BitVM Status Report

February 2025

BitVM Overview

BitVM is a mechanism to execute arbitrary programs on Bitcoin in an optimistic manner: the execution happens off-chain but in case of failures, disputes are resolved and enforced on-chain. Think Optimism, but on Bitcoin.

The two main use cases are Bitcoin rollups and trust-minimized bridges. In both cases, we want to allow users to deposit and withdraw BTC from an L2 without trusting a 3rd party. With BitVM we can ensure BTC deposits cannot be stolen as long as there is a single honest and online node in the network - this node can be the depositor themself.

The latest and practical version is BitVM2. Please refer to the latest paper for a full protocol specification of the canonical BitVM2 protocol.

BitVM2 Protocol ELI5

1. Compress a program into a SNARK verifier, implemented in Bitcoin Script. Groth16 is approximately 1GB in size.
2. Split the verifier into sub-program chunks, max 400KB each (each can be run in a Bitcoin transaction).
3. The operator commits to the program during setup.
4. When attempting to withdraw funds from BitVM2, the Operator can be challenged by anyone (e.g. if the peg-out was wrong).
5. If challenged, the Operator must reveal all intermediary program results.
6. If the Operator is cheating, one of the claimed sub-program results will be wrong. Anyone can disprove the Operator by executing that specific sub-program in a Bitcoin transaction, showing that the Operator claimed a fake computation.
7. Done. The faulty Operator is kicked out and cannot access the BitVM funds (invalidated spend transaction).

BitVM2 Bridge Protocol ELI5

The BitVM Bridge makes use of BitVM2 to implement a bi-directional light-client bridge between Bitcoin and an L2/sidechain: the L2 verifies Bitcoin, Bitcoin verifies the L2. The most interesting part is the peg-out (un-wrapping).

1. Operators pay BTC to the withdrawing user from their own funds and then reclaim the BTC from BitVM.
2. BitVM checks that for a un-wrap transaction on the L2, there is a correct peg-out on Bitcoin.
3. If all is correct, the Operator gets the BTC refunded.

BitVM1 vs BitVM2. For comparison:

- BitVM1: runs any RISC-V program and uses an interactive bi-section protocol to detect faults.
    - Up to 70 transactions to complete a challenge over multiple months.
    - Permissioned challenging: only a fixed set of Operators to challenge each other.
    - Security: secure as long as one of the pre-selected operators is honest.
- BitVM2: runs any program that can be represented in a SNARK verifier. Uses a one-shot, non-interactive protocol to disprove faulty claims.
    - Only 3 transactions to complete challenge within similar timeframes as ETH L2 finality periods.
    - Permissionless challenging: anyone can challenge.
    - Security: as long as there is 1 honest node in the network, the BTC cannot be stolen ("existential honesty"). Anyone can run their own node.

Recently, improved flavors of BitVM1 have been introduced by BitVMX that reduce the number of challenge/response transactions.

Timeline. First prototypes went live in Q4 2024, and public testnets that can be tested by users and operators alike are rolling out in Q1/Q2 2025. Mainnet deployments are likely to go live in the second half of 2025.

Challenge: Light Clients. One of the biggest challenges and complexities of implementing a BitVM bridge is verifying the connected L2 from within BitVM. This requires implementing a light client for the L2's consensus in BitVM. Given how BitVM is designed and the limited tools we have on Bitcoin, BitVM has no state or storage, i.e., the light client must be stateless. This means that we either compress the L2 state to generate a stateless proof (proving the L2 from a certain pre-agreed / trusted checkpoint or genesis block), or regularly update the BitVM instance to agree on the latest version of the L2 state. The latter works well with fixed-size challenger sets, as e.g. in BitVM1, but is impossible if we wish to achieve permissionless challenging as in the latest version of BitVM2. Multiple high potential approaches are being actively researched. As of today, the practical way forward is to introduce a fixed-size set of pre-defined challengers (similar to “watchtowers” in the Lightning Network), and then eventually upgrade to a more robust light client design.

Roles in BitVM2

Operators: Responsible for operating the BitVM bridge.

- Run node software that monitors BTC deposits and withdrawals triggered on the L2, and participates in the setup process for BitVM each time a new deposit is requested.
- Operators lock up collateral[1] to cover fees paid by users in case of a challenge due to faulty operation.
- When users withdraw from the L2, operators front the capital, sending BTC to the withdrawing users from their own funds, and then reclaiming the BTC from the BitVM deposits, including fees. The duration of this will range between 7 and 14 days initially, similar to ETH L2 bridge finality times. This is similar to liquidity bridge mechanics on Ethereum L2s.

Challengers: Ensure the safety of the peg-out process by challenging an operator in case of misbehavior. Anyone can act as a challenger, including the operators. Challengers will initially have to run additional software. Eventually we expect this software to potentially be included in wallets and browsers, reducing friction.

Covenant Committee: A committee of n signers that is responsible for the correct setup of a BitVM instance. One of the n signers is assumed to be honest (existential honesty). This is a temporary necessity until Bitcoin supports covenants - the committee ensures that operators can only withdraw BTC from the BitVM deposit in a way that can be challenged, enforced via pre-signed BTC transactions. We anticipate that this setup committee can be operated similar to the Ethereum KZG ceremony, on a continuous basis.

