has effectively increased the block size to 4mb and reduced transaction weights, contributing to its significant impact on Bitcoin's development. Today, over 95% of transactions use SegWit[2] to enable complex monetary and non-monetary transactions. This change demonstrates how modifications to the Bitcoin protocol can have unintended consequences — often leading to new opportunities, as well as challenges.

In the first entry to our "Bitcoin Renaissance" series, we highlighted Ordinals as a pivotal turning point in the modern utilization of Bitcoin's scarce blockspace. For this next installment, we have selected a range of use cases that span both monetary and non-monetary transactions, demonstrating the versatility of new and enhanced functionality, including the impact these changes may have on Layer 2 (L2s) solutions.

As highlighted in Part 1.0 of this series, Bitcoin security becomes more reliant on transaction fees with each subsequent halving. Additional functionality could drive an increase in demand for blockspace and more sustainable transaction fee market.

Building Blocks of Innovation

Throughout this report, the use of a capital "B" when describing "Bitcoin" indicates a reference to the protocol; a lower-case "b" indicates a reference to "bitcoin" as an asset.

In the Bitcoin Whitepaper, Bitcoin's pseudonymous creator Satoshi provided the following definition for the protocol:

"...a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership."[3]

Since the publication of this whitepaper, the utilization of digital signatures has evolved significantly, enabling a wide range of use cases beyond simple peer-to-peer transactions. The original concept of a "chain of digital signatures" has given rise to more complex transactions, inscriptions, and innovative financial instruments.

This has transformed Bitcoin into a robust platform designed for programmable money and data availability. As this paper will explore, the humble digital signature has become a fundamental building block for a vast array of cryptographic and financial innovations, rendering Bitcoin a far more versatile and powerful technology than its initial description as a mere "peer-to-peer cash" system.

So, what are Bitcoin transactions actually?

As detailed by Satoshi, a Bitcoin transaction is a chain of transaction outputs[4] where each output is locked and unlocked by scripts. The locking script (ScriptPubKey) contains the recipient's public key, while the unlocking script (ScriptSig) contains the sender's signature. This allows the recipient to both unlock and spend bitcoin.

Using this model, a standard Bitcoin transaction occurs when the sender (Party A) unlocks their previously received bitcoin (UTXO) and locks it to a new script, which only the recipient (Party B) can unlock with their private key.[5]

As we dig deeper into Bitcoin's scripting language, it's worth emphasizing that it is:

Stateless: lacking persistent storage and incapable of "remembering" data from previous transactions (unless explicitly included)

Not Turing-complete: intentionally designed to be limited in its computational capabilities

Lightweight: simple and constrained to ensure predictability and security

Bitcoin's script was deliberately designed in this limited manner to establish effective measures against malicious attacks, and it met the needs of a burgeoning network that lacked monetary value and only aspired to become peer-to-peer cash. However, times have changed and so too has Bitcoin.

On Bitcoin, an unspent transaction output (UTXO) refers to the specific chunk of bitcoin a user has received. Think of each bitcoin UTXO as a different bill in a user's wallet.

The Bitcoin community has "flipped the script."

Following the SegWit upgrade, new address formats (such as Bech32) changed how transactions are locked and unlocked by moving the unlocking code from the input field to the witness field, enabling more efficient transaction processing and improved scalability. This update is one of a number of enhancements made to Bitcoin over the past 16 years that have rendered many original concepts and terminology obsolete ("timechains," anyone?).

Legacy Transaction	SegWit Transaction
Inputs Previous TX ID: UTXO to be spent ScriptSig: unlocking script	**Inputs** Previous TX ID: UTXO to be spent ScriptSig: empty
Outputs Value: # of Satoshis being spent ScriptPubKey: locking script	**Outputs** Value: # of Satoshis being spent ScriptPubKey: locking script
N/A	**Witness** Script data for unlocking

The Bitcoin community is currently researching and debating ways to address the limitations of Bitcoin's Layer 1. Its stateless and non-Turing-complete scripting language is restrictive, but researchers are exploring potential solutions. This includes soft forks and the development of new technologies like BitVM[6], a virtual machine that could enable more expressive and efficient scripting capabilities, potentially paving the way for more complex decentralized applications and use cases on the Bitcoin network. For instance, some stakeholders are looking to Ethereum's secure and Turing-complete blockchain for inspiration and potential parallel solutions.

