

# The Future of Bitcoin

## #3: Scaling Bitcoin

MAY 2024



# Table of Contents

<b>Key Takeaways</b>	<b>3</b>
<b>Introduction</b>	<b>5</b>
<b>Introduction to Bitcoin Scalability</b>	<b>6</b>
Why do we need to scale Bitcoin?	6
The size of the Bitcoin L2 opportunity	9
A Framework for Analyzing Bitcoin scaling strategies	10
<b>Bitcoin Scalability Solutions</b>	<b>14</b>
Comparing Key Protocols	14
Underlying Technologies	15
Taproot	15
Schnorr Signatures	16
Merkelized Alternative Syntax Trees	17
BitVM	19
Drawbacks	21
BitVM 2	21
Trustless Bridging with BitVM 2	21
State Channels	22
Lightning Network	22
Overview	22
Payment Channels	23
Hashed Timelock Contracts	24
Limitations	25
RGB/RGB++	26
Overview	26
Technical Architecture	27
AluVM	28
Technical Limitations	30
RGB++	30
Sidechains (kind of)	32
Stacks	32
Overview	32
Functionality	32
Technical Architecture	32
Nakamoto Upgrade	32
Proof of Transfer (“PoX”)	33
Limitations	34
BounceBit	34

Overview	34
Functionality	35
Technical Architecture	35
Bitcoin Restaking	35
Limitations	36
ZK-rollups	36
Merlin	36
Overview	36
Functionality	36
Technical Architecture	37
zkProver	37
Decentralised Oracle Network	38
Limitations	39
Citrea	39
Overview	39
Functionality	39
Technical Architecture	39
Block Production	39
Proof Generation	41
Citrea’s Trust-minimized Bridge with BitVM	42
<b>Outlook and Closing Thoughts</b>	<b>44</b>
<b>References</b>	<b>45</b>
<b>Latest Binance Research Reports</b>	<b>46</b>
<b>About Binance Research</b>	<b>47</b>
<b>Resources</b>	<b>48</b>

## 1

# Key Takeaways

- ❖ Recent developments in Bitcoin, including Ordinals, Inscriptions, BRC-20 tokens, and Runes, have spurred the discussion of Bitcoin scalability solutions to new heights. Bitcoin's average transaction fee rose from US\$1.5 in 2022, to US\$4.2 in 2023, and is US\$9.5 in 2024 so far.
- ❖ Ethereum is valued around US\$450B, with ~US\$45B in total value locked ("TVL") across its various Layer-2 ("L2") solutions, i.e. L2 solutions represent ~10% of Ethereum's total value. Bitcoin, valued at US\$1.4T, has only around ~US\$2B of L2 TVL, representing just ~0.13% of Bitcoin's total value.
- ❖ Key aspects to consider when analyzing Bitcoin scalability solutions include (i) how they solve the trustless two-way bridge issue, (ii) relationship and alignment with the Bitcoin base layer, (iii) whether there are any fork requirements, (iv) what level of incentive alignment they have between users, developers, and crypto newbies.
- ❖ The development of fundamental Bitcoin technologies at the infrastructure level, namely Taproot and BitVM, has expanded the possibilities of protocols that can be built on Bitcoin. Although some of these implementations are still in their infancy, it has not prevented projects from devising innovative solutions to Bitcoin's scaling problem.
- ❖ "Bitcoin-native" projects such as Lightning Network and RGB both aim to increase Bitcoin's P2P transaction capabilities, as well as introduce smart contract capabilities to the chain, while retaining the integrity of Bitcoin. Lightning has launched with relative success so far, while RGB remains in the development stage.
- ❖ Other kinds of scaling solutions also exist, ranging from sidechains, to EVM Layer 1s that use bridged BTC as the staked asset to secure their chains. Although somewhat utilizing the economic security of Bitcoin, bridged versions of Bitcoin often have centralized components, and these protocols cannot truly claim to inherit much Bitcoin security.
- ❖ Zero-knowledge rollups that have appeared in the Bitcoin "Layer 2" scene of late utilize BitVM as its underlying technology to more securely verify rollup data, compared to other scaling solutions that purely post a hash of their block

data into Bitcoin blocks. These rollups arguably inherit the most Bitcoin security at the current stage.

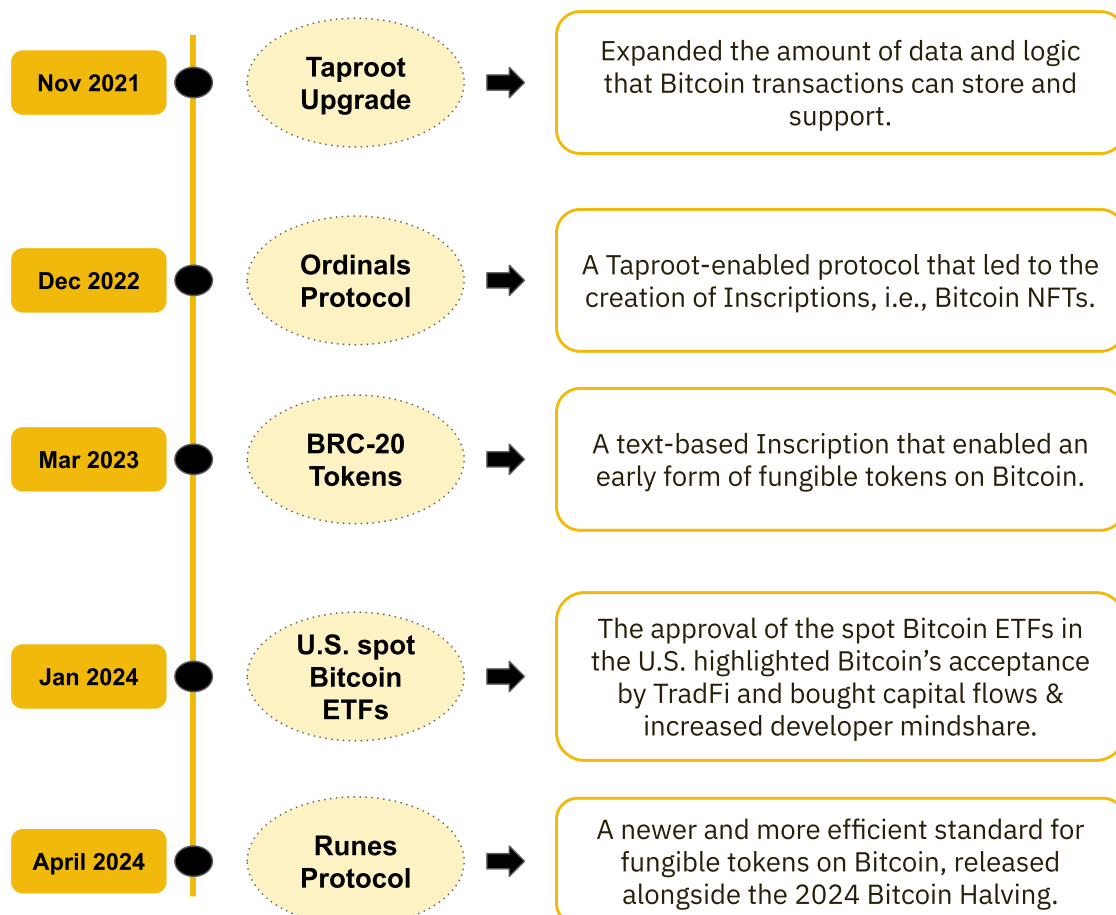
- ❖ As Bitcoin expressivity continues to forge its path, and DeFi primitives such as stablecoins, money markets, staking & restaking, and perpetuals emerge, the importance of Bitcoin L2 solutions will continue to grow. An exciting time ahead, with lots of development expected over the next few months.

## 2 Introduction

While Bitcoin remains the largest cryptocurrency and the flagship asset in the crypto space, it has traditionally lagged behind in scalability, programmability, and developer interest. However, things have been changing.

Casey Rodarmor's launch of Ordinal Theory in December 2022, which led to the creation of Inscriptions and a subsequent Bitcoin NFT hype cycle in 2023, was a pivotal moment. Suddenly, Bitcoin's blockspace was in more demand than ever, with fees skyrocketing as the [mempool](#) became more crowded. This was followed by the community further innovating and finding a way to put fungible tokens on top of Bitcoin, with BRC-20s. This continued the mania, with increasingly visible effects on Bitcoin's key metrics. More recently, we also saw the launch of the Runes Protocol, a more efficient and simple way to put fungible tokens and encourage meme-activity on Bitcoin.

**Figure 1: A short history of notable recent Bitcoin developments**



Source: Binance Research

This Bitcoin renaissance has meant that there is now a whole new group of users, builders, traders, and even degens, who are more interested in Bitcoin than ever before. Bitcoin projects are being funded and developed at a rate we have not seen for some time, and we are even seeing some builders transition from alternative Layer-1s (“L1s”) to Bitcoin. Naturally, some of these teams are very focused on the scalability aspect. While some of the original OGs of the game, including Stacks, continue to innovate, we also have a new group of builders making their first foray into the world of Bitcoin scalability.

In this report, we will focus on this aspect of the Bitcoin story. How do we grow to accommodate an ever-growing ecosystem and build Bitcoin to a level where it can sustain true mass adoption? Read on.

This report is part of our new ***The Future of Bitcoin series***, where we will cover the major areas in which Bitcoin is growing over a set of focused reports. In this edition, we talk about the issues and solutions surrounding Bitcoin scalability, digging into rollups, sidechains, state channels, and more.

*Note: When referring to Bitcoin, we may sometimes use its ticker, BTC. Technically speaking, Bitcoin (BTC) is the native token of the Bitcoin blockchain.*

## 3 Introduction to Bitcoin Scalability

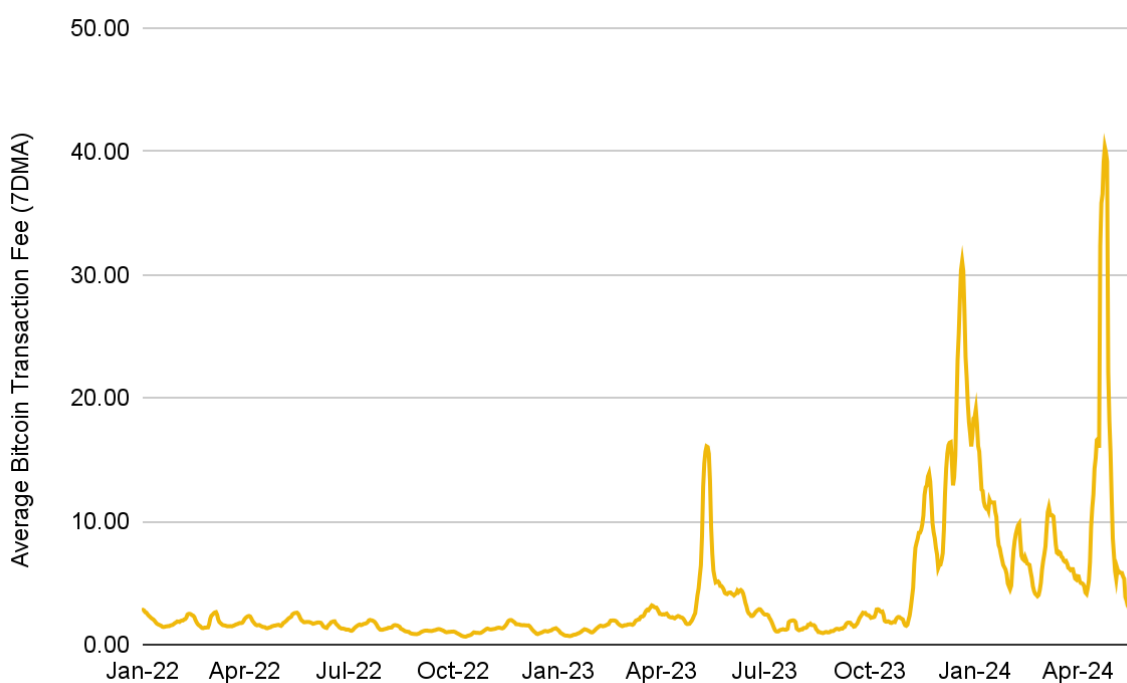
### Why do we need to scale Bitcoin?

The scalability of Bitcoin through L2s or other forms of scalability solutions is not a new topic. This discussion has been ongoing since as early as 2009, when Satoshi Nakamoto implemented a 1MB limit on Bitcoin blocks. The 2017 SegWit fork was a later example of the scalability debate. Projects like Lightning Network, Stacks, and Rootstock have been building solutions for many years.

However, there have been some recent developments that have spurred this discussion to new heights. Central to this [new era in Bitcoin](#) has been the introduction of fungible and non-fungible tokens (“NFTs”) through the advent of Ordinals, Inscriptions, [BRC-20 tokens](#), and [Runes](#). As we can see in Figure 2, this has had a very direct impact on **Bitcoin’s average transaction fees, which rose 175% between 2022 and 2023, from US\$1.5 to US\$4.2. This pattern has continued, with the 2024 average Bitcoin transaction fee upwards of US\$9.** This development has directly highlighted the importance of Bitcoin

scalability solutions, which can help move some of these transactions away from the Bitcoin L1, and towards L2s.

