Bitcoin Vaults Liquidation Engine

BOB Research, October 2025

Dominik Harz
research@gobob.xyz

Abstract

We present a novel trust-minimized liquidation engine for Bitcoin vaults that enable native BTC as collateral for lending and stablecoins on DeFi chains like Ethereum, Base, BOB, Solana, and others. Our solution addresses limitations in BitVM-style Bitcoin vaults, where liquidations take several days to process, the entire BTC amount must be liquidated, and the liquidator set is static and predefined. The Bitcoin Vault Liquidation Engine restores atomic liquidations with an open liquidator set as used by lending protocols today. The liquidation engine maintains trust minimization for the depositor: the depositor needs to trust the Bitcoin and DeFi chain consensus, as well as the correct implementation of the DeFi protocol and the Bitcoin vault. The depositor can (1) enforce withdrawal of their BTC via a proof powered by BitVM on Bitcoin without having to trust bridge operators, and (2) receive exact BTC from their deposit UTXO, offsetting concerns around receiving BTC from unknown third parties.

1 Introduction

Bitcoin lending represents a critical component of the evolving DeFi ecosystem. However, the unique constraints of Bitcoin's scripting language and security model have historically limited the development of sophisticated lending protocols that maintain the trust-minimized properties that define Bitcoin. Likewise, since the early days of Ethereum, there was a desire to make BTC usable within its ecosystem. Early attempts like BTC Relay[1] pioneered the concept of verifying Bitcoin transactions on Ethereum, laying groundwork for future bridge designs. Today, BTC is widely used as collateral to borrow stablecoins such as USDT or USDC and to mint new stablecoins using CDP-style designs. Almost 50% of wBTC on Ethereum is used in lending. However, current BTC bridges do not allow participants in DeFi protocols to receive precisely the BTC they deposited back, and they have no way to enforce the return of their BTC. There is a large untapped market of users who want to keep their BTC on Bitcoin, secured by Bitcoin, but still participate in lending on other chains, for example, borrowing USDC or USDT against native BTC on Ethereum, BOB, Base, Arbitrum, Solana, and other chains.

With BitVM [5–7], we can create trust-minimized vaults that allow anyone to participate in DeFi in other chains with their native BTC [4]. Compared to Bitcoin wrappers like wBTC and cbBTC, Bitcoin vaults ensure that, under normal operations, the depositor of the BTC (1) can enforce a withdrawal from the bridge by submitting a cryptographic proof to Bitcoin without requiring cooperation from bridge operators and (2) receive exactly the BTC they deposited from one of their UTXOs back into their wallet, improving concerns around potentially receiving tainted UTXOs when using a fully fungible wrapper.

To achieve this, the depositor transfers BTC into a BitVM-style vault, creating an NFT on the DeFi chain that represents the value of the BTC locked in the vault. The depositor then uses this NFT as collateral to, e.g., borrow USDT or USDC from a lending protocol or mint stablecoins backed by the BTC collateral. The depositor can withdraw the BTC at any time, without the need for a third party, by generating a ZK proof from the DeFi protocol that the loan was repaid or the minted stablecoins burned. This differs from a fungible BTC bridge, where the depositor must trust the operators (e.g., custodians in centralized bridges or operators in BitVM) to facilitate withdrawals. **In the Bitcoin vaults, the depositor has no trust assumption in a third party**.

A lending protocol needs to accept the minted NFT as collateral. However, since NFTs the non-fungible, liquidations require careful consideration. Typically, a lending protocol relies on liquidators to return the loan amount, e.g., the borrowed USDC or USDT. In return, they receive the underlying collateral. Babylon has recently demonstrated on Ethereum mainnet a Bitcoin vault that was used to lock collateral used in a Morpho lending pool including the liquidation of the loan [9]. In the Babylon design, a pre-defined set of liquidators is added as recipients to the Bitcoin vault: if the depositor returns the loan, they can withdraw the BTC. If the depositor's loan is liquidated, any of the liquidators in the Bitcoin vault can claim the BTC. Liquidators will perform liquidations if it is economically feasible for them.

Current Bitcoin vault implementations face significant limitations that prevent efficient capital utilization and risk management. Traditional Bitcoin vaults suffer from two primary constraints: liquidations cannot be triggered until the loan repayment deadline expires, and liquidations are not atomic, creating inefficiencies and risks for both lenders and borrowers.

