Crossvault protocol: a decentralized BTC liquidity layer

Vinh The Nguyen (vinh.nt@nerd-labs.io)

I. The best solution to omni-chain liquidity shortage is BTC

The cryptocurrency space is grappling with an intensifying omni-chain liquidity shortage, particularly affecting emerging ecosystems. As evidenced in Figure 2, most established ecosystems experienced weak Total Value Locked (TVL) growth from 2023 to 2024. In stark contrast, Ethereum's TVL tripled and Bitcoin reached a new all-time high during this period. This weak TVL growth, particularly in DeFi - focused ecosystems, is undermining investor confidence and user adoption. To combat this trend, ecosystems are resorting to unsustainable practices, competing on high yields and generous token airdrops to attract liquidity. This situation disproportionately impacts growing ecosystems, forcing them to allocate substantial resources to user acquisition campaigns at the expense of critical ecosystem development and innovation.

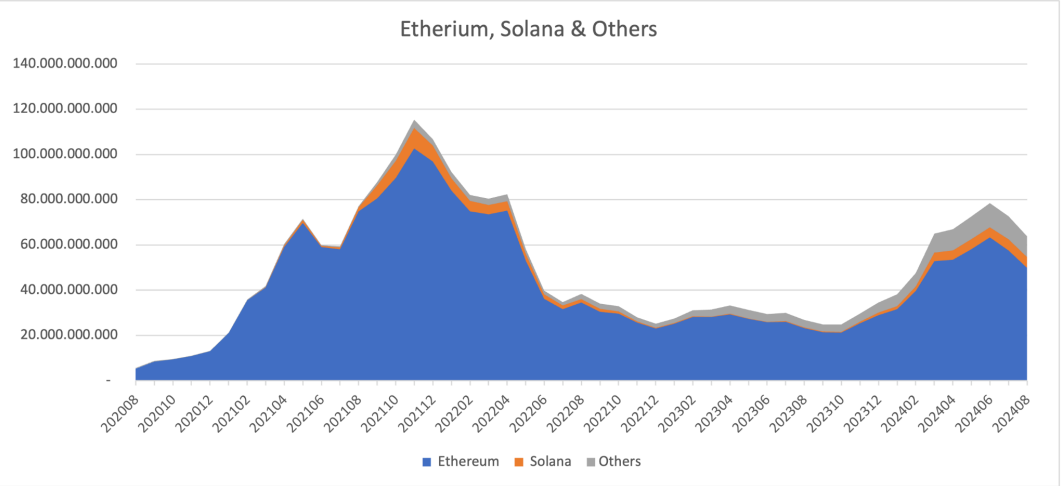

Fig 1. Ethereum total TVL from 2020 - 2024 (from DefiLlama)

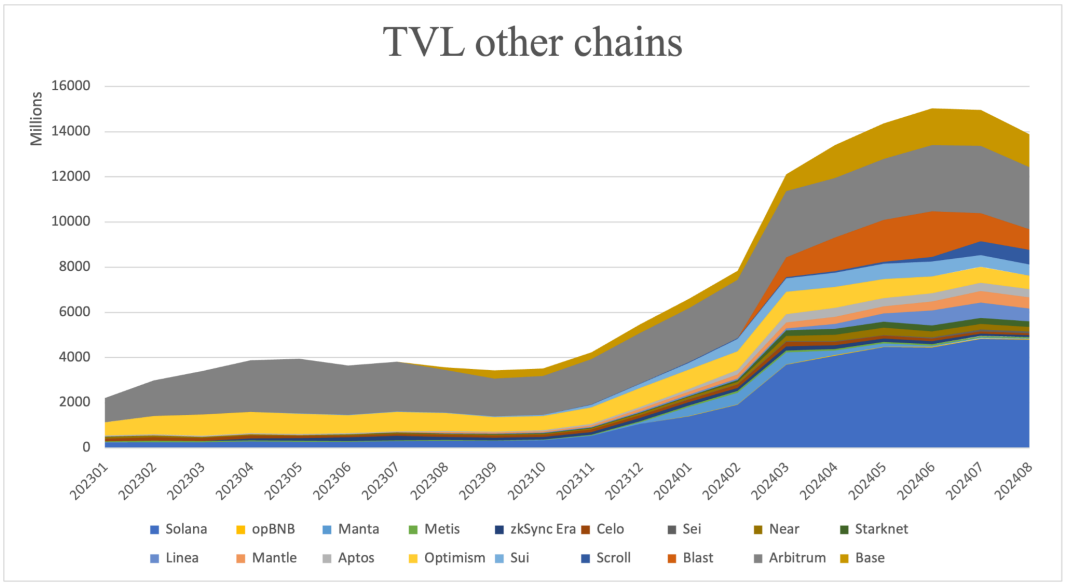

Fig 2. Other chains total TVL from 2023 - 2024 (from DefiLlama)