**Figure 2: Bitcoin average transaction fee rose from US\$1.5 in 2022, to US\$4.2 in 2023, and is US\$9.5 in 2024 so far**



Source: The Block Data, Binance Research, as of May 23, 2024

Not only have these innovations had the direct impact of increased fees and a more congested mempool, but they have also had **significant indirect effects**. Ordinals & Inscriptions have helped usher in a **renaissance for Bitcoin expressivity**. Numerous new Bitcoin projects have either launched in the last year, or are currently being funded and developed. These range from all sorts of activities, whether that be **projects focused on creating money markets on Bitcoin, or those focused on bringing other primitives like staking & restaking** to the largest cryptocurrency. All of these new activities are already contributing to, or are expected to contribute to, the Bitcoin mempool, and thus also affect fees. Bitcoin L2s are crucial for these projects, with many building their own, or others using existing providers. Newer projects should also have a choice to deploy on a Bitcoin L2, rather than consider deploying and further congesting the L1.

Even if someone believes that **Bitcoin should only be used for currency transactional purposes, there is still a need for L2s**. 152 million transactions<sup>(1)</sup> occurred on Bitcoin last year. If we anticipate at least 2% of the world population i.e. 160 million people, to make 10 Bitcoin transactions per year - that would be 1.6 billion transactions. For further context, Bitcoin only recently crossed the 1 billion transaction mark. If users are already



complaining about a congested mempool and rising fees with such a relatively low amount of transactions, then clearly there is an issue. If true global mass adoption really is the goal for Bitcoin, then it should be clear that at least a few Bitcoin scalability solutions would be necessary.

*“152 million transactions occurred on Bitcoin last year. If we want at least 2% of the world population i.e. 160 million people, to make 10 Bitcoin transactions per year - that would be 1.6 billion transactions.”*

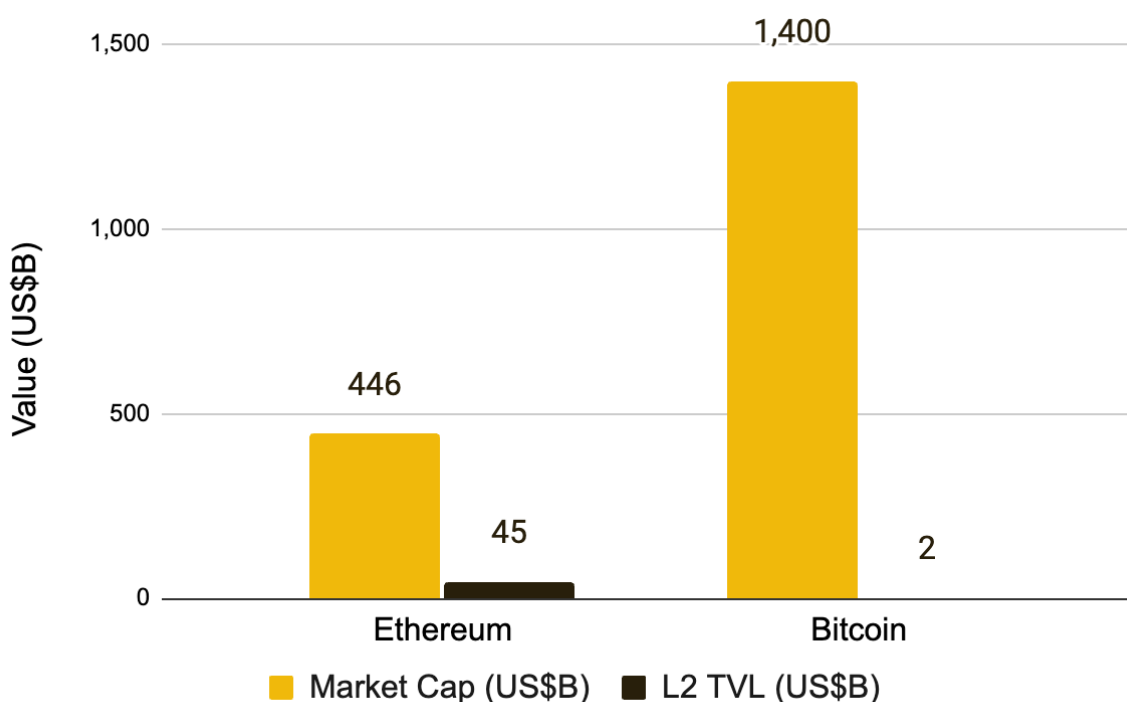
When we consider these factors in conjunction, the **need for strong Bitcoin scalability solutions becomes clear**. However, we should note that this is still a relatively nascent part of the Bitcoin L2 journey, and it is not clear whether the current crop of Bitcoin L2s will win out, or new winners will emerge in the next few years.

## The size of the Bitcoin L2 opportunity

To consider the potential size of the Bitcoin L2 opportunity, we can consider Ethereum, which is the largest smart contract L1, and taking a L2-focused approach to scalability (as opposed to Solana, which is more focused on scaling the L1 itself).

Ethereum is currently valued at ~\$446B<sup>(2)</sup>, with ~US\$45B in total value locked (“TVL”) across its various L2 solutions, i.e. L2 solutions represent ~10% of Ethereum’s total value.

**Figure 3: Ethereum’s L2 solutions represent ~10% of Ethereum’s market capitalization**



Source: Binance Research, CoinMarketCap, L2beat.com, defflama.com, as of May 21, 2024

Note: Bitcoin L2 TVL include Merlin Chain, Lightning Network, Rootstock, Stacks, Bitlayer

Similarly, with **Bitcoin currently valued at ~US\$1.4 trillion, its relatively small L2 solutions currently have a TVL of around US\$2B. This represents ~0.13% of Bitcoin value.**

The leading Ethereum L2, Arbitrum One, has a TVL of ~US\$18B, representing approximately 40% of Ethereum L2s. Extrapolating this, if the Bitcoin L2 market grows to ~US\$14B, the largest L2 could be ~US\$6B. If the Bitcoin L2 market grows to a similar proportion to Ethereum (i.e., 10% of Bitcoin value in L2s), then the largest Bitcoin L2 could have over US\$60B of TVL.











## A Framework for Analyzing Bitcoin scaling strategies

While we cover a selection of some of the larger Bitcoin scaling projects in this report, the reader should note that the actual number of such projects has become large this year, and is increasing every week. A few points to consider when trying to differentiate and evaluate different Bitcoin scaling strategies:

- ❖ **Trustless two-way bridge:** One of the critical points of contention with Bitcoin L2s is the bridge between the Bitcoin L1 and L2. **Due to the limited smart contract functionality of Bitcoin, a trustless two-way bridge has not been possible.** This means that **some form of centralization is typically required to move assets from Bitcoin to the L2 and back.** This may come in the form of a federation i.e., a group of parties that are tasked to manage the two-way Bitcoin bridge, as in the case with Liquid<sup>(3)</sup>.
  - How Bitcoin L2s manage this fundamental issue is an important aspect to monitor when evaluating different projects.
  - BitVM, introduced in a December 2023 paper by Robin Linus<sup>(4)</sup>, proposes a smart contract solution for Bitcoin that will allow it to perform more complex computation. **BitVM might be able to provide a significantly more trust-minimized way to solve the two-way trustless bridge issue** (discussed more [below](#)).
- ❖ **Relationship and alignment with the Bitcoin base layer:** Bitcoin L2s should **maintain close economic alignment with Bitcoin**, with many viable strategies, including the use of native \$BTC as collateral or denominating fees in \$BTC, etc.
  - This may **cast the widest net in terms of securing a user base**, including some of the more monetary-focused members of the Bitcoin community.
  - This may also be a good strategy considering **Bitcoin remains the largest, most decentralized, and arguably the more attack-resistant cryptocurrency.** Thus Bitcoin L2s might choose to maintain alignment through using Bitcoin blocks for transaction settlement, or data availability, or even execution in some cases.
- ❖ **Fork requirements:** Some Bitcoin scalability projects, both pre-Ordinals, and post-, propose solutions that require Bitcoin to undergo changes in the form of a [hard or soft fork](#). **As we [previously highlighted](#), Bitcoin is usually quite slow to change, and has only seen two soft forks in the last seven years (SegWit in 2017 and Taproot in 2021).**

- This means that the **viability of Bitcoin scalability projects which are relying on a fork, is relatively limited in the short term.**
  - Although, some projects might be worth pursuing in the medium to long term if they might bring significant scalability benefits and as market conditions change.
  - It should also be noted that **some soft fork proposals, including OP\_CAT and OP\_CTV have started to gain renewed momentum**, at least partially driven by the work done by teams such as Taproot Wizards<sup>(5)</sup>.
  - Interest in Bitcoin soft forks is also increasing given that they can be used to add interesting new features to Inscriptions and Runes, which appeals to traders, NFT collectors, and simply degens. This has meant that **individuals that have previously not had much incentive to lobby for Bitcoin soft forks are now more interested in them, adding a whole new level of support** that has previously been missing.
- ❖ **Incentive alignment:** Bitcoin L2s need to ensure incentive alignment across the stack in order to grow and gain mindshare. We can very broadly divide this into three categories:
1. Developers: Bitcoin L2s must ensure that **developers are sufficiently incentivized and motivated to switch to working on Bitcoin from other chains, or start working on Bitcoin**. This might be through many strategies, including developer incentive programs, or retrospective airdrops (like Optimism in the Ethereum L2 world has used).

**Figure 4: Bitcoin ranks at the bottom of the top 10 in terms of full-time developers**

Logo	Ecosystem	Full-Time Developers		Total Developers	
		31 Dec 2023	1Y%	31 Dec 2023	1Y%
	Ethereum	2,392	-17%	7,864	-25%
	Polkadot	792	-10%	2,107	-19%
	Polygon	790	-33%	2,800	-36%
	Cosmos	669	-17%	2,035	-21%
	Arbitrum	592	-19%	1,823	-15%
	BNB Chain	498	-20%	1,650	-36%
	Avalanche	455	-5%	1,486	-6%
	Solana	436	36%	1,615	-46%
	Optimism	432	-15%	1,299	-16%
	Bitcoin	356	-15%	1,071	-19%

Source: Binance Research, Electric Capital, as of Dec 31, 2024

- Users: We can divide this group into **existing Bitcoin holders**, and **those active on other chains**. Both sets of users must be incentivized to experiment with new L2s. For **older Bitcoin users**, this might be through creating more safety mechanisms and a **focus on decentralization**. For **newer users**, this might be through user

**incentive programs, airdrops, effective marketing** to EVM users, etc.

3. Crypto Newbies: One thing to always remember is that **Bitcoin is by far the most recognizable name in the crypto industry**. And that is particularly true following the approval of the spot Bitcoin ETFs in the U.S. earlier this year. Not only can we count financial juggernauts like **Morgan Stanley**, and **JPMorgan** as **Bitcoin spot ETF holders**<sup>(6)</sup>, we can also add more traditional investors like the **State of Wisconsin's pension fund**<sup>(7)</sup>. The point is that as crypto attracts more new users and investors, **Bitcoin may often be the first or among the first assets they may be interested in and look into**. This presents a major opportunity for Bitcoin L2s and they should seek to ensure that they help onboard an outsized share of this new user group to Bitcoin.

Now that you have an idea of why scaling Bitcoin is important and an understanding of a few key aspects to consider when analyzing various Bitcoin L2 solutions, we can start talking about some key protocols.

# 4 Bitcoin Scalability Solutions

## Comparing Key Protocols

Figure 5: An overview of the various Bitcoin scalability solutions we will be discussing

	Lightning Network	RGB	Stacks	BounceBit	Merlin	Citrea
Type	State Channel	State Channel	Bitcoin Sidechain	EVM Layer 1 Chain	Type-2 zkEVM (Polygon CDK)	Type-2 zkEVM (RISC Zero)
Status	Mainnet, live March 2018	Pre-testnet	Mainnet, live 14 January 2021	Mainnet, live April 2024	Mainnet, live April 2024	Pre-testnet
Bitcoin Security	<b>High</b>	<b>Medium</b>	<b>Low</b> Only block proposer selection uses the Bitcoin miner network	<b>Low</b> Merely utilizes bridged BTC	<b>Medium</b> Transactions can be verified through zk fraud proofs on Bitcoin	<b>Medium</b> Transactions can be verified through zk fraud proofs on Bitcoin
Consensus Mechanism	Bitcoin PoW	Bitcoin PoW	PoX - Proof of Transfer (tracks BTC transfers on Mainnet)	PoS (\$BB AND \$BTC tokens)	PoS (for Oracle Nodes) 2	Unknown (as of time of writing)
Level of Decentralization	<b>High</b>	<b>Medium</b>	<b>Medium</b>	<b>Low</b> Relies on BounceBit - BTC Bridge	<b>Low</b> Bridged controlled by MPC, data stored off-chain	<b>Medium</b>
Censorship Resistance	<b>High</b>	<b>Medium</b>	<b>Low</b>	<b>Low</b>	<b>Medium</b>	<b>High</b>
Level of Technical Innovation	<b>High</b>	<b>High</b>	<b>Medium</b>	<b>Low</b>	<b>Medium</b>	<b>High</b>
Execution Environment	N/A	AluVM	Clarity VM	EVM	EVM	EVM (but can support others as well)
Smart Contracts	N/A	Rust, Contractum	Clarity	Solidity	Solidity	Solidity
Main Advantage	Very fast p2p transactions	Most 'native' smart contract solution	Upcoming Nakamoto update to increase BTC security	BTC restaking on <u>BounceBit</u>	Plans for decentralized prover and sequencer network	Trust-minimized bridge for two-way BTC peg
Settlement Layer	Bitcoin	Bitcoin	Stacks	<u>BounceBit</u>	Bitcoin	Bitcoin

Source: Binance Research, <https://l2.watch/>

# Underlying Technologies

Before diving into the protocols, we can have a quick look at the key underlying technologies behind the majority of these Bitcoin scalability solutions. The two primary developments are 2021's Taproot Upgrade, and the recent discussion around BitVM.