This paper introduces the Bitcoin Vault Liquidation Engine, a solution that leverages smart contracts on a DeFi chain, existing bridges, and BitVM to enable trust-minimized, efficient liquidation mechanisms for Bitcoin lending vaults. Our system addresses the fundamental challenges of Bitcoin-native lending while maintaining the security guarantees that make Bitcoin the most trusted blockchain network.

1.1 Key Contributions

We are introducing the BOB Bitcoin Vault Liquidation Engine to offset the shortcomings of Bitcoin vaults with static liquidator sets. The key contributions are:

- **Open liquidator set**: We lift the restriction to have predefined liquidators to allow anyone to participate in liquidations. By allowing anyone to join liquidations, lending protocols do not need to make additional trust assumptions about the availability and liquidity of a predefined set of parties. It also allows for offsetting liquidity crunch scenarios back to the base assumption of lending protocols: as long as liquidations are sufficiently profitable, there should be parties executing the liquidations.

- **Multi-party liquidations**: Instead of a single liquidator having to liquidate the entire amount of the loan, the liquidation engine allows for multiple liquidators to act together.
- **Partial liquidations extension**: Instead of liquidating entire positions, the liquidation engine offers to process liquidations to only liquidate funds until a safe LTV ratio is reached.
- **Fast liquidations extension**: We can reduce the challenge period from a few days to a few Bitcoin block confirmations, e.g., 1 to 6, depending on the required confirmations settings of the lending protocol or its curators.
- **Atomic liquidations extension**: Liquidations can be atomic. This will allow liquidators to claim underlying collateral directly, in a single transaction, when repaying a loan as part of the liquidation process.

2 Background

To understand the liquidation engine, we first introduce the basic Bitcoin vaults construction and will then add the liquidation engine and its extensions. In Figure 1, we show a basic Bitcoin vault with a static liquidator, as used in the Babylon vault mainnet experiment. The Bitcoin vault is a BitVM implementation that, simplified, has a BTC deposit as an input and n outputs.

Output 0 for the depositor. If the depositor pays back their loan, the depositor can claim that they did so correctly and submit a transaction to Bitcoin to spend output 0. In typical BitVM logic, this can be challenged by the liquidators and require submission of a proof from the depositor. However, assuming that the claim of the depositor is provably correct, the challengers would be slashed for a false challenge and the depositor still receive their BTC.

Outputs 1..n for the liquidators. In the event of a liquidation, only one of the liquidators would repay the loan. Then, the liquidator would claim this to the Bitcoin vault via generating a proof of the onchain liquidation. Similar to the process of the depositor, the claim can be challenged, in this case by other liquidators and the depositor. After the challenge period, typically a few days, the liquidator receives the BTC. Note here that we need the liquidator to pay back the full loan amount (no partial liquidations) and one of the liquidators needs to provide the entire loan capital.

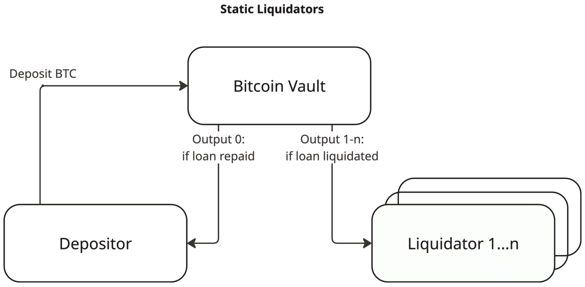

Figure 1: Basic Bitcoin vault with static liquidator

2.1 BitVM Overview

BitVM enables arbitrary computation verification on Bitcoin through optimistic challenge-response protocols. The system allows off-chain computation with on-chain dispute resolution, maintaining Bitcoin's security properties while enabling complex smart contracts. The BitVM bridge design[7] extends this concept specifically for cross-chain asset transfers.

Key properties of BitVM relevant to our liquidation engine:

- **Fraud Proofs**: Any dishonest behavior can be proven on-chain on Bitcoin.
- **Optimistic Execution**: Operations proceed unless challenged.
- **Capital Efficiency**: Minimal on-chain footprint for honest execution especially with improvements around BitVM3 [2, 4, 6, 8].
- **Trust Minimization**: Security relies on at least one honest participant.

2.2 Current Limitations of Bitcoin Vaults

The Bitcoin vault construction with pre-defined liquidators imposes three limitations:

(1) **Delayed reimbursement of liquidators**: When a liquidation occurs, the liquidators need to front the loan value out of pocket. Withdrawing BTC from the vault will take a few days, even in the optimistic case, to allow for a sufficiently long challenge window on the Bitcoin vault. Delayed reimbursements negatively impact the profitability of liquidations (opportunity cost) and increase risk (what if the value of BTC continues to fall against the USD loan value?).