A soft fork is a backward-compatible change to blockchain's consensus; conversely, a non-backward-compatible network change is called a hard fork.

The major changes brought about by SegWit highlight the malleability of Bitcoin's immutable ledger. The chain of digital signatures remains unbroken; however, what those signatures lock and unlock is no longer limited to a digital asset's transfer of custody from Party A to Party B.

How is Bitcoin used today?

While the original Bitcoin Whitepaper envisioned a "peer-to-peer electronic cash" system[7] (ultimately contributing to the intentional limitations of Bitcoin's programmability), the recent surge of transactions that includes Ordinals, runes, and BRC-20 tokens shows that many users want to use Bitcoin for more than just digital cash.

Key use cases range across both economic and functional applications; based on recent activity and discourse, these might include the following:

Digital asset creation:

Issuing unique digital assets like Ordinals, similar to NFTs; this could be used to create digital art, collectibles, or virtual items in online games.

Tokenization:

Creating and managing fungible tokens like Runes and BRC-20 tokens (for example, a company could issue tokens representing ownership shares or loyalty points).

Proof of existence:

By inscribing data on the Bitcoin blockchain, users can prove the existence of data or files.

Censorship-resistant messaging:

Protocols like Satogram have utilized Bitcoin to send messages between users and avoid censorship of third-party messaging platforms.

Data storage:

Storing data on the Bitcoin blockchain through inscriptions; this provides a secure and tamper-proof storage environment for important documents and information. As Bitcoin's use for data storage grows, demand for blockspace may increase, leading to fuller blocks and higher fees — which could impact the adoption of L2 solutions.

As we consider these use cases, it's important to note that the adoption of non-monetary use cases on Bitcoin has been limited, and several factors have contributed to this. One major obstacle is Bitcoin's limited smart contracting capabilities. While some of the examples above are possible without robust smart contracts, it is certainly easier to develop in an explicitly smart-contracting environment.

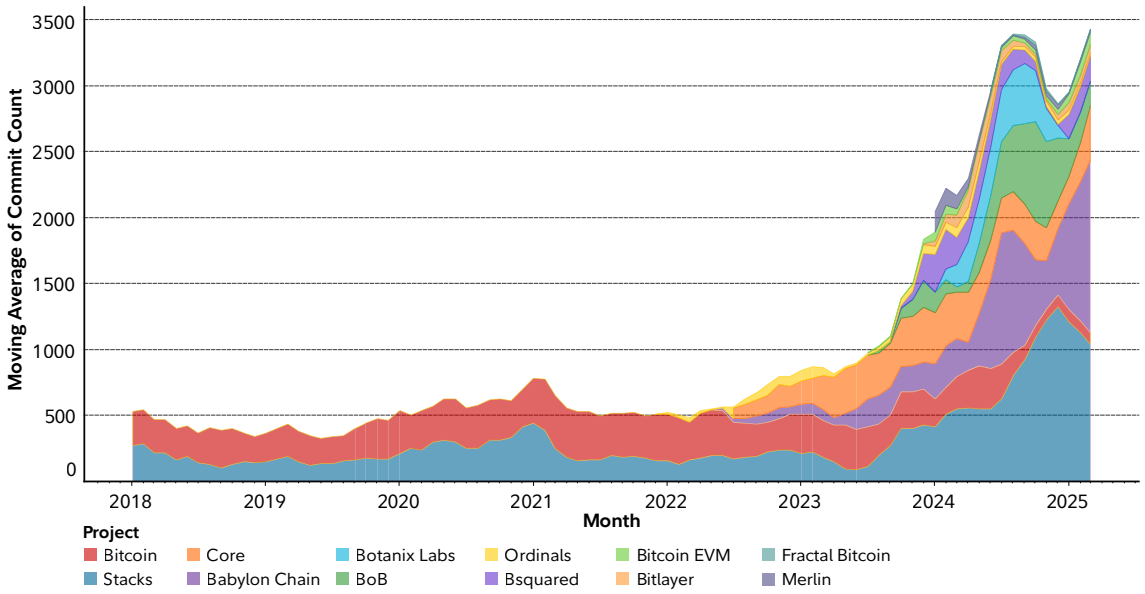
One potential solution to the limitation around smart contracting is to leverage a more programmable sidechain or L2 solution. However, developing non-custodial bridges from the L1 to L2 has also proven to be a significant challenge. To address this issue, Bitcoin Virtual Machine (BitVM) and various soft forks proposals have emerged.