## Taproot

Taproot was a 2021 soft fork upgrade to Bitcoin that consisted of three distinct Bitcoin Improvement Proposals ("BIPs"); BIP 340 (Schnorr Signatures), BIP 341 (Taproot) and BIP 342 (Tapscript). These updates brought more privacy, scalability, and composability to Bitcoin. Two major effects that Taproot had was allowing **advanced scripting in the Witness section of a block**, as well as, **removing the data limits between the two sections of a block** i.e. allowing up to 4MB of data in the Witness section.

What follows is a technical breakdown of Taproot and its various components:

### BIP 340 - Schnorr Signatures

A major enhancement to Bitcoin was the introduction of Schnorr Signatures. These signatures offer several advantages over the previous ECDSA mechanism used for key generation and signature verification.

**Key aggregation** is a standout feature, allowing **multiple parties to merge their keys into a single public key and enabling them to sign a single message**. This component of the Taproot upgrade enhances the speed, security, and efficiency of Bitcoin digital signatures. Furthermore, Schnorr signatures are backward compatible with Bitcoin's existing cryptographic algorithm, which is what enabled Taproot to be implemented as a soft fork upgrade, rather than a hard fork

### BIP 341 - Taproot

BIP 342 introduced **changes to Bitcoin's scripting language** to accommodate Schnorr signatures. This proposal also integrates two essential elements to maximize Schnorr's capabilities: MAST and P2TR.

**MAST** ("Merkelized Alternative Syntax Trees") conceals any predetermined conditions associated with transactions. Outcomes that are not utilized remain off-chain, **enhancing privacy and reducing the transaction data size**. This update significantly aids Bitcoin's scalability by minimizing data requirements.



**P2TR** (“Pay-to-Taproot”) introduces a new method for executing transactions using Taproot addresses. It merges features from the earlier P2PK and P2SH scripts into a new script type, **enhancing privacy and improving the mechanisms for authorizing transactions**.

Additionally, P2TR ensures all Taproot outputs look uniform. Due to key aggregation, whether a public key is used individually or as part of a multisig setup remains undisclosed. This greatly bolsters privacy for transactions on the Bitcoin blockchain.

### **BIP 342 - Tapscript**

Tapscript, the final component of the Taproot suite of BIPs, updates Bitcoin’s original scripting language to support Schnorr Signatures, P2TR, and other essential coding for Taproot’s immediate functionality. Over time, **Tapscript is designed to facilitate the implementation of further script updates**, simplifying future enhancements to Bitcoin’s infrastructure.

### **Schnorr Signatures**

Schnorr signatures offer several improvements over the traditional ECDSA signatures used in Bitcoin. A key feature of Schnorr signatures is the linearity in signature generation, which enables the aggregation of multiple signatures into a single one, as compared to ECDSA signatures that are not linear. This not only **enhances privacy** by making transactions with multiple inputs look like those with a single input but also **boosts scalability by reducing the data volume** on the blockchain.

Schnorr signatures provide provable security based on the discrete logarithm problem, which is a foundational concept in cryptography, and also utilizes Elliptic Curve Cryptography. This security is more straightforward to demonstrate than with ECDSA due to the simpler mathematics involved. Furthermore, Schnorr signatures are **non-malleable**, meaning they cannot be altered without the corresponding private key, enhancing their security against certain types of attacks.

$$s = r + ek \pmod n$$

Signature generation in Schnorr Signatures, where: r is a randomly generated nonce, e is the value to be hashed, k is the private key, and n is a large prime number.

$$s = k^{-1}(e + dr) \pmod n$$

Signature generation in ECDSA signatures, where:  $k$  is a randomly generated nonce,  $e$  is the hash of the message converted to an integer,  $r$  is the x-coordinate of an elliptic curve point, and  $n$  is a large prime.

### The Discrete Logarithm Problem

The discrete logarithm problem is a mathematical problem that forms the foundation for many cryptographic systems, including those based on elliptic curve cryptography (“ECC”), such as the Schnorr and ECDSA signature algorithms.

The discrete logarithm problem involves finding the exponent in the context of modular arithmetic, given a base and a result. Formally, it is defined as follows:

Given a finite group  $G$ , a generator  $g$  of the group, and an element  $y$  in  $G$ , find the integer  $x$  (if it exists) such that:

$$g^x = y \pmod p$$

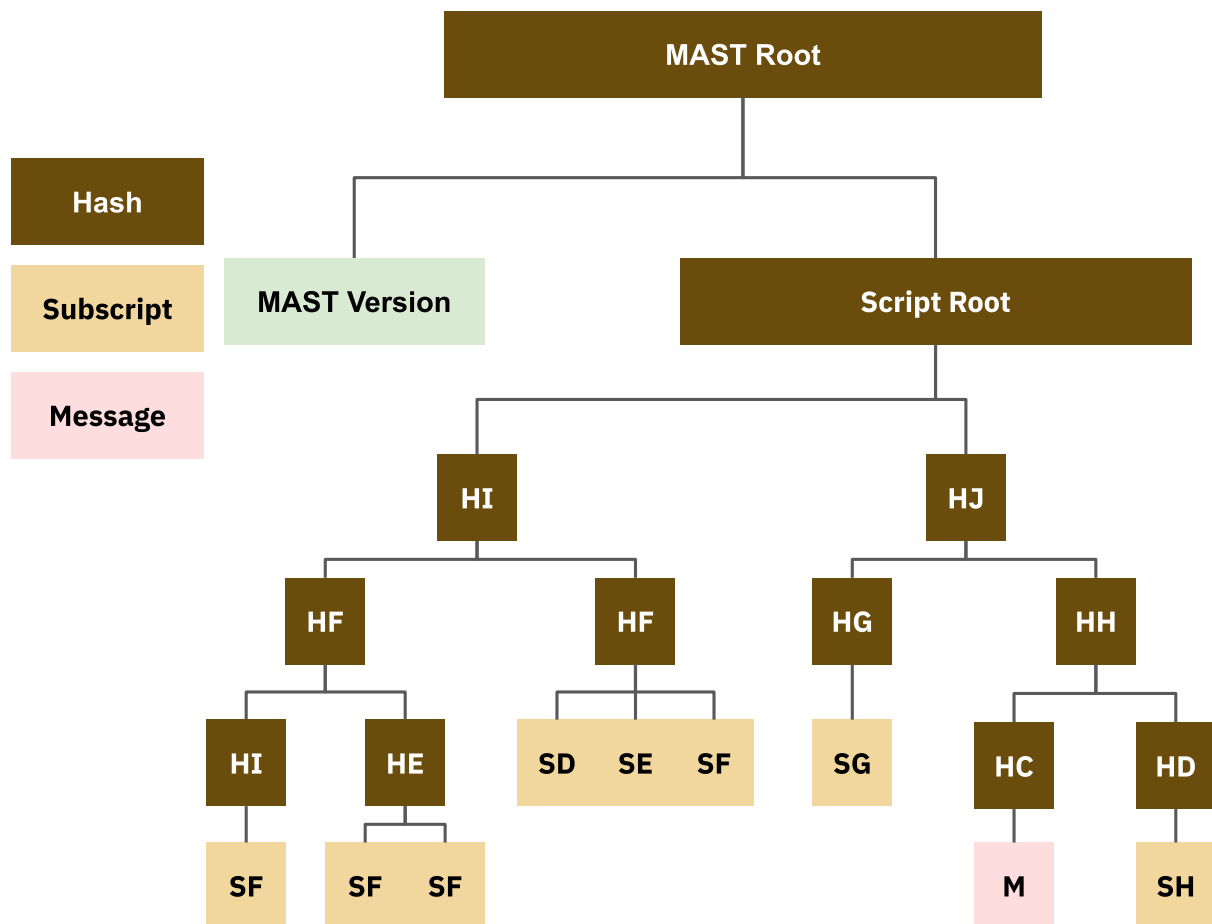
Where  $p$  is a prime number defining the modulus for the group.

Given that the prime number  $p$  is large enough, it is computationally tedious to find  $x$ , i.e. the private key.

### Merkelized Alternative Syntax Trees

MASTs enable **more complex conditions for how bitcoins can be spent, but with enhanced privacy**. With MAST, various spending conditions are included in a Merkle tree, and only the relevant branch is revealed at the time of the transaction. This means that the details of unused spending conditions remain hidden, improving privacy.

**Figure 6: Derivation of a MAST root**



Source: Binance Research, [BIP-341 \(MAST\)](#)

The MAST root is calculated using the hashes of the scripts and messages to be included in the root.

MAST introduces some improvements to Bitcoin’s functionality:

### 1. Large multi-signature constructs

The current CHECKMULTISIG function supports up to 20 public keys, but expanding beyond this number becomes complicated and can quickly exceed Bitcoin's script size and operation count limits. However, with Merkelized Abstract Syntax Trees (“MAST”), more extensive and complex multi-signature constructs can be simplified.

For instance, a 3-of-2000 multi-signature scheme could be broken down into 1,331,334,000 smaller 3-of-3 CHECKMULTISIGVERIFY conditions within a 31-level MAST.

This approach **keeps the scriptPubKey size constant at 34 bytes** and reduces the redemption witness to under 1,500 bytes, making it much more efficient.

## 2. Commitment of non-consensus enforced data

MAST can enhance how non-consensus enforced data, like message-signing keys, are committed. Typically, committing such data requires the use of OP\_RETURN, which takes up block space. With MAST, **this data can be integrated as a branch of the tree**, potentially requiring no additional witness space or at most 32 bytes. This feature is particularly beneficial for users who need to sign messages with keys that are not meant for spending, allowing them to do so without accessing their main funding key.

## BitVM

Introduced in December 2023, the primary goal of BitVM is to scale the Bitcoin network by **introducing smart contract capabilities** that are similar to those offered by Ethereum's EVM, but **without requiring significant changes to Bitcoin's existing infrastructure**.

The BitVM protocol, akin to Optimistic Rollups and the "Merkelize All The Things" ("MATT") proposal, **operates on a foundation of fraud proofs and a challenge-response protocol**. It is designed to function without necessitating any modifications to Bitcoin's existing consensus rules. The core mechanisms of BitVM include the **use of hashlocks, timelocks, and extensive Taproot tree structures**, which collectively support its operational framework without altering the fundamental principles of the Bitcoin network.

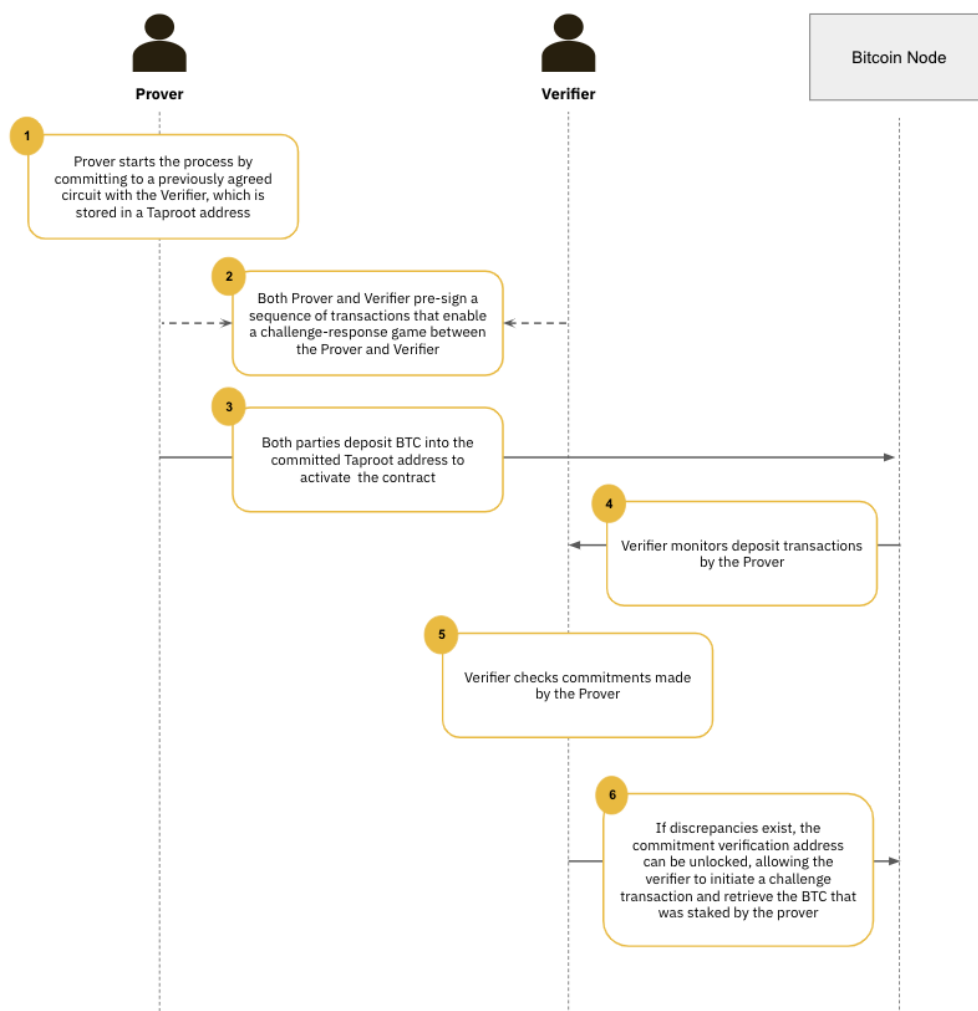
In this system, a prover claims that a specific function will produce a certain output from given inputs. If this claim proves to be false, the verifier can then provide a concise fraud proof to challenge and penalize the prover. This mechanism allows for the verification of any computable function directly on the Bitcoin network.