(2) **Static liquidator sets**: By limiting the set of liquidators ahead of time, lending protocols would need to accept a ceiling on available capital during liquidation, introducing additional risk by assuming these liquidators will stay online, capitalized, and execute liquidations. In addition, market downturns often result in large amounts being liquidated within a short period. Exchanging large quantities of funds can lead to haircuts during market turbulence and liquidity shortages (see [3] for more formal economic work that we have been conducting in the past). Thus, the static liquidator's limitations are exacerbated by typical market situations.

(3) **No partial liquidations**: If a depositor uses a single BTC vault for their lending position, liquidations are all-or-nothing. That means, partial liquidations are not possible, and total loss of BTC occurs. Most lending protocols now execute partial liquidations, limiting the loss of collateral to returning the loan to a safe loan-to-value (LTV) ratio.

The success of a lending protocol, as well as CDP-style stablecoins, hinges on the ability to handle large-scale liquidations quickly and effectively during turbulent market conditions. The base design with static liquidators is insufficient for both the risk appetite of lending protocols and their curators, as well as the liquidators.

3 BOB Bitcoin Vault Liquidation Engine

Intuitively, the liquidation engine is a drop-in replacement for the liquidator's address on Bitcoin with a special Bitcoin address: a deposit address for a Bitcoin bridge. In the event of a liquidation, the BTC is sent to the bridge. Instead of having multiple liquidators, we replace this with one or multiple bridge deposit addresses as shown in Figure 2. The deposit to the bridge is triggered by operators who facilitate the setup of the BitVM instance and receive a cut of the liquidation profits to align their economic interests. We note that bridge providers seeking to upgrade their bridges to BitVM or BitVM bridges may reuse their current operator set.

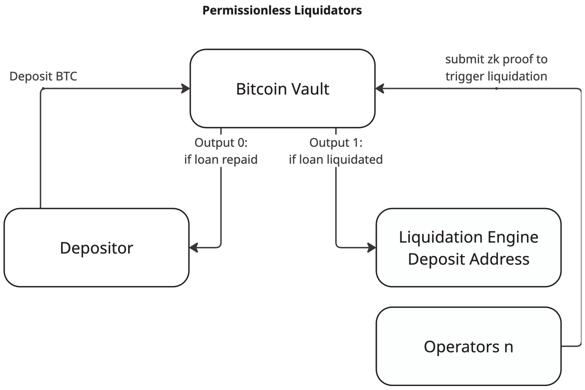
Permissionless Liquidators
Deposit BTC
Bitcoin Vault
submit zk proof to trigger liquidation
Output 0: if loan repaid
Output 1: if loan liquidated
Depositor
Liquidation Engine Deposit Address
Operators n

Figure 2: Bitcoin Vault Liquidation Engine High-Level Architecture

For the depositor, as long as they repay their loan, they do not have to trust this bridge. In the base version of the liquidation engine, their trust assumptions remain the same: they can withdraw their BTC from the Bitcoin vault at any time, given they have repaid their loan.

We will introduce the liquidation engine in four parts:

(1) **Base Liquidation Engine**: Enabling both an open liquidator set and allowing multiple liquidators to work together. We can achieve these two properties by keeping the trust assumptions for the depositor and not requiring any modifications to the infrastructure that the liquidation engine needs to interact with.
(2) **Partial Liquidation Extension**: It is possible to execute partial liquidations if depositors are willing to trust the bridge used for the liquidation engine. On a high level, if depositors accept bridged BTC on the chain of the lending protocol, then liquidations can be partial. The partial liquidation extension could even be applied on a depositor-by-depositor preference, i.e., depositors can opt-in to this feature.
(3) **Fast Liquidation Extension**: If the depositor is willing to trust 1 of n operators for safety and n of n operator for liveness, we can directly transfer the BTC to the bridge by requiring all operators to spend the BTC to the bridge without having to go through the BitVM assertion and possibly challenge process. The BitVM fraud-proofing logic serves as a fallback if at least one of the operators is not willing or available to sign the fast withdrawal. This feature is also on an opt-in basis for the depositor.
(4) **Atomic Liquidation Extension**: If a Bitcoin bridge accepts Bitcoin vaults with outputs that transfer BTC to its bridge as pre-deposits, then liquidations can occur atomically.