About BitVM2

Published in December 2023, the BitVM whitepaper has sparked a surge in Bitcoin development by introducing a way to verify off-chain computations on-chain. This breakthrough enables the possibility of rollups, Layer 2 solutions, and decentralized finance (DeFi) on Bitcoin — and could even allow Bitcoin to serve as a data availability and settlement layer for computations currently on other chains.

As a result, it is now theoretically possible to connect Bitcoin to a sidechain or L2 that supports smart contracts and bridge back to the network. Developer excitement following the introduction of both BitVM and Ordinals is evident based on the resulting, unprecedented increase in Bitcoin development, detailed below.

Figure 1: Monthly Code Commits in Public Repositories



Source: Sherlock Data as of 03/31/2025

Galaxy Research estimates that over $47 billion[8] worth of bitcoin could be bridged to L2s by 2030, largely using centralized solutions, such as wBTC and CbBTC. If BitVM delivers on trust-minimized bridges, this figure could be even higher.

While this introduction was a monumental accomplishment, BitVM is no panacea, as it was restricted to two parties, severely limiting feasibility. To address some of these challenges, the BitVM2 whitepaper was released in August 2024 and was met with enthusiasm when the community identified tangible benefits in its ability to facilitate permissionless verification.

Unlike previous solutions that relied on a limited set of pre-selected verifiers, BitVM2 allowed anyone to participate in the verification process. It also improved efficiency by reducing the number of on-chain transactions needed to resolve disputes from 70 to 3.

Together, this helps to minimize the cost to use BitVM and enhance system decentralization and security by removing the risk of collusion or compromise among a small group of verifiers.

How BitVM2 works

BitVM2 operates on the principle of optimistic computation — i.e., assuming that operators are honest unless proven otherwise. This is combined with the use of cryptographic proofs that allow for efficient verification of complex computations called Succinct Non-Interactive Arguments of Knowledge (SNARKs)[9].

The process occurs as follows:

Setup	The process starts with a one-time setup where multiple parties (n-of-n) generate keys and agree to participate in verification. While an assertion can be challenged later, all parties pre-sign transactions and contracts, ensuring they all agree to the terms of participation before any computation occurs.
Assertion	A prover — who wants to execute a computation on Bitcoin — submits an assertion claiming that a specific function, when given certain inputs, produces a particular output.
Verification	Unlike the original BitVM design, anyone can act as a verifier; if a verifier suspects that the prover's assertion is false, they can issue a challenge.
Challenge	Upon receiving a challenge, a verifier must provide evidence to support their claim; this evidence takes the form of a SNARK proof, which demonstrates the validity of their computation without revealing the underlying data.
Resolution	Computations are either proven or disproven: if the verifier fails to provide a valid SNARK proof within a specified timeframe, their assertion is rejected, and they may lose funds as a penalty; if they provide a valid proof, their assertion is accepted, and the computation is considered verified.

SNARK proofs are akin to magic tricks: the magician (prover) can convince you (verifier) that they performed a specific action without showing you how they did it.

While BitVM2 remains promising, full implementation continues to face barriers due to the complexity and cost of generating and verifying proofs. Researchers are working to optimize performance, and developers are exploring new use cases — likely helping to support a growing ecosystem of innovative Bitcoin-based solutions.

Covenants, CAT & More

The adoption of BitVM to design more flexible and efficient ways to emulate smart contracts is not the only driver of innovation on the Bitcoin network. Whether it's for increased security, privacy, or enhanced expressivity, developers are seeking additional ways to create more complex transactions. This has all served to reignite interest in older proposals designed to enable Bitcoin covenants through opcodes, such as OP_CTV, OP_CSFS, and OP_CAT.[10]

> "In the context of Bitcoin, the most useful definition of covenant is that it's when the scriptPub-Key of a UTXO restricts the scriptPubKey in the output(s) of a [transaction] spending that UTXO."
>
> – Anthony Towns[11]

The term "covenant" is actually borrowed from contract law, and in this context, it refers to the enforcement of certain rules for or restrictions on how a bitcoin may be used in the future.