*“In this system, a prover claims that a specific function will produce a certain output from given inputs. If this claim proves to be false, the verifier can then provide a concise fraud proof to challenge and penalize the prover. This mechanism allows for the verification of any computable function directly on the Bitcoin network.”*

## Workflow

1. **Initialization:** The prover transforms the verification function into a Boolean circuit, then converts all the logical operations of the circuit into a GateScript using a logic gate promise verification script. This GateScript is then organized into a Merkle tree to create a Taproot address, also known as a promise verification address.
2. **Transaction:** The prover sends a Bitcoin transaction, depositing a specified amount of BTC to the Taproot address and discloses zk-proof data and commitment data.
3. **Verification:** The verifier checks if the Tapscript in each leaf of the Taproot tree can be unlocked using the zk-proof and commitment. If it can be unlocked, they issue a challenge transaction to penalize the prover.

**Figure 7: Proof verification flow in BitVM**



Source: Binance Research, <https://bitvm.org/>

## Drawbacks

BitVM is currently still in the **theoretical stage, and there remains a significant gap before it can be fully implemented**. The following are some of its limitations:

- ❖ BitVM primarily operates within a **two-party system** involving a prover and a verifier. This structure restricts the types of interactions and transactions that can be conducted, potentially limiting broader adoption and application in scenarios that require multi-party coordination.
- ❖ While the on-chain footprint of transactions is minimized, the requirement for on-chain execution in the case of disputes can be **computationally expensive and complex**. This may hinder the system's scalability and efficiency, particularly under high load or in complex dispute scenarios.
- ❖ BitVM addresses only the verification of Layer 2 execution results on Bitcoin, and does not resolve the issue of cross-chain transfers of BTC assets between L1 and L2.

## BitVM 2

To address some of the limitations of the original BitVM implementation, an **improved version coined BitVM 2** was conceived by the team as well.

It aims to reduce the need for two predetermined parties to constantly be in a challenge-response state, and makes it **permissionless for anyone to run a verifier**. This solution would still require a one-time setup where at least 1-of-n parties are honest, but while the program is running, anyone is able to challenge an invalid proof without having to be part of the initial group. This allows multiple verifiers to synchronously challenge a prover's claims, improving the robustness of the system.

This theoretically sounds like a great improvement, but it may be some time before enough research and development has been done to use BitVM 2 in production, seeing as the original version of BitVM is still in the works.

## Trustless Bridging with BitVM 2

BitVM could pose as a solution to **enhance the security and efficiency of current Bitcoin bridging solutions** - many of which are currently managed by centralised entities. It could enable the implementation of a light client for a target chain (such as a rollup) to accurately verify transactions like peg-ins and peg-outs on chains that support smart contracts.

In BitVM, deposits are managed by a committee of Provers and Verifiers. The security of the deposits is assured as long as **at least one member of this committee remains honest**. When a user initiates a peg-out, the current Prover checks the rollup's state off-chain and, if verified as correct, transfers BTC to the user. Verifiers monitor and confirm the accuracy of this process. Should the Prover act improperly, such as failing to respond or sending BTC to an incorrect address, Verifiers can initiate an on-chain challenge to block the Prover from accessing the deposits.

BitVM leverages what are known as **cross-chain light clients**—programs capable of confirming state changes on other blockchains. A sidechain, presumed to support smart contracts, would implement a Bitcoin light client to validate Bitcoin transactions and vice versa. The expressiveness limitations of Bitcoin's Script language prevent light clients from being implemented as on-chain programs. Instead, the sidechain light client is realized through BitVM, which involves participants committing in advance to the program through **pre-signed Bitcoin transactions**. The program runs off-chain, and disputes are resolved through a challenge-response protocol to ascertain the correct result.

## State Channels

### Lightning Network

#### Overview

The Lightning Network was proposed in 2016 by Joseph Poon and Thaddeus Dryja to directly address the limitations of the Bitcoin blockchain, primarily its scalability issues. Bitcoin has famously been limited to a relatively low transaction capacity, usually between 3-7 transactions per second ("TPS"). Taken in combination with transaction fees that can be high for daily transactions, in addition to the need to wait for confirmation across six blocks, it is clear that the Bitcoin L1 is not ideal for small, regular payments. This is where the Lightning Network comes in.

**The Lightning Network is composed of "payment channels," which are practically just [multisig](#) smart contracts that facilitate transactions between two users.** Participants can create accounts and deposit funds, with the deposited amount setting the balance of the channel, and all subsequent transactions occurring off-chain. This translates to higher throughput and low fees, as users don't have to compete for blockspace or wait for L1 consensus to transact.

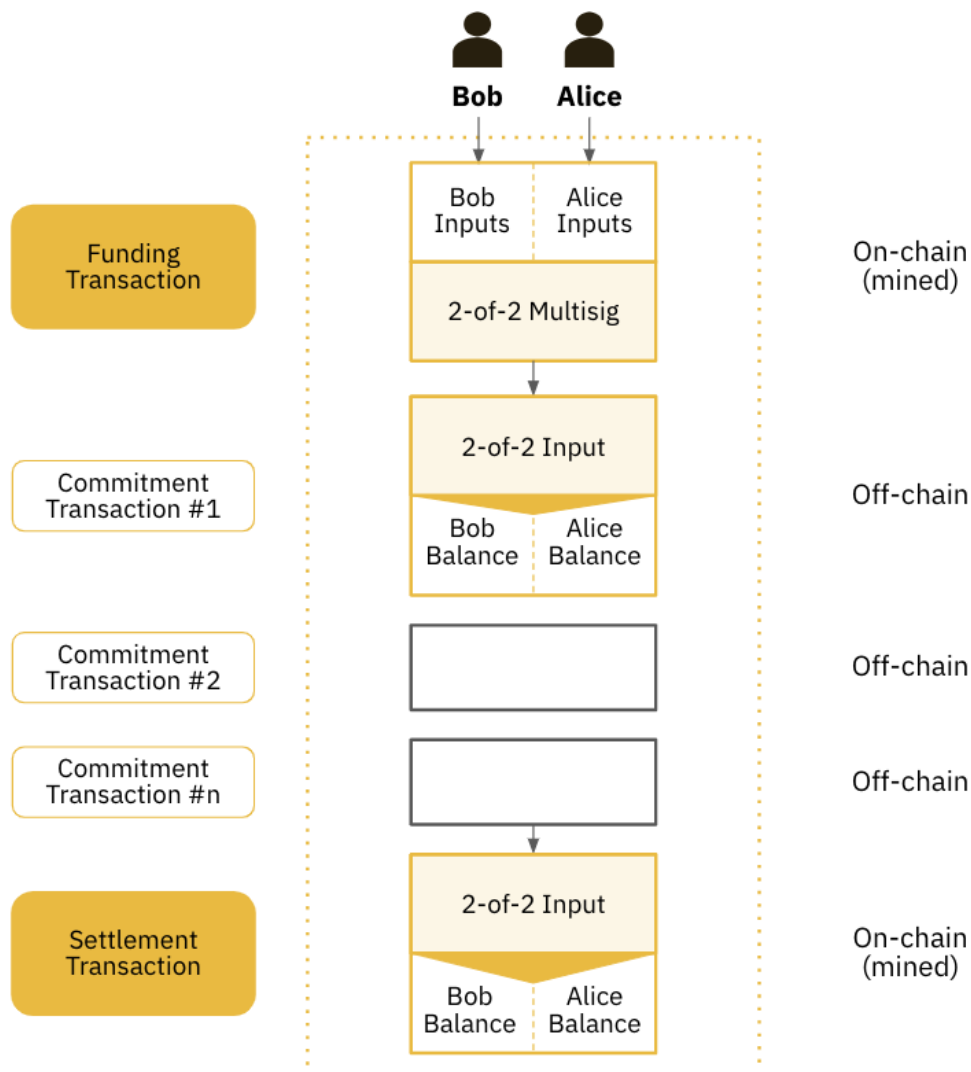
Ultimately, once Lightning Network users decide that they are finished transacting via the payment channel, they can elect to close the channel. Subsequently, an aggregate transaction that summarizes off-chain activity is settled on-chain to the Bitcoin network. Intermediate transactions remain off-chain and are not recorded on the L1, improving transaction privacy. In this way, the **Lightning Network inherits Bitcoin's security, and allows for cheaper, more private transactions for users.** The network's design even allows participants to send funds even without a direct channel to one another, provided there is a connective path of channels across the network.

## Payment Channels

1. **Recharge** - The first transaction determines the balance of a channel, which we call a "recharge transaction". This transaction needs to be broadcast to the network and recorded on the blockchain to indicate that the channel is open.
2. **Update** - To update the balances of both parties in the channel, both parties need to manually exchange signed "**commitment transactions**". These transactions themselves are valid and can be sent to the Bitcoin network at any time, but both parties will temporarily save them locally and will not broadcast them unless they are ready to close the channel. In this way, the balance status of both parties in the channel can change thousands of times per second without issues.  
The speed of update is limited only by the speed at which both parties create, sign, and send commitment transactions to each other. Every time both parties exchange a new commitment transaction, they also **invalidate the previous state of the channel**; therefore, only the latest commitment transaction can be "executed". The purpose of this design is to prevent one party from deceiving the other and sending an outdated but favorable state to the chain to close the channel.
3. **Close** - Ultimately, the channel can be closed in one of two ways: 1. Both parties mutually agree to close it by sending a **closing transaction** (also known as a "settlement transaction") to the Bitcoin network, or 2. One party unilaterally decides to close it by sending the **last commitment transaction** to the network. This mechanism prevents situations where one party going offline could indefinitely "lock" the balance of the other party in the channel.



**Figure 8: How a Lightning Channel works**



Source: Binance Research, Lightning Network [Documentation](#)

Throughout the **entire lifecycle of the channel, only two transactions are sent to and recorded on the Bitcoin blockchain:** the initial funding transaction and the final settlement transaction. Between these two transactions, both parties may exchange countless commitment transactions, none of which require recording on the blockchain.

### Hashed Timelock Contracts

The security and trustworthiness of transactions are maintained through Hashed Timelock Contracts (“HTLC”), which ensure that transactions are safe and require no trust between parties. HTLCs allow the conditional transfer of funds between parties based on the

revelation of a pre-agreed secret within a specific timeframe. If the secret (usually a cryptographic hash) is not revealed before the deadline, the funds are returned to the sender.

### Hashed Timelock Contracts (“HTLC”) in the Lightning Network

#### 1. Agreement and Hash Creation:

- Two parties, Alice and Bob, agree to a transaction where Alice will pay Bob.
- Bob generates a secret, computes its hash, and sends the hash to Alice.

#### 2. Setting Up the HTLC:

- Alice sends the funds to a special type of smart contract or script on the blockchain that implements the HTLC.
- This HTLC holds the funds and stipulates that Bob can claim the funds only if he reveals the preimage (original secret) of the hash before the time lock expires.
- If Bob fails to reveal the secret within the time limit, Alice can reclaim the funds.

#### 3. Bob Claims the Funds:

- To claim the funds, Bob submits a transaction to the HTLC that includes the original secret.
- The HTLC script verifies that the hash of the provided secret matches the hash stored in the contract.
- If the verification is successful and the time lock has not expired, the funds are released to Bob.

#### 4. Fallback if Bob Does Not Act:

- If Bob does not reveal the secret within the specified time, the HTLC includes a mechanism allowing Alice to reclaim her funds after the time lock expires.
- This ensures that Alice's funds are not permanently locked if Bob decides not to cooperate or is unable to provide the secret.

### Limitations

The Lightning Network has arguably worked well in achieving its aim of providing a cheap, quick, and relatively simple way to transact using Bitcoin. Nonetheless, there are clear limitations:

- ❖ **Channel liquidity:** Users **need to deposit enough BTC** into the channels in order to transact. This can limit the usability of the Lightning Network, as it may not be suitable for users with limited liquidity or those who engage in infrequent transactions.