Meanwhile, a staggering 58% of the total cryptocurrency market capitalization, representing over $1 trillion in value, sits dormant on the Bitcoin network, not actively participating in any major DeFi dApps. This untapped potential stems from the historical lack of secure, scalable infrastructure for omnichain BTC transportation. However, the recent Taproot [1] upgrade on the Bitcoin network makes it possible to unlock this vast liquidity pool. Should BTC be effectively integrated into other ecosystems, it would more than double the total available liquidity. This massive influx of capital could catalyze unprecedented growth in TVL across multiple ecosystems, potentially growing the entire DeFi landscape.

II. Shortcomings of existing solutions

Existing protocols aim to attract and transport BTC liquidity to other ecosystems, but face significant challenges in balancing two critical, often conflicting properties: distributed security and liquidity user experience (UX). The inherent tension between these factors creates a complex optimization problem that current solutions struggle to solve effectively. Given Bitcoin's core ethos of decentralization and its community's strong anti-institutional stance, distributed security must be prioritized over UX. Current protocols often fall into one of two inadequate categories: they are either overly centralized, or they offer such poor liquidity UX that they fail to attract significant adoption. This dichotomy highlights a critical gap in the market: no existing solution has successfully navigated the trade-offs to provide both robust distributed security and seamless liquidity UX at scale.

1. **Distributed security assessment**

   A robust security protocol for cross-chain BTC transportation must prevent fund loss and avoid freezing under adverse conditions. However, current solutions, primarily implementing either global or self-custodial vault architectures, fall short of this ideal. When having a non-scalable set of authorities, global vault designs expose users to risks of fund loss and freezing. Self-custodial designs, while providing stronger safeguards against fund loss, still fail to fully mitigate the risk of asset freezing when faced with the same authority scalability constraints.

   - Global bridge vaults are implemented as on-chain multisig addresses entirely controlled by a set of designated authorities. This design requires users to deposit their BTC, after which an equivalent amount is minted on a destination chain. With limited scalability of the trusted authority set, users face significant risks of fund loss or freezing if even a single authority is compromised.

     Notable examples like WBTC and LBTC currently employ an m-of-n multisig scheme with trusted parties. Following the Taproot [1] upgrade, this scheme is best implemented as an aggregate Tapscript [1] of C(n, m) branches, containing MuSig [3] (an m-of-m signing scheme). Tapscript verification requires revealing only a single 66-byte MuSig branch, resulting in minimal gas costs. However, increasing the number of authorities leads to an exponential growth in branches. Each time an inactive or dishonest signer needs replacement, a new

Tapscript must be recreated. For instance, a 70-of-100 multisig would require an impractical number of branches, reaching 10^30. This limitation effectively restricts the number of trusted parties to less than 10, making the protocol vulnerable to majority attacks by just a few compromised authorities. As the value of deposits increases, so do the incentives for attacks, while the cost to execute such attacks remains the same.

Nomic offers an alternative implementation using an m-of-n weighted multisig script [4], periodically checkpointing bridge orders to Bitcoin. This script incorporates all authority public keys along with their respective voting power. During evaluation, it verifies authority signatures, sums the voting power, and unlocks if a supermajority is reached. While the script size scales linearly with the number of authorities, this approach, despite its reduced scaling complexity, incurs significantly higher gas costs when revealed on chain. For instance, an on-chain script size of 2.5 kB for 20 authorities [15] can result in gas fees of approximately $10,000 per day when checkpointing every block. Attempts to reduce these costs by decreasing the checkpointing frequency result in slower bridge order resolution, potentially extending user waiting time from minutes to hours. These prohibitive gas expenses severely constrain the scalability of the authority set.

- Self-custodial vaults represent an alternative approach, implemented as on-chain contracts jointly managed by individual users and an authority set. This mechanism involves initial contract setup, followed by locking user deposits and minting equivalent amounts on the destination chain. These on-chain contracts, enforceable by Bitcoin's Proof-of-Work, can prevent fund loss even with a non-scalable set of trusted authorities although still vulnerable to freezing.