While there is currently a long list of potential opcodes to enable covenants or covenant-like use cases, three stand at the forefront of discussion:

OP_CHECKTEMPLATEVERIFY (CTV), proposed by BIP-119 author Jeremy Rubin, enables a transaction output to have specific conditions around how it may be spent in the future; the ability to impose restrictions on transaction output creates a whole host of possibilities directly on Bitcoin's L1: from vaults with secure withdrawals to Discreet Log Contracts (DLCs).[12]

OP_CHECKSIGFROMSTACK (CSFS) is similar to OP_CHECKSIG but opens up signatures to be used for arbitrary data (as opposed to only checking the public / private keys used to sign a transaction); there are numerous capabilities, including the enablement of full-transaction introspection and replication of other partial introspection opcodes, such as OP_CLTV and OP_CS[13]

OP_CAT (Concatenation) is an original opcode that was removed back in 2010 but has received a large amount of attention and interest as it unlocks significant expressivity; this opcode allows users to concatenate two objects on the stack to combine them into one object, unlocking the ability to create and evaluate hashed data structures in-script (like Merkle trees).[14]

When creating a transaction, there are several ways to restrict how a UTXO may be unlocked in the future:

» For example, a script can be used to lock the UTXO so that only a specific private key can spend it; this is a common practice for most transactions.
» Additionally, a multisig script can be used, which requires a certain number of participants to sign a transaction before the UTXO can be spent.
» Time-lock scripts impose a temporal constraint, mandating that a specific duration or block count must elapse before the UTXO can be unlocked and made available for transfer.

The multisig approach is often referred to as an "M-of-N scheme," wherein N equals the total number of keys, and M is the number of keys required to sign the transaction.

However, covenants would allow for even more complex rules to control spending. By applying this concept, conditions would be enabled based on any part of a transaction, providing users with greater flexibility and control. This unlocks the ability for bitcoin scripts to implement further introspection, evaluating spending conditions from any component of the transaction data.[15]

Clear use cases have already emerged — namely the creation of "bitcoin vaults," which would add a time lock on the output of a withdrawal transaction while also including a safety mechanism to recover funds if a malicious actor attempts to withdraw them. This would provide an added layer of security for bitcoin holders.

While it's theoretically possible to create covenants without changing the underlying Bitcoin code, there are unfortunately technical challenges that hinder practical application. Researchers have explored workarounds, such as using Bitcoin PIPEs[16], but these have proved infeasible and are subject to practical limitations.

Given its underlying utility, there is broad consensus that a soft fork is needed. However, the diverse perspectives surrounding Bitcoin's purpose, infrastructure, and applications have left the developer community fielding input from numerous competing interests. To illustrate this point, we have highlighted a recent attempt to collect developer perspectives, including a review of opcodes that support covenants and key developers' respective preferences for the proposed opcode to be merged into Bitcoin.

Figure 2: Developer Perspectives on Opcodes

Source: Bitcoin Wiki as of 05/19/2025[17]

Consensus is messy, and the path forward is unclear. However, there remains significant interest and enthusiasm for seeing these changes implemented. With a mounting queue of bitcoin projects seeking to utilize one or more of these features to develop the next killer app, many users are making considerable sacrifices to find workarounds that allow them to achieve their vision. All of these efforts would likely be simpler, more secure, and less reliant on third parties if one or more of the aforementioned opcodes were activated.