4 Base Liquidation Engine: Open Liquidator Set

The base liquidation engine fundamentally transforms Bitcoin vault liquidations by replacing static, predefined liquidator sets with an open system that allows anyone to participate. By routing liquidations through a Bitcoin bridge deposit address and coordinating them via a smart contract, as indicated in Figure 3, the base engine enables multiple liquidators to collaborate on single liquidations while ensuring continuous liquidity availability during market stress. Crucially, this design maintains the trust-minimized properties for depositors, who can still withdraw their BTC directly if they repay their loans, requires no modifications to existing bridge infrastructure, and preserves the core security guarantees of Bitcoin vaults.

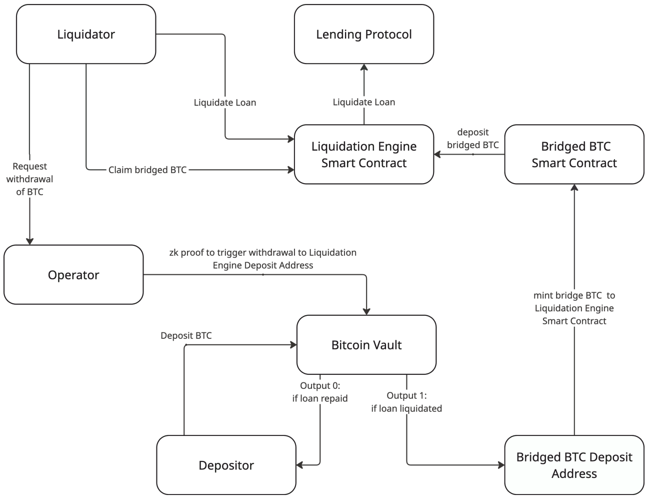
Liquidator
Lending Protocol
Liquidate Loan
Liquidate Loan
deposit bridged BTC
Liquidation Engine Smart Contract
Bridged BTC Smart Contract
Request withdrawal of BTC
Claim bridged BTC
Operator
zk proof to trigger withdrawal to Liquidation Engine Deposit Address
mint bridge BTC to Liquidation Engine Smart Contract
Deposit BTC
Bitcoin Vault
Output 0: if loan repaid
Output 1: if loan liquidated
Depositor
Bridged BTC Deposit Address

Figure 3: Liquidation Engine with Smart Contracts on DeFi Chain included.

The base liquidation engine consists of the following parts:

(1) **Outputs on the Bitcoin vault into a BTC bridge deposit address.** In the figure below, we use bridged BTC. We require that the BTC bridge can provide a BTC address, such that any BTC deposited is minted as wrapped BTC via a smart contract on a pre-defined destination chain. In the base version of the liquidation engine, we can use any bridge that supports such functionality, and we do not require the bridge to modify its code or operations.
(2) **Liquidation Engine smart contract.** This smart contract tracks the liquidations, i.e., how much each liquidator has repaid of the loan, and is responsible for reimbursing the liquidators. In the simple version, the liquidation engine smart contract is deployed on the same chain as the lending protocol. However, it would also be possible for the liquidation engine smart contract to be on another chain than the lending protocol. For example, the liquidation engine can be deployed on a fast and cheap chain like BOB and serve lending protocols on Ethereum, Base, BOB, and other chains at the same time.

(3) **Operators for the Bitcoin vault.** Operators facilitate BTC withdrawals from the Bitcoin vault to the BTC bridge during liquidations.

4.1 Lending Flows

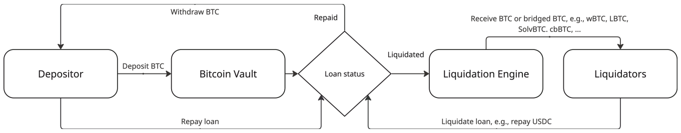

Figure 4: Liquidation process flow in the base liquidation engine

The processes depend on the state of the loan as indicated in Figure 4. If the depositor repays the loan, the depositor can withdraw directly from the Bitcoin vault without any reliance on the liquidation engine components.

If the loan is below the safe LTV, then it can be liquidated. Now, the liquidation engine comes into play. In the base liquidation engine, we require the following steps to happen:

(1) **Vault setup.** When a depositor seeks to lock BTC in a Bitcoin vault, a set of operators collaborates with the depositor to create the vault with at least two outputs: one to the depositor and one to a BTC bridge deposit address. The BTC deposit address needs to be requested such that the bridge provider issues the BTC to the liquidation engine smart contract in case the BTC is sent to the BTC deposit address.
(2) **Liquidator liquidates the loan.** A liquidator repays (part of) the loan via the liquidation engine smart contract. The smart contract creates a mapping of the amount provided by the liquidator and the value of collateral owed to the liquidator. Multiple liquidators can partake in a liquidation of a loan. The liquidation engine simply tracks the amounts contributed and the owed collateral. Contributions can be made until the loan is fully repaid.
(3) **Operators trigger the liquidation on the Bitcoin vault.** One of the operators asserts that the loan was liquidated and submits this to the vault on Bitcoin. The depositor and the other operators can challenge this (BitVM fraud proof) if this claim is incorrect. Assuming the claim is correct, the BTC locked in the vault is deposited into the BTC bridge.
(4) **BTC minted to the liquidation engine smart contract.** The BTC deposited into the bridge is minted to the liquidation engine smart contract when the challenge period of the Bitcoin vault has passed without a successful challenge, and the BTC deposit has sufficient confirmations for the BTC bridge.
(5) **Liquidators claim bridged BTC.** The liquidators claim the bridged BTC from the liquidation engine smart contract. Note: If the bridged BTC is already in the liquidation smart contract but parts of the loan still need to be liquidated, the liquidations are atomic from this point onwards. This works because the entire BTC is bridged within one deposit process.
(6) **Operators claim bridged BTC.** The operators receive a fee from the liquidated and bridged BTC.

4.2 Assumptions

There are assumptions for the liquidation engine to work correctly: We assume that the bridged BTC version used is acceptable to the liquidators. This would typically entail that the bridged BTC is liquid, i.e., they can sell the bridged BTC for the funds they used to repay the loan without significant slippage. Also, the counterparty and technical risks of the bridged BTC are acceptable.

We also assume that the price of the bridged BTC asset stays roughly the same during the liquidation period. If liquidators have reason to believe that the price of the bridged BTC asset might be subject to large price fluctuations compared to BTC, they might not accept it.

We also assume that the BTC locked in Bitcoin vaults is not higher than the possible minting limits or asset ceilings for the BTC bridged, i.e., if BTC is deposited into the bridge, it is actually minted and not returned. We assume that the deposit addresses for the BTC bridge remain valid over the lifetime of the vault. Individual vaults might be around for multiple years, and deposit addresses need to remain valid. Especially for BitVM-style bridges, this poses a challenge as operators change over time and with it deposit addresses. Additionally, instances need to be pre-created, raising concerns around multiple liquidation paths and timeslot assumptions.

5 Partial Liquidation Extension

The partial liquidation extension addresses one of the most significant limitations of basic Bitcoin vaults and the base liquidation engine: the all-or-nothing liquidation mechanism. In traditional lending protocols, partial liquidations are essential for maintaining capital efficiency and reducing unnecessary losses for borrowers. This extension enables liquidations to be executed only to the extent necessary to return the loan to a safe loan-to-value (LTV) ratio, rather than liquidating the entire collateral position.

The partial liquidation extension operates through the following mechanism:

(1) **BTC Bridge Deposit:** When a liquidation is triggered, the entire BTC amount from the vault is still sent to the bridge deposit address.
(2) **Partial Liquidation:** The liquidation engine smart contract uses the lending protocol to determine the safe LTV ratio and the amount that can be at most liquidated.
(3) **Bridged BTC Distribution**: Once the bridged BTC is minted to the liquidation engine smart contract, liquidators claim only the portion of bridged BTC corresponding to their loan repayment, plus the liquidation bonus. The remaining bridged BTC (representing the non-liquidated portion of collateral) is reserved for the depositor. Operators receive their fee from the liquidated portion.
(4) **Depositor Reclamation Options:** The depositor has two options for the remaining collateral. Claim Bridged BTC: The depositor can claim the bridged BTC directly on the smart contract chain and use it within that ecosystem. Bridge Back to Bitcoin: The depositor can initiate a withdrawal through the same bridge to receive native BTC back, though this incurs additional bridging fees and time

If the depositor does not claim their collateral immediately and the loan is liquidated again, any subsequent liquidation is atomic, since the BTC are already bridged.

5.1 Assumptions

The partial liquidation extension introduces a necessary trust trade-off for depositors:

- **Additional Trust Required:** Depositors must trust the bridge provider with the portion of their collateral returned to them after partial liquidation. This deviates from the trust-minimized nature of the base Bitcoin vault.
- **Reduced Trust for BitVM Bridges:** If the underlying BTC bridge is a BitVM bridge, very large depositors or trusted providers (such as custodians or infrastructure providers) may be part of the BitVM operator set. In this case, the trust assumptions would be reduced, though they would still need to trust the correct implementation of the bridge.
- **Opt-in Mechanism**: This extension can be implemented as an opt-in feature at the vault creation stage. Depositors can choose between standard vaults with complete liquidation but no bridge trust requirements or enhanced vaults with partial liquidation capabilities, but requiring trust in a chosen bridge. We imagine that liquidators would be able to chose among multiple bridges.
- **Economic Considerations**: Most depositors are likely to accept this trade-off because the risk of losing unnecessary collateral in complete liquidations often outweighs the bridge trust requirement. Many users already interact with bridged BTC in DeFi applications and the bridge is only trusted for the non-liquidated portion, not the entire collateral.