  One notable implementation utilizes Discrete Log Contracts (DLC) [5] for self-custodial vaults. The DLC.link [6] solution requires both parties to agree on pre-signed payment paths before user deposits. These paths are activated when the authority set broadcasts corresponding event signatures. While this approach allows users to verify payment paths for potential issues, it heavily relies on authority liveness. The inactivity of a single authority can prevent the derivation of aggregated event signatures, resulting in frozen funds. This vulnerability intensifies as the number of authorities increases, making the protocol more susceptible to liveness issues.

  Babylon's approach [11], while limited to only remote security rather than liquidity provision, employs Extractable-One-Time-Signatures (EOTS) [7] to create trustless self-custodial vaults. This method requires stakers to deposit BTC as remote proof-of-stake to secure other ecosystems. In cases of double-signing, anyone can extract the vault's private key from two mismatched signatures for the same state,

enabling them to burn all BTC in the staker's self-custodial vault. As long as stakers remain honest, their vault private keys stay hidden, allowing them to capture yields in other ecosystem's native tokens. Since the self-custodial vault remains secure as long as stakers act honestly, the authority set lacks the ability to freeze the deposits, ensuring complete vault control for stakers.

2. **Liquidity UX assessment**

   While many protocols prioritize decentralized security, they often overlook the equally crucial aspect of liquidity user experience (UX). Users have little incentive to bridge their BTC if faced with long waiting times or cumbersome processes. Moreover, the yields in unfamiliar tokens further diminishes the appeal, as users must then navigate complex exchanges to convert back to BTC or stablecoins.

   - Global bridge vaults excel in liquidity flexibility by centralizing all bridged BTC, offering a range of significant advantages. It enables the implementation of batch processing for bridge orders, substantially reducing gas costs for users. The centralized pool facilitates seamless liquidity sharing among participants, improving overall market depth and efficiency. Perhaps most importantly, users gain the ability to flexibly rebalance their BTC across various strategies or ecosystems at any time, maximizing their opportunities and adapting quickly to market conditions. This combination of features in global vault designs maximizes capital efficiency and provides users with unparalleled optionality in managing their cross-chain BTC positions.
   - Self-custodial vaults, while offering enhanced security, present significant challenges in terms of liquidity user experience. By design, these vaults fragment liquidity, isolating each user's funds and preventing efficient sharing of resources. The redemption process requires burning an exact amount of bridged BTC, introducing complexity and potential inefficiencies. This rigid structure makes the protocol vulnerable to exploitation through social engineering tactics. For instance, malicious actors could hoard bridged BTC to later manipulate prices, selling at predatory rates. Even more concerning, users might trade their bridged BTC for stablecoins and permanently abandon their vaults. Such inflexibility severely restricts users' ability to rebalance their BTC holdings or participate across multiple ecosystems.

3. **Conclusion**

   In conclusion, current cross-chain BTC bridge solutions present a stark trade-off between distributed security and liquidity user experience (UX). Global vault designs offer better liquidity flexibility but at the cost of increased centralization risks. Conversely, self-custodial vaults provide enhanced security through individual control but suffer from fragmented liquidity. With limited scalability in authority set, neither approach fully satisfies the dual requirements of decentralized security and seamless liquidity UX that are crucial for widespread adoption.

These shortcomings, if left unaddressed, threaten to erode user confidence and impede the effort of bringing BTC to omnichain. The limitations in both designs underscore the pressing need for an innovative solution that doesn't compromise on either security or UX. A better approach would not only balance these competing priorities but potentially enhance both aspects simultaneously.

III. Proposed security solution: Schnorr Proof-of-Stake vault consensus

Global and self-custodial vaults each have distinct challenges. Global vaults primarily suffer from a non-scalable authority set, while self-custodial vaults, though designed for isolation, remain vulnerable to social engineering even with a scalable authority set. Our solution adopts the global vault approach, as we've developed a method to effectively scale its authority set, addressing its main weakness. The global vault design consists of two key components: a Bitcoin address managed by the authority set, and an off-chain account book detailing deposits.

To address the global vault's scalability issue, we need a threshold multi-signature scheme that can scale effectively while remaining succinct on-chain to keep costs low. This requirement presents two main challenges: coordinating a large number of authorities efficiently and overcoming Bitcoin's limited script expressiveness. Our solution employs Weighted Schnorr Threshold Signatures (WSTS) [8] and the CometBFT [12] consensus engine to overcome these hurdles.