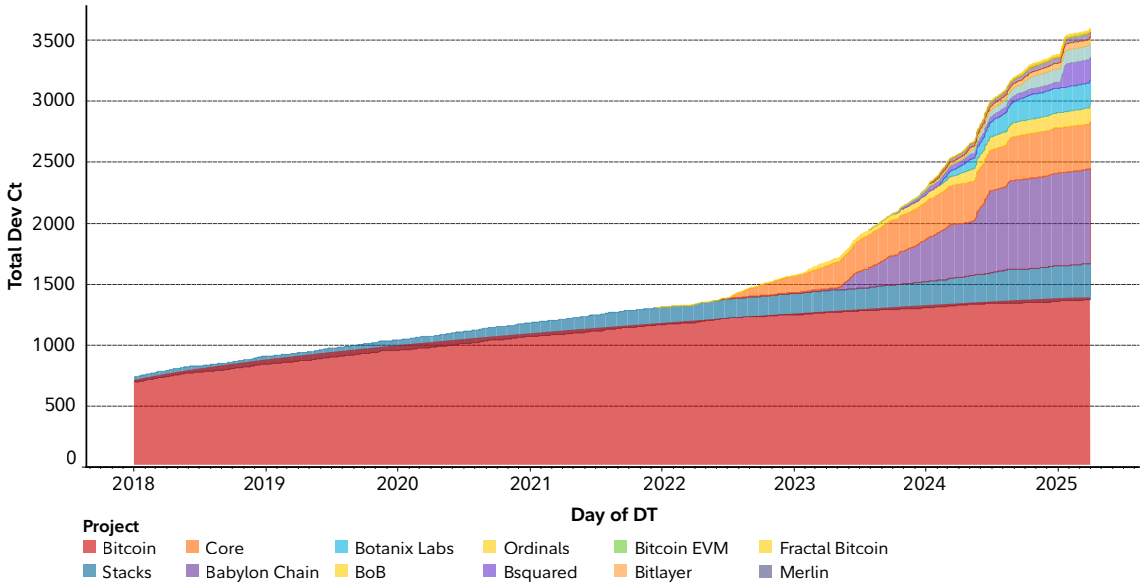
A New Wave of L2s

Enhancements on Bitcoin can only scale the network so far. There is ultimately a limit to the number of transactions that can be squeezed into each block and, given the approximate ten-minute block time, the total amount of transactions that can ever be processed. Therefore, use cases that require near-instantaneous settlement, storage of data greater than 4mb, Ethereum Virtual Machine (EVM) compatibility, and other similar considerations will remain incapable of being built on Bitcoin alone.

That said, it is still possible for developers to build these use cases on top of Bitcoin by leveraging key aspects of the world's largest decentralized public ledger — particularly as some Layer 2 solutions seek to increase transaction velocity and create trust-minimized bridges from Bitcoin's L1 to the L2, where activity can flourish. As an alternative effort, others continue to use bitcoin transactions to store the state of the L2.

Looking at data compiled in Sherlock Data, the rise of developer activity on some Bitcoin L2 projects is evident. Projects like Fractal, Bitlayer, BSquared, Botanix Labs, Core, Stacks, Bitcoin EVM, Merlin, BoB, and Babylon have all significantly increased the overall count of developers in the broader Bitcoin Ecosystem.

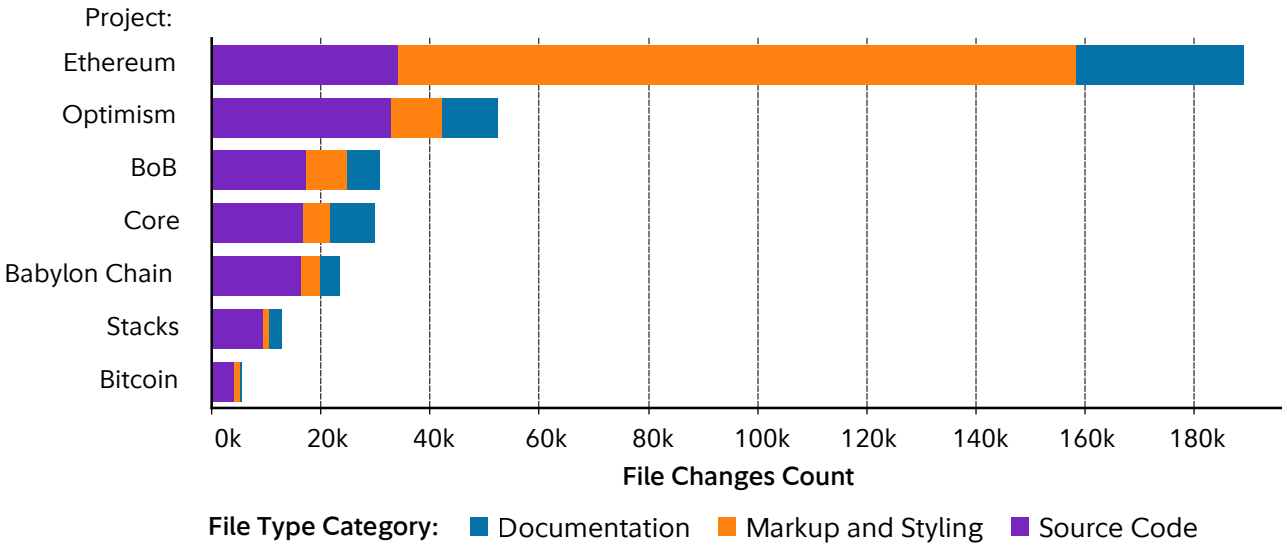
Figure 3: Cumulative Count of Developers on Bitcoin



Source: Sherlock Data as of 03/31/2025

Additionally, in benchmarking the type of file changes occurring on these emergent Bitcoin L2s against blue-chip projects like Ethereum, there are significant updates being made to source code. This indicates strong core development and subsequent interest in expanding the utility and scalability of the network.