6 Fast Liquidation Extension

The fast liquidation extension substantially reduces the settlement time for liquidations from several days to 1 to 6 Bitcoin block confirmations (approximately 10-60 minutes). This improvement is crucial for liquidators who need to manage capital efficiency and reduce market risk during volatile periods.

The fast liquidation extension leverages a modified operator set structure:

1. **Multi-Operator Setup:** Multiple operators are designated during vault creation to ensure both liveness for liquidators and safety. Only operators in the set can perform fraud proofs, creating a balanced security model. These operators collectively control the fast-track liquidation path.
2. **Flexible Consensus Models**: The system can support various consensus requirements that allows protocols that already use a multisig setup to integrate seamlessly.
    - *n-of-n consensus*: All operators must agree for fast liquidation (strongest security, same as 1 of n security assumption as BitVM bridges)
    - *m-of-n consensus*: Simple majority or supermajority (e.g., 2/3) of operators must agree (balanced security and liveness similar to wBTC, cbBTC, LBTC, ...)
    - *m-of-n consensus with spending restrictions (emulated covenants)*: Essentially, we apply the same m-of-n consensus, but the Bitcoin vault is set up so that the only spending paths are back to the depositor or into the bridge. That way, even a malicious majority of m operators cannot steal the BTC from the vault.
3. **Cooperative Fast Path**: When the required threshold of operators agrees (based on the chosen consensus model), they can immediately authorize the transfer of BTC to the bridge deposit address without going through the BitVM assertion and challenge process. This works because if sufficient operators are honest and agree, there's no dispute about the liquidation's validity. The agreement serves as implicit verification of the liquidation conditions.
4. **BitVM Fallback**: If the required consensus threshold is not met (e.g., majority disagrees or operators are unavailable), the system automatically falls back to the standard BitVM fraud-proof mechanism. This ensures that no subset of operators below the threshold can block legitimate liquidations, the system remains resilient to operator failures or attacks, and trust assumptions degrade gracefully to 1-of-n security in the fallback case.

6.1 Process

The fast liquidation process follows these steps:

1. **Liquidation Trigger**: When a loan falls below the safe LTV, liquidators initiate the liquidation through the liquidation engine smart contract.
2. **Operator Notification**: All operators are immediately notified of the liquidation request.
3. **Signature Collection**: Fast Path (Required threshold signs within timeout period): The BTC is immediately sent to the bridge deposit address. Fallback Path (Threshold not met): The standard BitVM assertion process begins, adding the usual challenge period.
4. **Settlement**: In the fast path, liquidators receive their bridged BTC within 1-6 Bitcoin confirmations (10-60 minutes), plus bridge processing time (varies by bridge type).

6.2 Assumptions

The fast liquidation extension modifies the trust assumptions:

- **Depositor**: In the n-of-n model: Must trust that at least 1 of n operators remains honest (1-of-n security). In the m-of-n model: Must trust that more than m of n operators remain honest (weaker than BitVM's 1-of-n but still robust). If the collusion threshold is reached, malicious operators could potentially steal BTC without proper liquidation unless (emulated) covenants are used.
- **Lending Protocol**: The lending protocol bears the primary risk during liquidations, as if they are not executed in a timely manner, the protocol might face bad debt. The lending protocol can decide whether to make this feature mandatory or optional for depositors, who in turn will have to accept the risk trade-offs. With this mechanism, it would also be possible to make fast withdrawals for depositors; i.e., the benefit could be applied in the happy case where depositors repay loans. Instead of waiting for days to receive their BTC, operators could also fast withdraw the BTC from the depositor with the selected trust model (n-of-m or n-of-n)

- **Liquidators**: Significantly reduced exchange rate risk due to faster settlement. This could also lead to lower capital requirements as funds are locked for shorter periods. However, liquidators assume additional counterparty risk in the operator set if an m-of-n model is applied. If n-of-n, the trust remains the same, since if no operator processes the withdrawal, the liquidators also do not get the BTC.