- ❖ **Lack of smart contracts:** Although Lightning is certainly useful for Bitcoin transfers, it does not support other types of smart contracts and does not provide functionality like many top Ethereum L2s do.

## RGB/RGB++

### Overview

RGB is a smart contract protocol built on top of Bitcoin that has been in the works since 2019. It was introduced as a way to implement smart contracts and tokenization on the Bitcoin network **without affecting the core protocol**, keeping the operations off-chain to maintain efficiency and privacy. Smart contracts run on a proprietary Turing-complete VM (AluVM) built by the [LNP/BP](#) association is in charge of Lightning Network and Bitcoin developments, and also spearheads the development of RGB.

It can be thought of as a Layer 2 (or Layer 3 if using the Lightning Network) protocol, whereby the data and functions of the **smart contracts are stored completely off-chain**, and are private to other parties that do not own the contract. Zero knowledge proofs of these transactions are shared among validators on RGB for client-side validation. Client-side validation checks that transactions are valid, ensures the **absence of double-spending**, and validates the user's authorization to interact with these smart contracts.

This is relatively different to the current general understanding we have of smart contracts on blockchains, whereby the transactions, contract code, and data are available on-chain. This is how the RGB protocol allows for privacy, while at the expense of decentralization, as all RGB smart contracts have an “owner(s)” that have full rights over who can interact with the contract, and whether contract data can be publicized.

### Functionality

RGB is theoretically technically able to support DeFi operations similar to those on the EVM. It currently supports basic standards for Fungible Tokens, Non-Fungible Tokens, Digital Identities, and domain names. It is also able to support inter-contract interactions.

Currently, existing projects building on RGB include the likes of DEXes, NFT Marketplaces, Money Markets, Bridges, and Wallets. However the majority of these are still in the testing or private beta phases, seeing as RGB itself is still very much a work in progress.

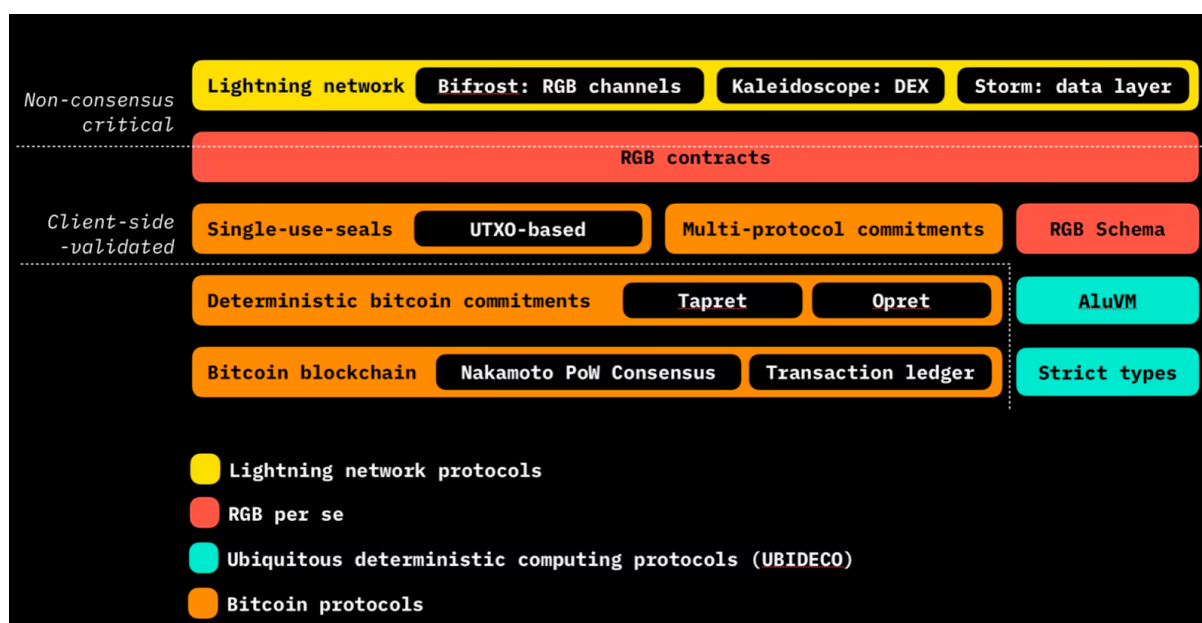
## Technical Architecture

Each RGB smart contract begins with a genesis state, established by a smart contract issuer (or simply, issuer), and evolves through a directed acyclic graph (“DAG”) of state transitions. These transitions are stored as *client-validated data*, meaning they are **not recorded on the blockchain** or within Lightning Network transactions/channel states. The state is linked to unspent Bitcoin transaction outputs (“UTXOs”), designating them as *single-use seals*.

The entity capable of spending the corresponding transaction output is considered the **owner of the state and holds the authority to modify the related smart contract state** by initiating a new *state transition*. This act of utilizing the transaction output that holds a previous state is referred to as closing of a seal, and the combination of the spending transaction and the additional extra-transaction data concerning the state transition is termed a witness.

Separately managed RGB contracts can interact through the **Bifrost** protocol over the Lightning Network, facilitating multiparty **coordinated state changes**. This interaction capability notably supports functionalities like decentralized exchanges (“DEX”) over the Lightning Network and similar applications.

**Figure 9: How RGB interacts with Bitcoin**



Source: Binance Research, RGB Documentation

### Deterministic Bitcoin Commitments

Deterministic Bitcoin commitments (“DBC”) offer a method to generate provably unique commitments within Bitcoin transactions. The RGB protocol supports two types of DBCs:

those based on taproot outputs (referred to as “**tapret**”) and those based on OP\_RETURN outputs (known as “**opret**”), the latter being suitable for older hardware that does not support taproot.

A tapret commitment utilizes an OP\_RETURN-based script, which includes a multi-protocol commitment. This is embedded within an unspendable script path inside the taproot script tree. As a result, **the script/commitment remains hidden within the Bitcoin transaction** or blockchain data (neither in the script output nor in the witness); only the scriptPubkey of the transaction output reveals a commitment to the actual tapret data, created following the standard procedure outlined in BIP-341.

## AluVM

AluVM aims to provide a lightweight, deterministic environment for executing **Turing-complete scripts** that govern asset issuance, transfer, and other custom rules or business logic directly linked to Bitcoin.

It was created as part of the RGB protocol, where AluVM serves as the computational backbone, handling the logic and execution of smart contracts that manage various types of assets, while maintaining confidentiality and scalability. It plays a critical role in executing the client-side validation required for RGB's client-validated smart contracts.

**Figure 10: Comparison between AluVM and other blockchain execution environments**

	AluVM	Bitcoin script	EVM, kEVM, IELE	WASM	JVM, CLR	LLVM
Type	Register	Stack	Stack	Stack	Stack	Stack
Random memory access	No	No	No	Yes	Yes	Yes
Dynamic memory allocations	No	Yes	Yes	Yes	Yes	Yes
I/O operations	No	No	No	Via extensions	Yes	Yes
Turing completeness	Yes (bounded)	No	Yes (bounded)	Yes	Yes	Yes
Static analysis	Simple	Simple	Complex	Hard	Hard	Hard
Sandboxing	Always	Always	Always	Poor	No native	No native
Runtime environment	Any sandboxed	UTXO blockchain	Account-based blockchain	Internet	OS	Compiler
Library code immutability	Yes	No libraries	Yes	No	No	No
Undefined behavior (UB)	Impossible	Possible	Possible	Possible	Possible	Possible

Source: Binance Research, RGB Documentation

## AluVM vs. BitVM

**Figure 11: Comparison between BitVM and AluVM**

Property	BitVM	AluVM
<b>Interaction with Bitcoin</b>	Direct interaction with Bitcoin	Requires a bridge to interact with Bitcoin
<b>Underlying Asset</b>	Operates directly on BTC	Operates on RGB tokens
<b>Collateral Requirements</b>	No collateral required	Uses collateral in the form of other tokens
<b>Tokenization</b>	Not required	Requires “wrapping” BTC into RGB tokens
<b>Dispute Resolution</b>	Disputes resolved on-chain with computation	Not specified in detail
<b>Developer Tools</b>	Standard tools, in development	More advanced tools due to longer development
<b>Use Cases Ready</b>	Fewer ready-to-use applications	Several applications available for use
<b>Accountability</b>	Requires mutual accountability between parties	Does not require mutual accountability
<b>Operational Complexity</b>	Simpler in terms of operation with Bitcoin	More complex due to bridging and token management

Source: Binance Research

AluVM is designed specifically to operate on RGB tokens and not directly on BTC. To facilitate interaction with Bitcoin, a bridge is necessary to convert Bitcoin into an RGB token. The proposed method for this, called Radiant, involves using a separate token (e.g., Tether) as collateral. If an issuer of "wrapped bitcoin" fails to honor the original value during redemption, they forfeit their collateral. The returned value may also vary due to any gains or losses incurred during the use of RGB, and fees charged by the issuer.

In contrast, **BitVM interacts directly with Bitcoin** without the need for a bridge, collateral, separate tokens, or issuers. Participants simply deposit Bitcoin into a BitVM address that contains a specific program. This program, run off-chain by both parties, determines who receives the deposited Bitcoins. In case of disputes, the correct result can be enforced on-chain by executing part of the computation.

While both AluVM and BitVM are theoretically capable of running any computable function, AluVM has several practical advantages. It has been in development longer, resulting in better tools for developers, a variety of readily usable contracts, and does not require mutual accountability between participants. On the other hand, BitVM's main advantages include direct operation with Bitcoin and the elimination of the complexities associated with bridges, collateral, and issuers.

## Technical Limitations

While the RGB protocol offers significant benefits, it has certain technical and usability challenges to consider. .

### 1. Data Availability Issues

Typical users are **unable to create or acquire proofs of their transaction histories**. In scenarios where users are utilizing straightforward client interfaces, they may lack the means or infrastructure to retain comprehensive transaction records, complicating the process of furnishing proof of transactions to their counterparts.

### 2. P2P Network Reliance

RGB transactions extend Bitcoin's transaction mechanism and depend on a **separate peer-to-peer (P2P)** network (in this case, Tor) for distribution. This means when users execute transactions, they must engage interactively, with recipients needing to issue confirmations. This entire process hinges on a P2P network that operates distinctly from the Bitcoin network.

### 3. Virtual Machine and Language Development Hurdles

RGB smart contracts are written in Rust, which is a decently well-known language, however the RGB protocol's virtual machine, primarily AluVM, is relatively new and currently suffers from a lack of mature development tools and established code examples, making it challenging to onboard new developers.

### 4. Challenges with Public Contracts

RGB does not yet offer an effective interactive mechanism for ownerless or public contracts, posing difficulties for facilitating interactions among multiple parties.

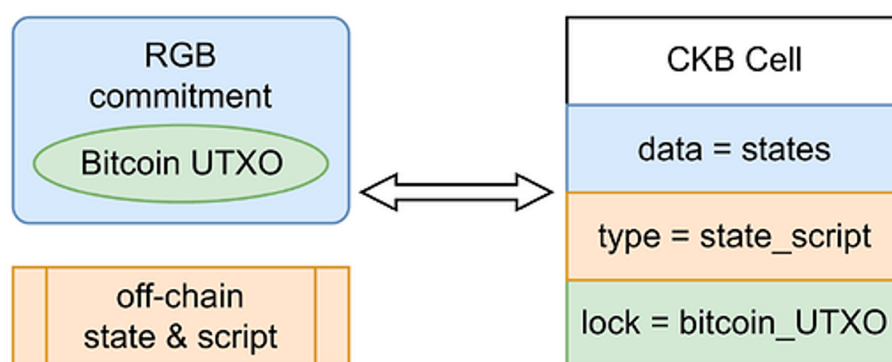
## RGB++

RGB++ is an evolution of RGB that was recently proposed by Nervos (CKB) in February 2024 that aims to solve the drawbacks of the RGB protocol. This new development has

brought more attention to the original RGB protocol, making it more commonly known among general crypto audiences.

RGB++ enhances the RGB protocol by introducing isomorphic binding, which directly connects Bitcoin's UTXOs with CKB Cells. This integration facilitates the **tracking of ownership and management of states** across both the BTC and CKB chains. This improvement allows all transactions to be verified on the CKB chain, streamlining the client validation process. It enables users to independently verify transactions using only the CKB chain, while still providing the option to utilize local Bitcoin transaction history for verification purposes.

**Figure 12: Isomorphic Binding in RGB++**



Source: Binance Research [\\_RGB++ Lightpaper](#)

The RGB++ transaction process utilizes off-chain computation to select seals and generate CKB transactions, which are confirmed through standard Bitcoin transactions using OP\_RETURN for embedding commitments. This method ensures transaction integrity and ownership validation across the BTC and CKB systems.

In client interaction, RGB++ eliminates the need for a dedicated client by enabling verification through Bitcoin and CKB light clients, simplifying user engagement and enhancing accessibility. The protocol advances in **managing shared states and non-interactive transfers**, streamlining multi-party operations and reducing the necessity for recipients to be online during transfers.