Beyond vault safety, we must also guarantee the immutability and slashability of the off-chain account book for true decentralization. We achieve this using Babylon's EOTS [7], which allows these properties to be enforced by Bitcoin's Proof-of-Work.

1. **Weighted Schnorr Threshold Signatures (WSTS)**
   Weighted Schnorr Threshold Signatures (WSTS) [8] is an advanced multisignature scheme built upon Flexible Round-Optimized Schnorr Threshold Signatures (FROST) [9]. WSTS leverages Schnorr [2] signatures to achieve a minimal on-chain footprint, making it a highly efficient Bitcoin script.

   WSTS improves upon traditional multisig schemes like MuSig [3] with Tapscript [1] branches by offering superior scalability in terms of the number of keys it can handle. This scalability comes from FROST's inherent threshold property, which allows for a larger number of participants without significantly increasing script complexity.

   However, this increased scalability does come with a trade-off. FROST requires more off-chain coordination and computation, primarily due to its use of a Distributed Key Generation (DKG) [10] and Lagrange interpolation. The communication and computation overhead grows exponentially as the number of keys increases. This is where WSTS can do better than FROST by separating the number of authorities from the number of keys. This feature reduces overhead by keeping the communication and computation to the much smaller number of participants instead of keys. For instance, 1000 keys are proportionally managed by 100 authorities based on each stake shares.

This separation also introduces weighted signatures crucial for PoS cryptoeconomics, where participants with more at stake can be assigned more keys and thus more voting power.

These properties make WSTS particularly well-suited for building a Proof-of-Stake (PoS) consensus mechanism, as it allows for fair distribution of voting power based on stake while maintaining the security and efficiency benefits of Schnorr signatures.

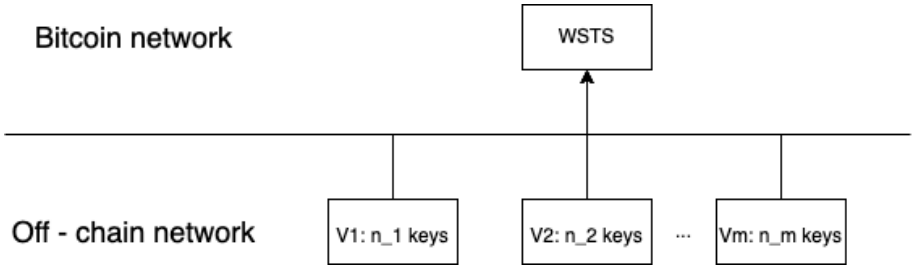

Fig 3. Composition of a WSTS scheme

2. **EOTS**

Extractable One-Time Signatures (EOTS) [7] is a cryptographic mechanism designed to prevent double-signing in consensus protocols. The key feature of EOTS is that if a participant signs two different messages (such as conflicting blockchain states) using the same private key, anyone can mathematically derive that private key from the two signatures.

This property creates a powerful deterrent against dishonest behavior. Participants' private keys are linked to their staked assets. If a participant double-signs, their private key can be extracted and used to transfer their staked assets to a burn address, effectively destroying them. This severe penalty discourages attempts to fork the blockchain or manipulate the consensus.

Babylon [11], a Bitcoin staking protocol, uses EOTS to secure remote blockchain networks using Bitcoin as collateral. In this system, consensus authorities can only attest to one version of the blockchain history. Any attempt to support conflicting versions would result in the loss of their Bitcoin stake. Importantly, this security holds even if a majority of participants collude in an attack. While they might temporarily disrupt the consensus, their Bitcoin stakes would still be at risk of being burned on the Bitcoin network by anyone.

This technology is crucial for maintaining the integrity of the "longest chain" in consensus protocols, as it provides a strong economic disincentive against forking attacks. In the context of the Bitcoin vault, EOTS can help ensure that the off-chain account books remain consistent and tamper-resistant, enhancing the overall security.