Figure 4: Public Repository File Changes (LTM)



Source: Sherlock Data as of 03/31/2025

Figure 5: Community Development Index (Past 90 Days)

This data represents a deep dive into developer activity on public repositories for Optimism, Bitcoin, Ethereum, Stacks, Babylon, BoB, and Core.

	Optimism	Bitcoin	Ethereum	Stacks	Babylon Chain	BoB	Core
Community Development Index	0.69	0.68	0.68	0.62	0.55	0.50	0.47
% of Pull Requests Linked to Issues	14%	23%	7%	21%	10%	1%	0%
% of Pull Requests Merged by Non-Author	40%	67%	43%	32%	25%	11%	69%
% of Code Commits Reviewed by Non-Author	75%	74%	58%	28%	19%	55%	6%
% of Code Commits Linked to a Pull Request	100%	100%	100%	100%	100%	100%	100%
% of Organization Repositories Modified	100%	100%	100%	100%	100%	100%	100%
Unique Code Contributors	47	26	52	16	20	15	5
Average Weekly Code Commits	75	13	103	71	12	15	4
Lines of Code Modified	57,763	2,285	1,261,103	59,827	29,347	13,929	26,910

Source: Sherlock Data as of 03/31/2025

Figure 6: Emerging Bitcoin Layer 2 Community Development Index (Past 90 Days)

This data represents a deep dive into developer activity on public repositories for Bitlayer, Fractal, Bitcoin EVM, Merlin, Botanix Labs, and Bsquared.

	Bitlayer	Fractal Bitcoin	Bitcoin EVM	Merlin	Botanix Labs	Bsquared
Community Development Index	0.40	0.35	0.33	0.29	0.28	0.28
% of Pull Requests Linked to Issues	0%	0%	0%	0%	0%	0%
% of Pull Requests Merged by Non-Author	33%	25%	0%	0%	8%	20%
% of Code Commits Reviewed by Non-Author	4%	5%	0%	0%	0%	0%
% of Code Commits Linked to a Pull Request	100%	100%	100%	80%	67%	80%
% of Organization Repositories Modified	100%	100%	100%	80%	67%	80%
Unique Code Contributors	4	3	1	3	13	1
Average Weekly Code Commits	2	0	2	2	7	1
Lines of Code Modified	1,221	37	1,017	37,931	715	1,045

Source: Sherlock Data as of 03/31/2025

As demonstrated by the data above, developer activity on Bitcoin has been incredibly consistent, and a paradigm shift is likely on the horizon. While certain L2 chains and capabilities require changes to the L1, developers are actively experimenting and building new use cases on top of Bitcoin.

Conclusion

The introduction of BitVM and Ordinals caused a Cambrian Explosion in terms of development activity and interest on Bitcoin. Not only have developers devoted significant time and effort to improving BitVM and building new L2s, but consensus also appears to be growing around a protocol upgrade to support covenants on Bitcoin. As the community moves forward and demand for increased utility and expressivity grows, stakeholders must also consider that the potential impacts and risks associated with these changes may also expand. Identifying this balance can help ensure any next steps are taken with careful consideration and effectively lay the groundwork for strategic development.

Decentralized development can be a slow and sometimes messy process. Where many subsequent crypto protocols have embraced the ethos of "move fast and break things," Bitcoin is driven by a commitment to security

and decentralization. While this approach has perhaps led to a reputation for being "behind" other cryptocurrencies in terms of functionality, it has also helped to ensure the network's stability and security, as well as mitigate developer burnout and brain drain.

The culmination of the Blocksize Wars has yielded a likely trajectory for Bitcoin's scalability, wherein the path forward will not be predicated on L1 alone. Rather, Bitcoin developers and users alike have opted for a strategic trade-off, prioritizing the decentralization and security of the network by limiting block size, while concurrently creating an avenue for the expansion of data availability on the blockchain. Ongoing discussions surrounding the implementation of covenants on Bitcoin, as well as the refinement of BitVM, are poised to contribute meaningfully to the growth and utility of Bitcoin's L2 ecosystem. The development of robust non-custodial bridges and trustless on-chain verification are key building blocks in fostering a vibrant L2 ecosystem, thereby significantly enhancing the overall utility of and use cases available on Bitcoin.