RGB++ enhances Bitcoin's functionality via the CKB chain, providing a Turing-complete environment for executing complex contracts and transactions. It uses isomorphic binding for token issuance and transfer, boosting privacy, transaction efficiency, security, and censorship resistance.



# Sidechains (kind of)

## Stacks

### Overview

The Stacks protocol was originally designed with the goal of extending the functionality of Bitcoin, and was not intended to be a L2 at the outset. While it is not definitively a sidechain, **Stacks is a blockchain that functions as a secondary layer for Bitcoin smart contracts**. Its initial (and current) version allowed for smart contract transactions that were technically faster and cheaper than Bitcoin L1 transactions. However, finality could only be achieved when Bitcoin blocks reached finality, i.e. 10 minutes per block, and ~6 confirmations. This is set to change with the upcoming **Nakamoto Upgrade, after which Stacks will more closely resemble a sidechain**.

The Stacks chain uses the **\$STX token** to incentivize miners and for transaction fees and relies on a novel **Proof of Transfer (“PoX”) consensus mechanism**. The STX token can also be “stacked” in order to earn BTC-denominated yield. Through PoX, the Stacks blockchain settles transactions on the Bitcoin L1, allowing Stacks transactions to benefit from Bitcoin’s security. However, given the \$STX token has been the primary incentive mechanism, it does not fully inherit Bitcoin security. Nonetheless, the relationship between Bitcoin security and Stacks is also set to strengthen with the upcoming upgrade.

### Functionality

Smart contracts on Stacks are written in **Clarity, a purpose-built language created for smart contracts on Bitcoin, and run on the Clarity VM**. The Clarity VM will allow for a range of smart contract capabilities similar to the EVM, include DeFi protocols, NFT marketplaces, gaming, and DAOs.

The Stacks team is also planning to introduce support for Clarity WASM in the future, which will allow for the possibility of smart contracts on Stacks being written in Rust and Solidity.

### Technical Architecture

#### Nakamoto Upgrade

The Nakamoto Upgrade is an upcoming hard fork designed to enhance the network's performance by increasing transaction throughput and ensuring closer Bitcoin finality. This upgrade will bring it **closer to the definition of a “Layer 2” network** on Bitcoin, by compressing multiple Stacks blocks within one Bitcoin block for faster confirmation on

Stacks transactions, using \$sBTC (Stacks token pegged to BTC) instead of \$STX as its native currency, and having the data of each Stacks block hashed and stored in a Bitcoin UTXO transaction (using OP\_RETURN).

*“The Nakamoto upgrade will bring Stacks closer to the definition of a “Layer 2” network on Bitcoin, by compressing multiple Stacks blocks within one Bitcoin block for faster confirmation on Stacks transactions, using \$sBTC (Stacks token pegged to BTC) instead of \$STX as its native currency, and having the data of each Stacks block hashed and stored in a Bitcoin UTXO transaction (using OP\_RETURN).”*

It will also make **forking and reorganizations of Stacks block impossible**, unlike before, as every new Stacks miner will need to include the Bitcoin commit transaction hash of the last Stacks miner in their block, which helps to ensure immutability of network history, since modifying the canonical state on Stacks would require modifying previous blocks on Bitcoin.

### **Proof of Transfer (“PoX”)**

Proof of Transfer (“PoX”) is the consensus mechanism used by the Stacks blockchain, which aims to leverage Bitcoin security. PoX involves participants spending Bitcoin to secure the network and distribute new Stacks (\$STX) tokens. This mechanism ties the security of Stacks closer to Bitcoin.

1. **Transferring Bitcoin:** Participants, known as miners, transfer Bitcoin to participate. Instead of using computational power to mine new blocks as in traditional proof-of-work systems, miners in PoX use Bitcoin to bid for the right to write new blocks and mint new STX tokens.
2. **Staking STX tokens:** STX holders can lock up their tokens to participate in Stacking. By locking their STX tokens for a certain period, users can earn Bitcoin rewards. This process supports network consensus by locking in economic resources.
3. **Reward Distribution:** Bitcoin sent by miners is distributed to Stackers. The Bitcoin that miners transfer as part of their mining activity is not kept by the network but instead is distributed to users who participate in Stacking. This distribution is based on the proportion of STX they have locked in relation to the total amount staked on

the network.

4. **Block Building:** The winning miner writes the next block. Miners are chosen based on a verifiable random function (“VRF”), ensuring that selection is fair and random. The chosen miner gets the right to write the next block on the Stacks blockchain and earns STX tokens as a reward for their Bitcoin expenditure.

## Limitations

The Stacks protocol was originally designed with the goal of extending the functionality of Bitcoin, and was not intended to be a Layer 2 at the outset. Its initial (and current) version allows for smart contract capabilities for Bitcoin, and while Stacks transactions were technically faster and cheaper than Bitcoin transactions, finality could only be achieved when Bitcoin blocks reached finality, i.e. 10 minutes per block, and ~6 confirmations.

In its current state, it also does not inherit much Bitcoin security, using its own native \$STX token as economic security on the network. However, as we highlighted above, this is set to change with the upcoming Nakamoto Upgrade.

## BounceBit

### Overview

BounceBit is a Proof of Stake EVM Layer 1 blockchain that utilizes bridged Bitcoin and its native token \$BB as the assets required for staking. There is no minimum stake or ratio of the tokens for each validator node.

BounceBit uses bridged Bitcoin assets such as wBTC and BTCB on BounceBit, which is currently supported by third-party bridges like MultiBit, Polyhedra, and LayerZero when the BounceBit mainnet is launched to transfer assets between these platforms. We should note that these wrapped versions of BTC are relatively centralized at this point in time, and may pose a potential risk to the security of the network.

BounceBit is using CometBFT for network consensus, a fork of Tendermint Core that implements Byzantine Fault Tolerance, whereby **security is achieved as long as no more than one-third of the machines fail**. This consensus model has been widely adopted and used by other live protocols such as Cosmos, Evmos, and Celestia, making it a relatively battle-tested and reliable option.

## Functionality

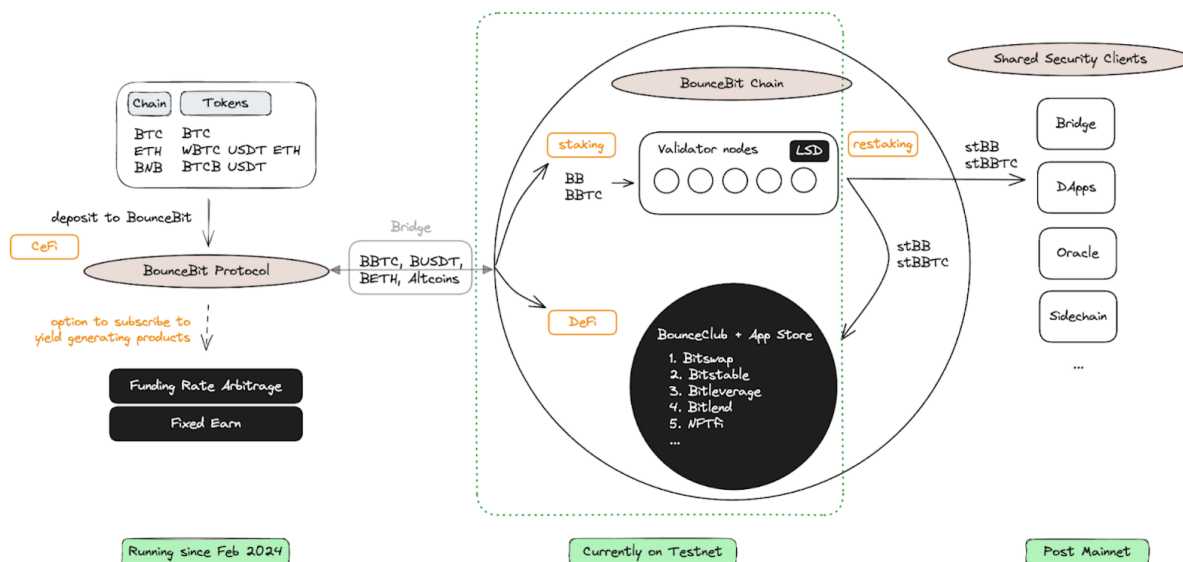
Aside from being a fully functioning EVM, BounceBit also allows users to restake their wrapped BTC tokens on the BounceBit protocol to receive yield from their Shared Security Clients. This is a feature only seen so far in Babylon, a Bitcoin restaking-focused protocol.

## Technical Architecture

Users deposit a variety of tokens such as BTC, ETH, USDT, along with their bridged forms like wBTC and BTCB into the BounceBit protocol through centralized finance (CeFi) methods. These tokens are then converted into the BounceBit chain's corresponding forms, such as BBTC, BUSDT, and BETH, among other altcoins.

Once within the BounceBit ecosystem, these assets are eligible for staking, where they are transformed into BB or BBTC tokens. The staked assets are managed by network validator nodes that partake in the consensus process. Through a process known as Liquid Staking, these assets can be re-staked, resulting in new denominations—stBB and stBBTC. These re-staked assets are then capable of interacting with various shared security clients (SSCs) such as bridges, decentralized applications (DApps), oracles, and sidechains.

**Figure 13: Restaking in the BounceBit protocol**



Source: Binance Research, BounceBit Documentation

## Bitcoin Restaking

BounceBit is one of the only Bitcoin scaling protocols also attempting to conduct Bitcoin restaking through its platform, aiming to provide yield to BTC holders through both CeFi and DeFi methods.

Functionally, **SSCs in BounceBit are akin to AVs on EigenLayer**, where the yields produced from these operations are redistributed back to the users who have staked their LSD tokens on these platforms.

## Limitations

It is important to distinguish that BounceBit should not be classified as a Bitcoin sidechain, and technically not a Layer 2 based on our definition, as it **does not post transaction data back to the Bitcoin mainnet**, a feature that is seen in other protocols such as Merlin Chain, where zero-knowledge proofs of transactions are integrated into Bitcoin's Taproot Script, or have the hash of its blocks stored in Bitcoin blockspace, like Stacks. This implies that BounceBit does not inherently possess the economic security features associated with Bitcoin.

# ZK-rollups

## Merlin

### Overview

Merlin is a zk-rollup-based scaling solution on Bitcoin that uses **Polygon CDK for its underlying zkEVM** infrastructure. Zero knowledge proof generation is currently centralised and outsourced to Lumoz, a third-party zk-RaaS provider, though in the future, with the launch of Lumoz's mainnet, Merlin Chain will be connected to Lumoz's decentralized ZK computing network.

Merlin is also **utilizing BitVM to power its on-chain fraud proof mechanism** on Bitcoin, with ZK proofs of transactions on Merlin being posted onto the Bitcoin mainnet via Tapleaf Script and stored permanently, though with all projects utilizing BitVM, this mechanism is still a work in progress.

### Functionality

Merlin runs a Type 2 zkEVM, which makes it **fully Ethereum-equivalent**. State transitions can also be represented through circuits and batched into zk-proofs that can be verified publicly. However, proof generation times may be slow, a weakness among all Type 2 zkEVMs.

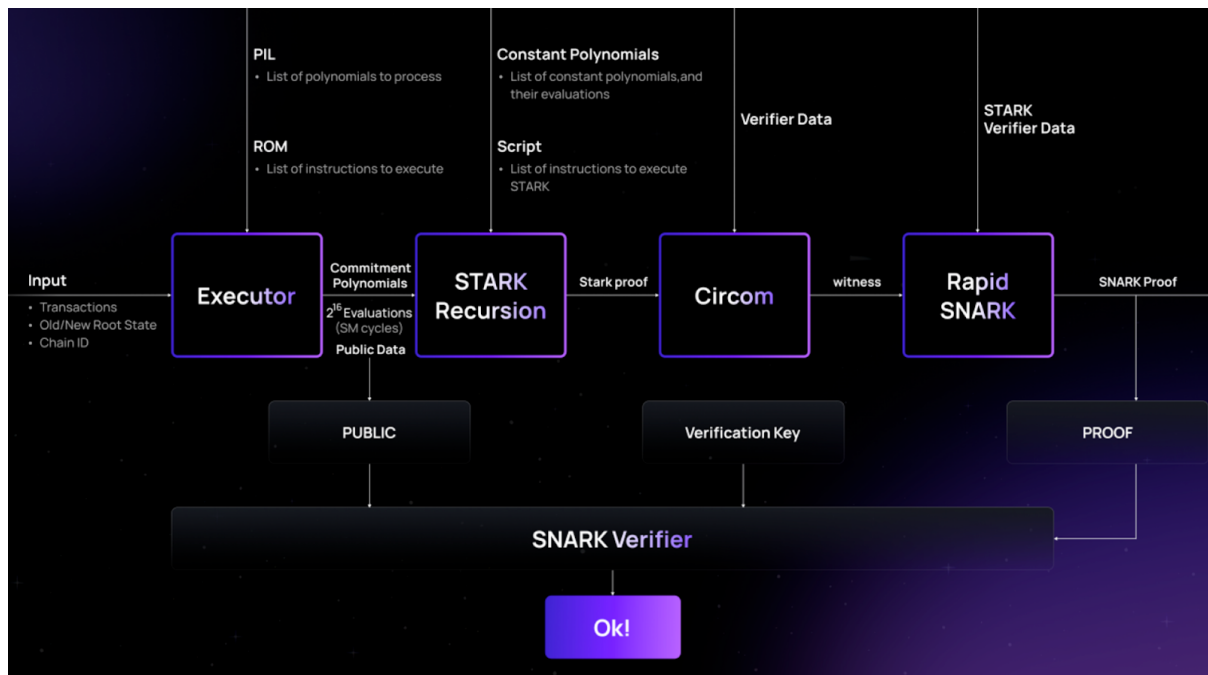
## Technical Architecture

### zkProver

Central to Merlin's technical architecture is the zkProver, a crucial component responsible for generating zero-knowledge proofs. These proofs verify transactions while keeping the underlying data private. zkProver operates interactively with network nodes and databases, retrieving necessary transaction data such as Merkle roots and hash values, which it then uses to generate verifiable transaction proofs. These proofs are subsequently returned to the nodes, ensuring the integrity and privacy of the transaction process.