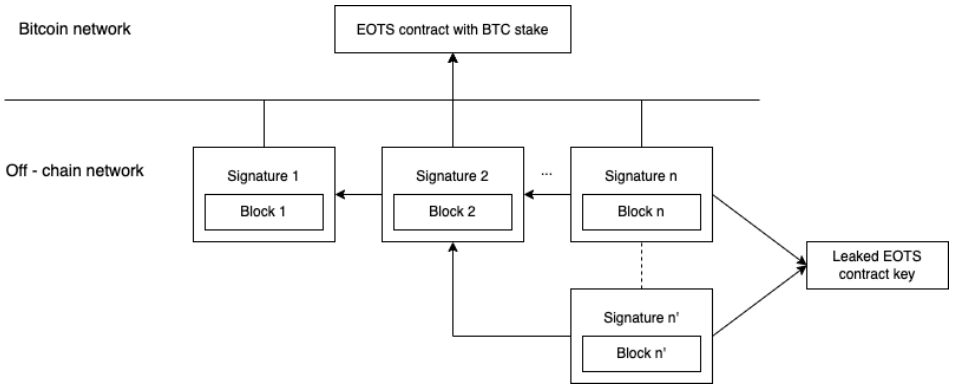

Fig 4. Mechanism of an EOTS contract

3. **Consensus engine CometBFT** (formerly Tendermint [12])
   As the number of authorities in WSTS [8] signing increases, coordinating them becomes exponentially more complex. Rather than building a coordination engine from scratch, we leverage CometBFT, a proven and production-ready consensus engine. CometBFT powers many prominent blockchain networks, including Terra, Thorchain, Celestia, and Cosmos Hub.

   CometBFT operates on a peer-to-peer voting mechanism for achieving consensus, which aligns well with WSTS's requirement for coordination. At its core, CometBFT uses a multi-round voting process where authorities propose, vote on, and commit blocks.

   Crucially, recent advancement of CometBFT's Application Blockchain Interface (ABCI++) [13] provides a native framework for implementing WSTS. ABCI++'s vote extension feature allows adding WSTS secret shares to authority vote exchanges and verifying these shares without modifying CometBFT's core code. Through this mechanism, authorities can securely exchange and verify secret shares needed for DKG [10]. This integration seamlessly combines CometBFT's consensus with WSTS, enhancing maintenance efficiency and ensuring up-to-date security advisories.

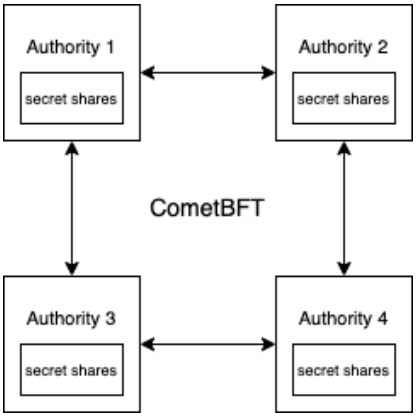

Fig 7. CometBFT coordination for DKG

4. **Vault consensus mechanism**
   With WSTS [8] and EOTS [7], we establish a robust vault consensus

mechanism, coordinated by the CometBFT [12] engine. Our goal is to ensure all honest authorities agree on the legitimate future state of the vault, maintaining security even if up to 1/3 of the signing power is compromised. WSTS enables scalable, efficient multi-signature operations, while EOTS guarantees the immutability of off-chain account records. By implementing a Proof-of-Stake (PoS) model, we overcome the scalability limitations of the global vault's authority set. This approach allows us to balance decentralized security with liquidity flexibility, creating a resilient and user-friendly Bitcoin liquidity layer solution.

a. Global vault structure

Our global vault is built on two fundamental components: a WSTS [8] address that holds deposited BTC, and per-authority off-chain account books that detail depositor amounts.

WSTS signing keys are distributed to authorities based on their stake shares. This process begins with a DKG [10] round, which derives the group public key and redistributes keys to authorities. To unlock the vault, 2/3 of partial signatures from authorities are required, with each partial signature attesting to the same execution sequence of bridge orders. Without sufficient partial signatures, the final aggregated WSTS signature cannot be derived, effectively preventing vault unlock.

We implement a dynamic authority set through periodic redistribution, which removes inactive or insufficiently staked authorities. Active authorities are incentivized to fill these vacancies with rewards and governance power. This approach fosters a competitive, responsible authority ecosystem and allows new, qualified authorities to join.

The off-chain account book records deposit details for efficient bridge execution. CometBFT [12] protocol ensures immutability by automatically disqualifying any mismatched state reports from potential attackers. However, the system is most vulnerable to hostile takeovers when the total native stake is low. To address this weakness, states are checkpointed to Babylon's EOTS [7]. This provides a slashable record of state checkpoints secured by Bitcoin's Proof-of-Work, effectively safeguarding against manipulation with Bitcoin stakes.