The ultimate outcome of this Bitcoin Renaissance remains subject to ongoing development and inquiry. Nevertheless, the emerging possibilities and novel primitives hold considerable promise. This series will continue to explore the evolving landscape of Bitcoin in Part 3.0, in which we plan to examine emerging use cases on the Bitcoin network and delve into the complex and multifaceted question of bitcoin adoption — helping to foster a deeper understanding of Bitcoin's future trajectory and its potential for widespread acceptance.

Works Cited

1 Coins, Colored. "Colored Coins: Vitalik's First Defi Project." *Medium*, Medium, 9 Jan. 2024, medium.com/@colored_coins/colored-coins-vitaliks-first-defi-project-e832ddb66560.

2 "Segwit Spending Payments." *Transactionfee.Info*, transactionfee.info/charts/payments-spending-segwit/. Accessed 31 Mar. 2025.

3 Nakamoto, Satoshi. *Bitcoin*, bitcoin.org/bitcoin.pdf. Accessed 31 Mar. 2025.

4 "Technical." *How Bitcoin Transactions Work*, learnmeabitcoin.com/technical/transaction/. Accessed 31 Mar. 2025.

5 Antonopoulos, Andreas M. "Mastering Bitcoin." *O'Reilly Online Learning*, O'Reilly Media, Inc., www.oreilly.com/library/view/mastering-bitcoin/9781491902639/ch05.html. Accessed 31 Mar. 2025.

6 "An Overview of Bitcoin Virtual Machine (BitVM)." *Fidelity Digital Assets,* www.fidelitydigitalassets.com/research-and-insights/overview-bitcoin-virtual-machine-bitvm. Accessed 31 Mar. 2025.

7 Nakamoto, Satoshi. *Bitcoin*, bitcoin.org/bitcoin.pdf. Accessed 31 Mar. 2025.

8 Parker, Gabe. "Bitcoin L2s." *Galaxy*, Galaxy, www.galaxy.com/insights/research/bitcoin-layer-2-modular-future/. Accessed 31 Mar. 2025.

9 *ZK-SNARKs: A Gentle Introduction*, www.di.ens.fr/~nitulesc/files/Survey-SNARKs.pdf. Accessed 31 Mar. 2025

10 *Enhancing Bitcoin Transactions with Covenants*, fc17.ifca.ai/bitcoin/papers/bitcoin17-final28.pdf. Accessed 31 Mar. 2025.

11 *Re: [Bitcoin-Dev] CTV Bip Review - Anthony Towns*, gnusha.org/pi/bitcoindev/20220120183822.GA1237@erisian.com.au/. Accessed 31 Mar. 2025.

12 Bitcoin Optech. "OP_CHECKTEMPLATEVERIFY." *Bitcoin Optech*, bitcoinops.org/en/topics/op_checktemplateverify/. Accessed 31 Mar. 2025.

13 Bitcoin Optech. "OP_CHECKSIGFROMSTACK." *Bitcoin Optech*, bitcoinops.org/en/topics/op_checksigfromstack/. Accessed 31 Mar. 2025.

14 Bitcoin Optech. "OP_CAT." *Bitcoin Optech*, bitcoinops.org/en/topics/op_cat/. Accessed 31 Mar. 2025.

15 Stevenroose, et al. "CTV+CSFS: Can We Reach Consensus on a First Step towards Covenants?" *Delving Bitcoin*, 10 Mar. 2025, delvingbitcoin.org/t/ctv-csfs-can-we-reach-consensus-on-a-first-step-towards-covenants/1509.

16 MishaKomarov, and GaloisField2718. "Bitcoin Pipes: Covenants on Bitcoin without Soft Fork." *Delving Bitcoin*, 14 Oct. 2024, delvingbitcoin.org/t/bitcoin-pipes-covenants-on-bitcoin-without-soft-fork/1195.

17 "Covenants Support." *Covenants Support - Bitcoin Wiki*, en.bitcoin.it/wiki/Covenants_support. Accessed 19 May 2025. .

References:

Linus, Robin, et al. "Bitvm2: Bridging Bitcoin to Second Layers." *BitVM*, 15 Aug. 2024, bitvm.org/bitvm_bridge.pdf.

Linus, Robin. "Compute Anything on Bitcoin." *BitVM*, 12 Dec. 2023, bitvm.org/bitvm.pdf.

An FCAT Publication