Further supporting the zkProver's operation is a robust system of Finite-State Machines (FSMs). This system features a main FSM alongside multiple auxiliary FSMs, each specialized to handle various aspects of proof generation including binary operations, memory management, and cryptographic functions. This modular design enhances the efficiency of proof generation, and also ensures a high degree of accuracy and security.

**Figure 14: ZK Proof generation on Merlin**

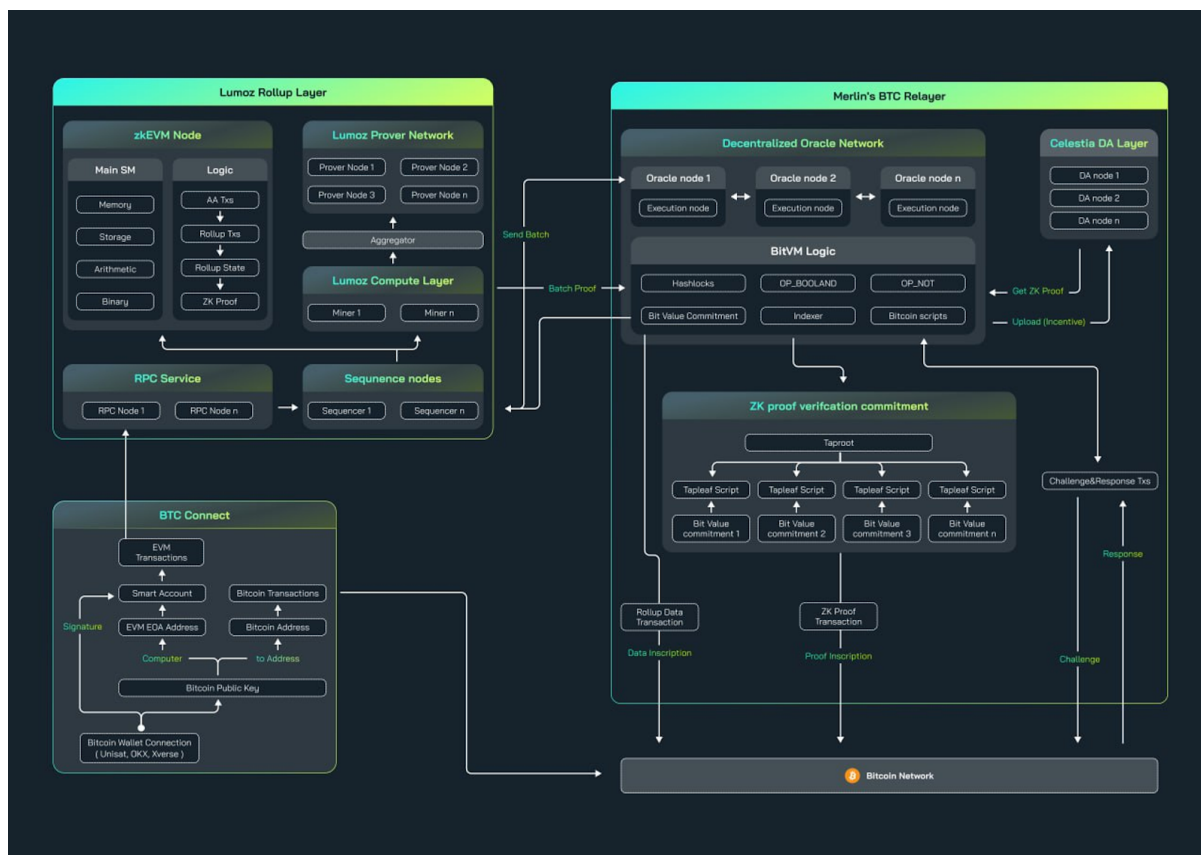


Source: Merlin Chain documentation

Additionally, Merlin's architecture utilizes two programming languages: Zero-Knowledge Assembly (zkASM) and Polynomial Identity Language (PIL). zkASM is tailored for mapping instructions directly to FSMs, thereby facilitating precise and efficient transaction processing. On the other hand, PIL is used to express calculations in the form of polynomial identities, which are crucial for verifying the correctness of state transitions within FSMs.

## Decentralised Oracle Network

Figure 15: Merlin Chain’s Proposed Technical Architecture



Source: Merlin Chain documentation

To ensure reliable data posting from the Merlin Chain to Bitcoin, Merlin aims to run a decentralized oracle network, whereby **oracle operators stake BTC on the network** and allow other parties (verifiers) to check and verify the proofs based on public transaction data.

Sequencer nodes on Merlin gather and batch transactions, while the zkProver generates the necessary proofs. Concurrently, raw transaction data, Merkle trees, Bitcoin state, and other relevant data are combined into a comprehensive proof, which is coordinated with the Oracle network.

Once the transaction data is ready, the Oracle network undertakes circuit compilation and proceeds to **upload the consolidated data and commitment proofs to the Bitcoin mainnet**. This data is embedded into Bitcoin Taproot, ensuring it is publicly available and verifiable by anyone within the network.

## Limitations

In its current iteration, raw transaction data is currently stored on a centralized settlement layer, due to the inability of Bitcoin to store data of this format, and Celestia being inoperable with Polygon's zkEVM yet, but decentralizing data availability is a direction that the team is moving towards.

Merlin is also currently running a multi-signature bridge to move assets to and from Merlin Chain and Bitcoin. Wallets on the Bitcoin layer are managed through MPC by several addresses controlled by the team and custody managers, and there is currently \$1.2 billion worth of BTC locked in these contracts.

It is important to note that all zk-rollup projects on top of Bitcoin are unable to attain the level of Bitcoin security as ZK Layer 2s do on Ethereum. This is due to **Bitcoin's inherent inability to conduct computations natively on-chain**, unlike Ethereum's Solidity contracts that are able to verify the validity of ZK proofs on mainnet.

## Citrea

### Overview

Citrea is a Type 2 zkEVM built using RISC Zero. It is fully EVM equivalent, and utilizes a scalable and trustless proof system based on zk-STARKs. Similar to Merlin, it also makes use of BitVM to conduct the on-chain fraud proof mechanism for its zk-proofs, and has also taken the liberty of using BitVM to design a trustless light client bridge. This additional complexity has caused Citrea to slightly lag behind in terms of mainnet launches, as compared to other Bitcoin ZK scaling solutions.

### Functionality

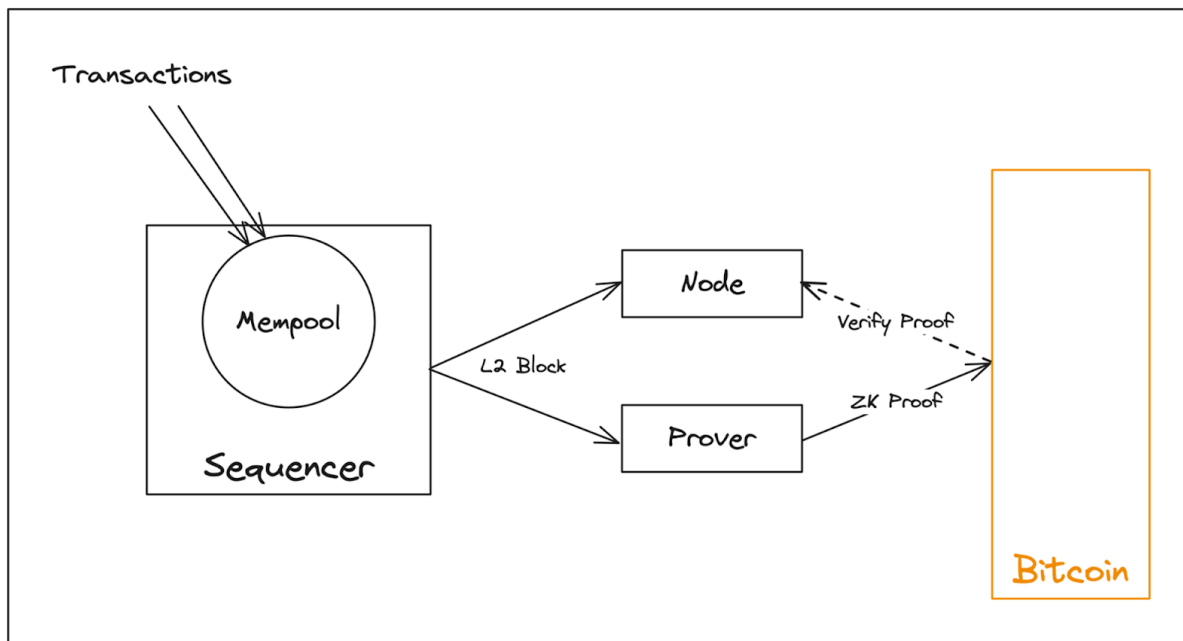
Aside from being a Type 2 zkEVM, **Citrea has the potential to implement other execution environments as well, such as WASM or the Solana VM**. This was an intentional choice made by the team to allow for greater compatibility, hence the choice of RISC Zero as its foundational layer, a general purpose zkVM.

### Technical Architecture

#### Block Production



**Figure 16: How Citrea's Sequencers Communicate with Bitcoin**



Source: Citrea Documentation

In Citrea, the role of producing blocks is handled by Sequencers. Unlike validators or miners in other blockchains, **sequencers in Citrea do not require validation from others** for the blocks they produce. This is because **each block undergoes a zero-knowledge proving process**, serving as a trustless validation mechanism that ensures the integrity and authenticity of the blocks.

The sequencer receives blocks using its local mempool, and is responsible for ordering and publishing them. Utilization of BitVM's fraud proof mechanism, the force transaction mechanism, and on-chain data availability prevent the sequencer from misappropriating or freezing user funds.

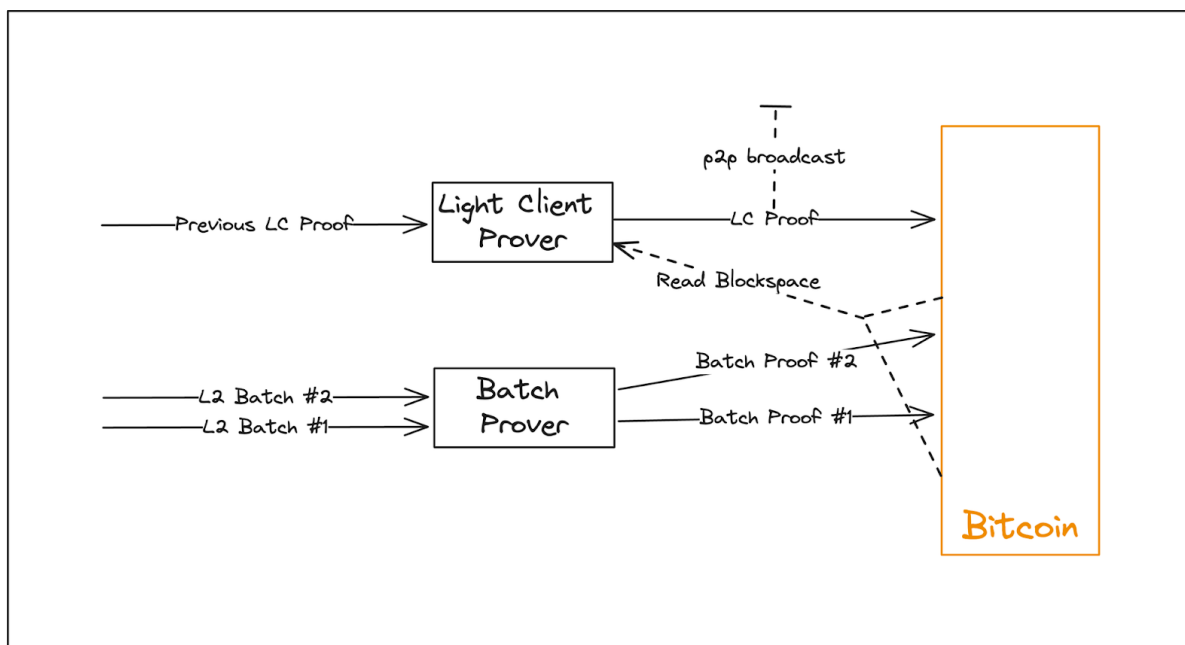
To increase the system's robustness and reduce the risk of censorship, Citrea is developing a solution that **allows multiple sequencers to produce and finalize blocks** almost instantly. This multi-sequencer approach minimizes the need for users to rely on Bitcoin's fallback mechanism for force transactions and ensures that no single sequencer can manipulate transaction ordering.