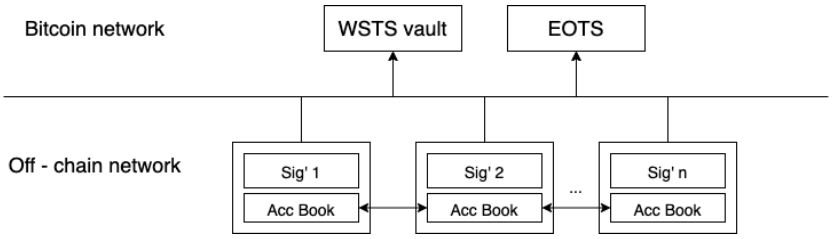

Fig 5. Global vault struct

b. Voting for next vault state

The global vault supports two critical operations: user BTC lock, unlock. Each operation requires a Bitcoin vault transaction to change the vault state. To safeguard these operations, we employ a robust voting mechanism that effectively guards against potential malicious intents of the authority set.

Following the DKG [10] process, each authority securely generates nonces and distributes nonce commitments for future vault transactions. This encryption, using the authority's private key, prevents other authorities from predicting or manipulating signatures.

Authorities process user requests through a secure, transparent procedure within each 10-minute Bitcoin block. They start by collecting batched user requests and deriving a Bitcoin vault transaction based on the current account book. Using pre-shared nonce commitments, authorities calculate partial signatures for this transaction. These signatures are then broadcast publicly, confirming both the account book state and transaction correctness. This public broadcast allows anyone to verify individual signatures, enabling detection of any anomalies.

Crucially, the vault state can only change when at least 2/3 of the partial signatures are aggregated and submitted to the Bitcoin network. This supermajority requirement is crucial in defending against malicious actors, as it prevents any group with less than 2/3 of the signing power from manipulating the vault state. Finally, Bitcoin's Proof-of-Work must validate the aggregated WSTS [8] signature before any vault state change is confirmed.

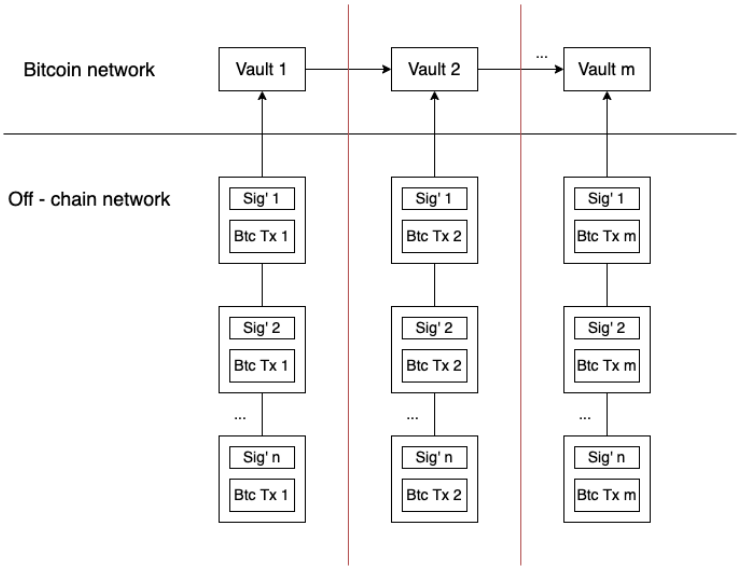

Fig 6. Global vault voting

c. **Safety properties**

If there are less than ⅓ of dishonest partial signature shares and at least one good authority decides on a vault state S, then the WSTS [8] vault will move to no other state rather than state S. Consider a signing round where a good authority decides on state S. This good authority partial signatures, along with ⅔ of other honest partial signatures, commit to state S. Meanwhile, less than ⅓ of Byzantine partial signature shares commit to state S'. During the WSTS check by Bitcoin PoW, there exists only ⅔ of partial signatures committed to state S, thus qualifying the state S and rejecting the state S'. With qualified state S, the vault moves to only state S, and to no other state.

If there is a majority ½ of dishonest partial signature shares, then the WSTS vault will be locked to protect the deposits. Consider a signing round where there are less than ½ of honest participant partial signature shares decided on value S. Meanwhile, a majority of Byzantine partial signature shares commit to state S'. During the WSTS check by Bitcoin PoW, there exists no ⅔ of partial signatures for either state S or S', thus rejecting both states. With no qualifying state, the vault remains locked until an honest supermajority is re - established.