LPs / Depositors: LPs deposit BTC into BitVM to mint wrapped BTC on the L2. For this, LPs participate in the setup process of BitVM, interacting with operators. Anyone can be an LP but we expect this role to be taken up by DeFi funds and companies running searchers for ETH liquidity and intent bridges. LPs then swap wrapped BTC against BTC on Bitcoin L1 with retail users that would otherwise struggle with the process.

Each deposit requires a new BitVM instance with a fixed size. Withdrawals require the LP or any user to have the exact amount of wrapped BTC as locked in the BitVM instance. This means

[1] Collateral can range from 0.1BTC to 1BTC, depending on the setup. Economic models are still being developed, more in a follow up report.

that LPs can offer this as a service to users and re-balance wrapped BTC vs L1 BTC as a service. Note: improvements to this are being developed, so by the time BitVM is live on mainnet, this might no longer be an issue.

Users: Users want to use BTC on L2s but require a simple UX. We expect most users to not be technical enough to participate in the BitVM setup/deposit process and rather use swaps to onboard into L2s, swapping BTC on L1 against wrapped BTC on the L2 offered by LPs.

BTC Rollup vs BTC Sidechain… and Bridges

Amidst the hype of Bitcoin L2s we believe it is important to clarify the different approaches and their feasibility.

The goal of Bitcoin L2s is to achieve 2 things:

1. Bitcoin security
2. Trust-minimized BTC bridge

The good news is that a (2) trust-minimized BTC bridge is finally possible via BitVM2. As explained above, BitVM allows us to create a so-called "light-client" bridge: the L2 verifies Bitcoin, Bitcoin verifies the L2. This means that the BitVM bridge is secure as long as:

- Bitcoin is secure (this is a given),
- The L2 is secure (discussed below),
- There is at least 1 honest, online node to trigger challenges / fraud-proofs.

The bad news is that while there are different levels of theoretical (1) Bitcoin security, few are practical today or are likely to be practical in the next 12 months. We discuss the current state and trade-offs below.

- **"ZK Rollups":** Run a full ZK verifier on Bitcoin.
    - **Good**: Continuous full validation of every state transition by every Bitcoin full node. Only limited DA requirements.
    - **Bad**: Not possible without OP_CAT or another Bitcoin op-code change, and even then likely too expensive in practice.
    - **Comments**:
        - "ZK rollups on Bitcoin" do not yet work. Today, we can only create optimistic rollups on Bitcoin via BitVM, i.e., not ZK validity rollups as we know them from Ethereum.

- **Optimistic rollups**: Arbitrum-style fraud proofs via BitVM.
    - **Good**: Optimistic verification by all Bitcoin full nodes in case of a dispute. Can be implemented with BitVM without a Bitcoin fork.
    - **Bad:** To achieve full Bitcoin security, we would need to use Bitcoin as a DA layer which will likely be too expensive (especially in the early days when Bitcoin L2s don't yet have millions of users).
    - **Comments**
        - BitVM2 uses a ZK compression step for practical reasons. This has resulted in some misunderstandings with respect to incorrectly calling BitVM2-style optimistic rollup designs "ZK rollups".

- **Optimiums**: Optimistic rollup that uses a different DA layer.
    - **Good:** Low DA costs. Easy to implement (e.g. Celestia out-of-the-box support for most rollup stacks).
    - **Bad**: Security is reduced to the security of the DA layer. Proper security requires running a light client for the DA layer in BitVM to check that the data was posted. Without this logic a bad sequencer could simply not post the data to successfully attack the chain.
    - **Comments**:
        - One can use a DA layer that has some level of Bitcoin security e.g. a merged-mined or BTC-staked sidechain. This makes it easier to verify the consensus of the DA layer from within BitVM, which is necessary to maintain security guarantees against sequencer-level attacks.
        - The most important question is: what is the point of running an optimium if the security is the same as that of a DA Bitcoin sidechain. Benefit: readiness to eventually switch to a full optimistic rollup by posting data to Bitcoin. Limitations: higher technical complexity early on in the chain's lifecycle.

In theory, these rollup models could enable a so-called "escape hatch" where users could force-exit back to the L1 if the Sequencer goes down. However, as opposed to Ethereum (and even there it doesn't fully work yet), there is no blueprint for implementing this on Bitcoin and it is not yet clear if this is at all feasible with Bitcoin's UTXO model. As of today, if all operators go offline in BitVM, funds will remain frozen. This may be resolved by the activation of some Bitcoin covenant opcodes in the future.

- **Bitcoin sidechain (merged mining)**

    - **Good:** Verification of the L2 state by a **subset** of Bitcoin network participants (Bitcoin miners). Contributes directly to Bitcoin’s fee sustainability.
    - **Bad:** Often centralized across a few large pools. Initially pools might not even pass on the rewards to their miners. Lack of clear economic benefit for the network (alignment with miners - yes, but hard to quantify the support in liquidity/network growth). No strong finality.
    - **Comments:**
        - BOB built a prototype with Marathon Digital and developed a more efficient merged-mining version compatible with L2 block production speeds (Optimine). RSK has been using merged mining to secure their sidechain with 60%+ of the Bitcoin hash rate for over 5 years.

- **Bitcoin sidechain (BTC staking via Babylon)**
    - **Good:** Verification of the L2 state by a **subset** of Bitcoin network participants (Bitcoin stakers). Easy enough to verify in BitVM. Strong economic alignment with BTC LSTs and BTC stakers: the more TVL, the more volume, the more fees accrue back to BTC stakers → potential flywheel effect.
    - **Bad:** Technical risk due to novelty of tech.