7 Atomic Liquidation Extension

The atomic liquidation extension is the most advanced optimization, enabling liquidations within a single blockchain transaction. This eliminates all settlement risk for liquidators and creates a seamless liquidation experience on par with current DeFi protocols. It enables flash loans for liquidations.

The atomic liquidation system requires deep integration between the Bitcoin bridge and the liquidation engine:

(1) **Pre-Deposit Recognition**: The bridge protocol recognizes Bitcoin vaults with outputs directed to their deposit addresses as valid pre-deposits, even before the BTC is actually transferred.
(2) **Pre-Mint Authorization**: Based on the vault structure, the bridge issues a cryptographically signed pre-mint authorization to the liquidation engine smart contract. This authorization includes: the vault identifier, the maximum mintable amount (the BTC locked in the vault), the expiration conditions tied to liquidation events, the revocation conditions if the loan is repaid, and signature from bridge operators confirming the pre-authorization.
(3) **Atomic Execution**: When a liquidation is triggered, the entire process occurs in one transaction: The liquidator repays the loan, the smart contract verifies the pre-mint authorization, the bridged BTC is instantly minted to the liquidator, the loan is marked as liquidated, and an event is emitted to notify bridge operators.
(4) **Asynchronous Settlement**: After the atomic liquidation, bridge operators begin the Bitcoin vault withdrawal process. BTC moves from the vault to the bridge's reserves with no additional minting as it was already pre-authorized. The bridge reconciles its books to reflect the completed liquidation.

If bridge providers seek to upgrade their bridge to allow BitVM-style Bitcoin vault deposits, then this extension is interesting in isolation. Upon depositing to a Bitcoin vault, the bridge would issue the bridged BTC to the depositor for use in DeFi. The depositor can get exactly their BTC back and enforce the withdrawal. If the depositor should be liquidated, the bridge adds the BTC from the vault to its inventory.

7.1 Requirements

For atomic liquidations to function properly, several conditions must be met. The bridge must support pre-deposit recognition for Bitcoin vaults. These pre-deposits need to be verifiable by the liquidation engine and integrated at the smart contract level. Within the liquidation flow, the actual minting of the bridged BTC must occur in the same transaction that repays the loan.

Bitcoin vaults must follow a standardized format that bridges can recognize and validate, including deterministic output scripts to bridge addresses, liquidation condition encoding, and standardized challenge periods.

The bridge must maintain sufficient liquidity to handle pre-mints. There is a time delay until the BTC is fully liquid within the context of the bridge provider (due to the nature of BitVM). In a bank-run scenario on the bridge, the bridge needs sufficient liquidity reserves to meet ongoing requests.

7.2 Assumptions

The atomic liquidation extension trust assumptions are preserved for the depositor, but change for other parties compared to the base liquidation engine:

- **Depositor**: The depositor can withdraw funds as with the base liquidation engine and as well as with the basic Bitcoin vault.
- **Lending Protocol**: The lending protocol improves trust assumptions by introducing counterparty trust in the bridge, but with atomic liquidations enabled, it dramatically reduces the likelihood of bad debt.
- **Liquidators**: Significantly reduced exchange rate risk due to faster settlement. This could also lead to lower capital requirements as funds are locked for shorter periods.

8 Considerations

The three extensions can be combined for maximum effectiveness:

(1) **Partial + Fast**: Enables quick partial liquidations, ideal for volatile markets where speed and capital efficiency are crucial. Logic is implemented entirely in the liquidation engine and BitVM logic.
(2) **Partial + Atomic**: Provides partial instant liquidation. Requires collaboration from bridge providers.
(3) **All Three**: Partial liquidations by default. Atomic liquidations if sufficient liquidity in the bridge. Fast fallback in case not enough liquidity on the bridge protocol. Slow fallback to base BitVM.

In Table 1, we provide an overview of lending using a generic, fully fungible BitVM bridge such as designed in [7], Bitcoin vaults with static liquidators [4], and Bitcoin vaults with the liquidation engine.