If the vault consensus goes down completely, any authority attempts to lie will result in loss of staked BTC, enforceable through EOTS [7]. Consider a scenario where there exists communications breakdown and no supermajority to advance the consensus. Any authorities lying about state S with a different state S' will result in EOTS private key reveal. The Babylon protocol [11], as a remote security provider, can immediately slash the dishonest authority with the revealed private key.

d. **Liveness properties**

If there are less than ⅓ of dishonest partial signature shares, then the vault consensus does not deadlock. In total of the remaining honest supermajority, there exists two honest parties producing partial signatures for accumulated Bitcoin transaction set S, and subset S'. Since there is no supermajority to advance the WSTS [8] vault, one honest party is locked to set S. Once S' eventually converges to S, there will be enough partial signatures in set S to advance the WSTS vault.

IV. Scalable security, optimized costs

1. Scalable security

The main challenge to scaling security in our system is managing the overhead of communications and computations from DKG [10] and signing rounds. Our goal is to ensure that the total overhead per authority remains

below Bitcoin's 10-minute block time, allowing for timely execution of user orders and frictionless UX.

To demonstrate the scalability of our solution, we conducted comprehensive benchmarks focusing on the computational aspects of our WSTS [8] scheme. These benchmarks were performed on affordable hardware (CPU: AMD Ryzen 9 7900X3D 12-Core, 64 GB DDR5 RAM) to reflect realistic deployment scenarios.

No. of authorities	No. of keys	DKG s/authority
100	500	9.703
100	1000	27.073
100	1500	49.257
100	2000	75.210
150	1000	46.508

No. of authorities	No. of keys	Partial signing s/authority
100	500	0.024
100	1000	0.078
100	1500	0.161
100	2000	0.272
150	1000	0.087

Interpretation of results:

- For 100 authorities with 2000 keys, the total computation time is approximately 75.5 seconds (75.210s for DKG + 0.272s for partial signing), well within our 10-minute target.
- Even as we scale to 150 authorities with 1000 keys, the total computation time remains under 47 seconds.
- The DKG process, being the most computationally intensive, scales roughly linearly with the number of keys, providing predictable performance as we scale further.

While our benchmarks focus on computation, the communication aspect is handled by CometBFT [12], a production-ready consensus engine. Based on CometBFT's documented performance, we expect it to coordinate 100-150 authorities with communication overhead of only a few seconds, complementing our computational performance.

Our WSTS scheme, coupled with CometBFT, provides a robust foundation for scalable security. Current benchmarks demonstrate practical performance for hundreds of authorities and thousands of keys, with clear pathways for further optimization. The benchmark code is available at [14] for transparency and further testing by the community.

2. **Optimizing costs with Schnorr**
   Schnorr signatures, introduced to the Bitcoin network through BIP 340 [2ss], offer a powerful combination of compactness and aggregation. With a fixed size of just 64 bytes, a single Schnorr signature can represent an aggregation of numerous individual signatures. Our vault consensus leverages this feature in its WSTS [8] scheme, allowing us to accumulate any number of partial signatures from authorities into a single on-chain signature, dramatically reducing transaction size and cost.

   To illustrate the cost-efficiency of this approach, let's consider a real-world example. As of Bitcoin block 858673, the average transaction fee is 3 sat/vB [16]. A SegWit Schnorr signature of 64 bytes (or 16 vBytes) would incur 48 sats per block, 6912 sats per 144 blocks in a day, and 207360 sats per month. With an exchange rate of $0.000617 for 1 sats on Kraken, the total cost in dollars would be $128 per month.

   Crucially, this cost remains low regardless of the number of authorities or partial signatures involved in our consensus. In contrast, a traditional multi-signature approach would see costs increase linearly with each additional signer, potentially reaching hundreds of thousands of dollars per month for a comparable level of security as outlined in the Nomic case [4]. The only practical limits to scaling are hardware and algorithmic constraints, both of which can be optimized over time. By leveraging Schnorr signatures in our WSTS scheme, we've created a solution that combines robust security with exceptional cost-efficiency, paving the way for a truly scalable Bitcoin liquidity layer.

V. Proposed Liquidity UX solution: protocol incentivization pools

From a BTC depositor viewpoint, one only cares about depositing BTC securely, to gain more BTC. Currently, most BTC yield protocols reward users with ecosystem native tokens rather than BTC itself. This approach creates unnecessary complexity, forcing users to research and perform multiple exchanges to convert these rewards back to BTC. Our protocol incentivization pool aims to simplify the user experience for BTC depositors while providing ecosystems with a powerful tool to attract and manage liquidity.