Transaction ordering is guaranteed only up until the next Bitcoin block is confirmed. Every 10 minutes, the Merkle root of the batched transactions is inscribed in Bitcoin, locking in the order of transactions within the Citrea network. This inscription validates the state root and ensures that **transaction ordering remains immutable once recorded on Bitcoin**.

In future, Citrea plans to implement a multi-sequencer network to reduce trust assumptions in the sequencing process, aiming for near-instantaneous finality of transaction ordering while maintaining minimal data publishing costs.

## Proof Generation

**Figure 17: Generating Light Client and Batch Proofs**



Source: Citrea Documentation

Citrea's use of a recursion-capable STARK-based zkVM called RISC Zero generates two kinds of proofs for the network:

1. **Batch Proof:** These are generated periodically for every few Bitcoin blocks. Citrea's circuit scans for batch roots in Bitcoin blocks, validating the corresponding L2 batches and outputting crucial data such as state differences, initial and latest state roots, and the scanned block's hash. This output is then recorded in Bitcoin.
2. **Light Client Proof:** Designed for lightweight and trustless nodes, these proofs recursively validate batch proofs, providing a comprehensive view of the entire rollup history. By processing a sequence of batch proofs and their related Bitcoin block headers, the circuit ensures continuity and accuracy of the state root throughout the rollup's history.

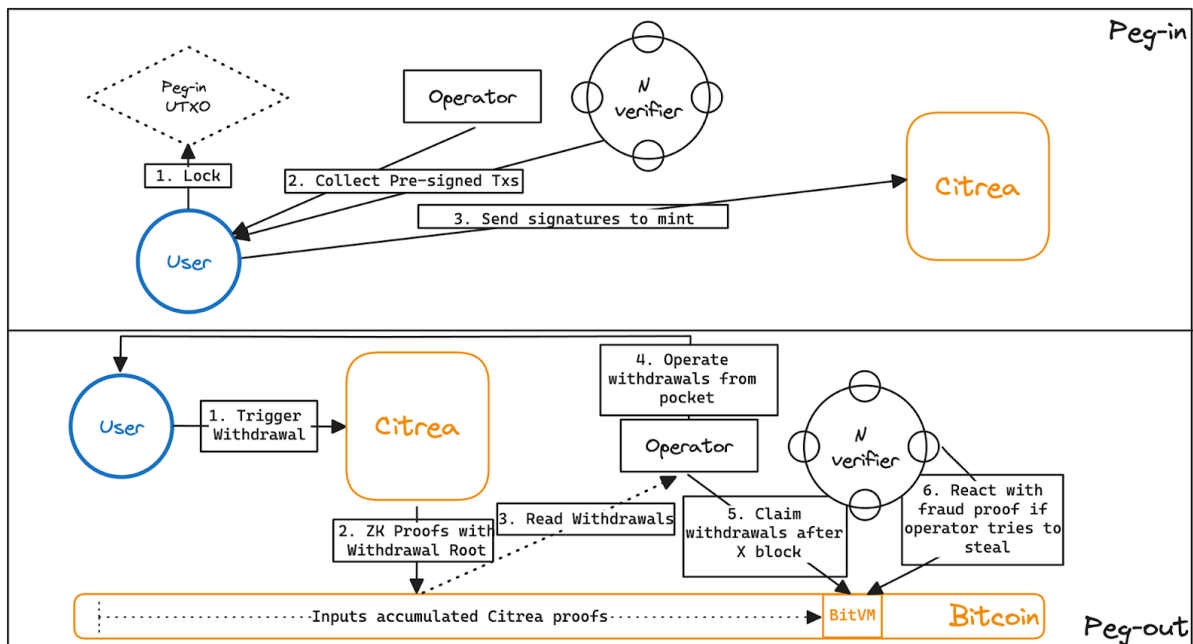
The core functions of Citrea's proof system include:

- **Execution Proving:** This process inputs the pre-state and new batches into the Citrea circuit to validate and compute state transitions, ensuring the integrity of each state change.
- **Blockspace Proving:** A new concept in Citrea, this involves scanning Bitcoin blocks to extract and verify Citrea batch proofs and state roots, ensuring their accuracy and authenticity.

By merging these processes in a single circuit for batch proofs, Citrea allows full nodes to verify state transitions. Light client proofs enable any user with access to Bitcoin block headers or the peer-to-peer network to trustlessly verify the rollup's entire history.

### Citrea's Trust-minimized Bridge with BitVM

**Figure 18: Proposed Technical Architecture of Citrea's Trust-minimized Bridge**



Source: Citrea documentation

Citrea's light client proofs are verified within Bitcoin using BitVM by employing a multi-verifier setup that enhances security for peg-in and peg-out transactions. In this system, an operator handles the transactions while multiple verifiers oversee and check for any invalid activities. The security of the peg is guaranteed as long as at least one of these verifiers remains honest, a notable advancement over traditional bridge models that rely on a majority consensus.

The BitVM setup **allows for immediate withdrawals without delays** once the proofs are validated in Bitcoin's optimistic scenarios. The operator funds these withdrawals upfront and later claims the equivalent BTC from the BitVM program, providing proof that the

transactions correspond with the activities on the Citrea chain. Should any fraudulent activities be detected, verifiers can intervene by submitting fraud proofs to Bitcoin, which in turn secures the peg by slashing the stakes of dishonest provers.

The BitVM contract is responsible for verifying several critical aspects:

- Light Client proofs that are recursively merged and include deposit and withdrawal roots.
- A Bitcoin Header Chain proof that demonstrates the latest block header and a Merkle tree of previous headers, similar to those used in ZeroSync.
- Bitcoin SPV proofs confirming that all withdrawals have been financially covered by the operator.

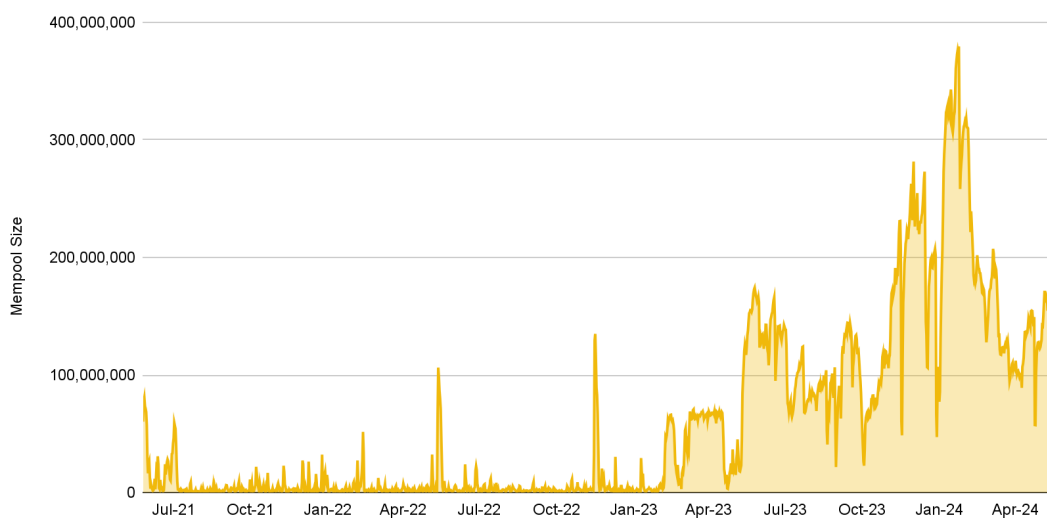
To optimize efficiency and minimize the size of the program committed on Bitcoin, Citrea's verification logic is encapsulated within two Groth16 circuits, with the **BitVM program operating as a single Groth16 verifier** pre-configured with the circuit's verifying key.

This two-way peg architecture is designed to be trust-minimized and is currently under intensive development. This system does not require changes to the Bitcoin network but may necessitate opcode adjustments to achieve full trustlessness in settling transactions on Citrea.

## 4 Outlook and Closing Thoughts

As Bitcoin expressivity continues to forge its path, and DeFi primitives such as stablecoins, money markets, staking & restaking, and perpetuals emerge, the importance of Bitcoin L2 solutions will continue to grow. As we previously highlighted, Bitcoin’s transaction fees are significantly higher than in the last few years, while its mempool continues to get busier.

**Figure 19: Bitcoin’s mempool has been getting increasingly populated since the first Ordinals boom in 2023**



Source: Binance Research, Blockchain.com, as of May 25, 2024

At this early stage of the Bitcoin L2 ecosystem, state channels like Lightning are perhaps the only protocols that come close to the widely accepted definitions of a “true L2”. However, these have clear limitations in terms of user tooling and functionality. A new wave of projects are getting close too, but are yet to reach their final stage.

zkEVM rollups that use BitVM seem to be the most promising at this point. Nonetheless, most are not yet close to reaching full production-level, especially considering that BitVM remains in a developmental stage. A potential ideal solution might be for the Bitcoin protocol to add native opcodes for verifying zero-knowledge proofs, something that is currently being worked on by teams including ZeroSync (also behind BitVM). This may potentially allow for verifiable zk-rollups on Bitcoin in the future. An exciting time ahead for Bitcoin scalability solutions, with lots of development expected over the next few months.

***This is part three of our new *The Future of Bitcoin* series. Keep an eye out for the next one!***

# References

1. <https://www.blockchain.com/explorer/charts/n-transactions-total>
2. <https://coinmarketcap.com/>
3. <https://help.blockstream.com/hc/en-us/articles/900003013143-What-is-the-Liquid-Federation>
4. <https://bitvm.org/bitvm.pdf>
5. <https://x.com/udiWertheimer/status/1782784004120338609>
6. <https://bitcoinmagazine.com/business/1-5t-morgan-stanley-is-buying-us-spot-bitcoin-etf>
7. <https://www.wpr.org/news/wisconsin-pension-fund-bitcoin>
8. Taproot - <https://github.com/bitcoin/bips/blob/master/bip-0341.mediawiki>
9. <https://zerosync.org/>
10. <https://l2.watch/>
11. <https://docs.citrea.xyz/>
12. <https://www.bitcoinlayers.org/>
13. <https://www.btceden.org/>
14. <https://github.com/chainwayxyz/bitvm-zk-verifier>
15. <https://lightning.network/>

# Latest Binance Research Reports



## Breakthrough DeFi Markets

An in-depth analysis of the emerging trends transforming DeFi



## Low Float & High FDV: How Did We Get Here?

A review of recent token trends



## The Future of Bitcoin #2: Tokens

A crypto-centric review of 2023



## The Future of Bitcoin #1: The Halving & What's Next

A look at the 2024 Bitcoin Halving, potential impacts on Bitcoin's key metrics, the mining industry, and more

# About Binance Research

Binance Research is the research arm of Binance, the world’s leading cryptocurrency exchange. The team is committed to delivering objective, independent, and comprehensive analysis and aims to be the thought leader in the crypto space. Our analysts publish insightful thought pieces regularly on topics related but not limited to the crypto ecosystem, blockchain technologies, and the latest market themes.



## Shivam Sharma

### Macro Researcher

Shivam is currently working for Binance as a macro researcher. Prior to joining Binance, he worked as an investment banking associate and analyst at Bank of America on the Debt Capital Markets desk, specializing in European financial institutions. Shivam holds a BSc in Economics degree from the London School of Economics & Political Science (“LSE”) and has been involved in the cryptocurrency space since 2017. Follow him on X: [@Sh\\_ivam](#).

## Chloe Tan

### Technical Analyst

Chloe serves as a Technical Analyst at Binance, evaluating the feasibility and security of protocols. She has a passion for exploring the intricacies of blockchain technology, particularly in the areas of cryptography, zk-proofs, and privacy.



# Resources



Read more [here](#)



Share your feedback [here](#)

**General Disclosure:** This material is prepared by Binance Research and is not intended to be relied upon as a forecast or investment advice and is not a recommendation, offer, or solicitation to buy or sell any securities or cryptocurrencies or to adopt any investment strategy. The use of terminology and the views expressed are intended to promote understanding and the responsible development of the sector and should not be interpreted as definitive legal views or those of Binance. The opinions expressed are as of the date shown above and are the opinions of the writer; they may change as subsequent conditions vary. The information and opinions contained in this material are derived from proprietary and non-proprietary sources deemed by Binance Research to be reliable, are not necessarily all-inclusive, and are not guaranteed as to accuracy. As such, no warranty of accuracy or reliability is given, and no responsibility arising in any other way for errors and omissions (including responsibility to any person by reason of negligence) is accepted by Binance. This material may contain 'forward-looking' information that is not purely historical in nature. Such information may include, among other things, projections and forecasts. There is no guarantee that any forecasts made will come to pass. Reliance upon information in this material is at the sole discretion of the reader. This material is intended for information purposes only and does not constitute investment advice or an offer or solicitation to purchase or sell any securities, cryptocurrencies, or any investment strategy, nor shall any securities or cryptocurrency be offered or sold to any person in any jurisdiction in which an offer, solicitation, purchase or sale would be unlawful under the laws of such jurisdiction. Investment involves risks.