9 Conclusion

The BOB Bitcoin Vault Liquidation Engine, with its extensions, represents a significant advancement in native Bitcoin DeFi participation. By addressing the key limitations of static liquidator sets, all-or-nothing liquidations, and slow settlement times, this system enables:

- **Trust-minimized lending:** Users maintain custody of their BTC while participating in DeFi
- **Capital efficiency:** Partial liquidations and fast settlement reduce unnecessary losses
- **Market resilience:** Open liquidator sets ensure sufficient liquidation capacity during stress
- Seamless experience: Atomic liquidations provide UX comparable to traditional DeFi

Desired Action	Lending using generic BitVM bridge	Lending using a Bitcoin vault	Lending using a Bitcoin vault with liquidation engine
Alice (lender) and Bob (borrower) create a lending contract	Trusts n-of-n signer committee and m-of-m operators	Trust minimized*	Trust minimized*
Bob (depositor) can withdraw collateral	1-of-n signer committee 1-of-m operators 1 challenger. Bob might receive different BTC than originally deposited.	Trust minimized*	Trust minimized*. Trust BTC bridge (centralized, threshold, BitVM, . . .) if opting into partial liquidations. Trust 1 of n operators if opting into the fast liquidations.
Bob (depositor) is only liquidated until safe LTV	Yes	No, full collateral is liquidated	Yes, if the partial liquidation extension is used.
Alice (lender) can liquidate collateral	1-of-n signer committee 1-of-m operators 1 challenger	Trust minimized*	Trust BTC bridge (centralized, threshold, BitVM, . . .) if opting into partial liquidations
Alice (lender) can liquidate collateral quickly	Within a single transaction if sufficient liquidity of BTC BitVM asset. Multiple days if BTC needs to be withdrawn from BitVM instance.	Multiple days	Multiple days in base liquidation engine. Few minutes to hours with fast liquidations extension. Within a single transaction with atomic liquidation extension and sufficient liquidity of the bridged BTC asset.
Lending protocol has no bad debt	No specific trust on BitVM parties. Anyone can execute liquidations. Liquidations need to be profitable.	Of the set of pre-defined set m of liquidators, sufficient liquidators participate in the liquidations. Liquidations need to be profitable.	Anyone can execute liquidations. Liquidations need to be profitable.

Table 1: Comparison of different BitVM lending approaches. Trust minimized in this context means: trust Bitcoin and DeFi chain consensus, trust or verify that the lending protocol implementation is correct, and trust or verify that the Bitcoin vault implementation is correct.

The modular nature of the extensions allows protocols to choose the right balance of features, trust assumptions, and complexity for their specific use cases. As the Bitcoin DeFi ecosystem matures, these liquidation mechanisms will be crucial for unlocking the full potential of native Bitcoin as collateral across multiple chains.

The liquidation engine also works for non-BitVM Bitcoin vaults, e.g., multisig-style deposit vaults, albeit the withdrawal times are less concerning and depositors cannot enforce withdrawals since they are at the mercy of the multisig committee.

The liquidation engine transforms Bitcoin vaults from an interesting technical concept into production-ready infrastructure for lending protocols. By addressing the fundamental liquidation challenges, we can finally bridge the gap between Bitcoin's security and the innovation in DeFi, enabling billions of dollars in dormant Bitcoin to participate in decentralized finance without sacrificing self-custody.

References

[1] BTC Relay Contributors. 2016. BTC Relay: A Bitcoin-to-Ethereum bridge. https://github.com/ethereum/btcrelay. Accessed: 2025-10-27.

[2] Liam Eagan. 2025. Glock: Garbled Locks for Bitcoin. Cryptology ePrint Archive, Paper 2025/1485. https://eprint.iacr.org/2025/1485.pdf

[3] Lewis Gudgeon, Daniel Perez, Dominik Harz, Benjamin Livshits, and Arthur Gervais. 2020. The Decentralized Financial Crisis. In *2020 crypto valley conference on blockchain technology (CVCBT)*. IEEE, 1–15.

[4] Babylon Labs. 2025. Trustless Bitcoin Vaults. *URL: https://docs.babylonlabs.io/papers/trustless-bitcoin-vaults.pdf* (2025).

[5] Robin Linus. 2023. BitVM: Compute Anything on Bitcoin. *URL: https://bitvm.org/bitvm.pdf* (2023).

[6] Robin Linus et al. 2025. BitVM3: Efficient Dispute Resolution for Advanced Bitcoin Smart Contracts. *URL: https://bitvm.org/bitvm3.pdf* (2025).

[7] Robin Linus, Lukas Aumayr, Alexei Zamyatin, Andrea Pelosi, Zeta Avarikioti, and Matteo Maffei. 2024. BitVM2: Bridging Bitcoin to Second Layers. *URL: https://bitvm.org/bitvm_bridge.pdf* (2024).

[8] Citrea Team. 2024. Clementine: A Trust-Minimized Two-Way Bitcoin Bridge. *URL: https://citrea.xyz/clementine_whitepaper.pdf* (2024).

[9] David Tse. 2025. Bitcoin Vaults on Morpho. https://x.com/dntse/status/1978625249660060153. Twitter/X post.