An ecosystem in need of growing TVL first opens an incentivization pool with initial deposits in CVBTC (our protocol synthetic BTC) and configurations for reward payout rate, and bridging rate. Interested depositors can commit their CVBTC to an ecosystem's pool, initiating the bridging process. The protocol bridges CVBTC at a preset rate, ensuring controlled liquidity inflow. Depositors start earning rewards once their CVBTC is bridged, with rewards calculated based on their proportion of total

bridged CVBTC and the set payout rate. After all rewards are distributed, depositors are notified and can choose to maintain their position or withdraw their liquidity. At any point, depositors can seamlessly exchange their CVBTC for BTC at a 1:1 ratio, ensuring a single-step process to BTC.

This mechanism offers ecosystems a powerful tool to attract substantial TVL with much smaller incentives. The configurable bridging rate allows ecosystems to control liquidity inflow, giving them time to allocate BTC and demonstrate consistent growth. Ecosystem dApps can offer additional incentives to further attract CVBTC deposits. For instance, a $50,000 incentive pool could potentially draw in $1 million worth of BTC at a 5% annual payout rate. With a bridging rate of $200,000 per month, BTC would steadily flow into the ecosystem, supporting sustained TVL growth over a 5-month period.

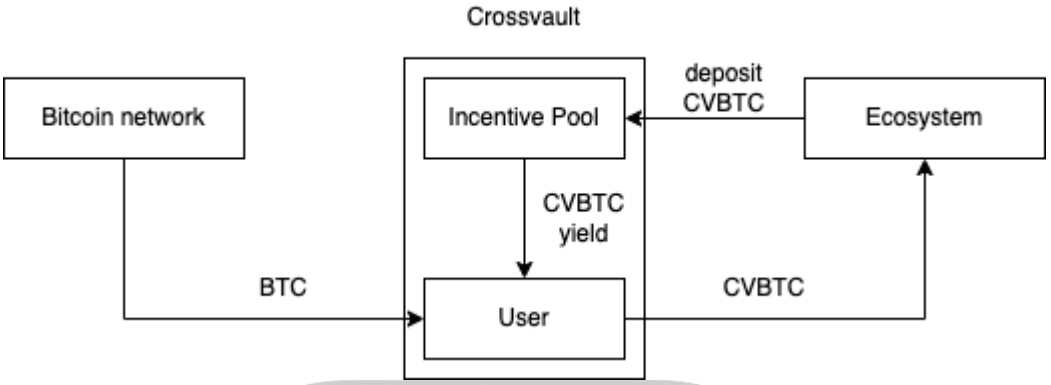

Fig 7. Incentive pool model

VI. Conclusion

The fragmentation of liquidity poses an existential threat to many growing ecosystems, potentially hindering their TVL growth and eroding trust. A Bitcoin liquidity layer has become an inevitable necessity to address this challenge. With the Bitcoin Taproot [1] upgrade and recent advancements on threshold multisig signing schemes, such a solution can finally be realized. Our proposal incorporates cutting-edge research to deliver a securely scalable BTC vault management system with user - centric liquidity UX, while ensuring low verification costs. By leveraging these innovations, we believe our solution has the potential to unlock significant BTC and foster TVL growth across many ecosystems.

VII. References

1. Bitcoin Taproot: https://github.com/bitcoin/bips/blob/master/bip-0341.mediawiki
2. Bitcoin Schnorr: https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki
3. MuSig: https://eprint.iacr.org/2020/1261
4. Nomic weighted threshold multisig: https://gist.github.com/mappum/da11e37f4e90891642a52621594d03f6
5. DLC: https://adiabat.github.io/dlc.pdf

6. DLC.link: https://818995421-files.gitbook.io/~/files/v0/b/gitbook-x-prod.appspot.com/o/spaces%2F1d1QXmk0rpzxLZAKPILL%2Fuploads%2Fbw4fGDDapSudY22xqsK6%2FDLC.Link%20Litepaper.pdf?alt=media&token=0f1bcbe9-b858-48f4-b570-169495240b1e
7. EOTS: https://coins.github.io/stakechains.pdf
8. WSTS: https://trust-machines.github.io/wsts/wsts.pdf
9. FROST: https://eprint.iacr.org/2020/852.pdf
10. DKG: https://www.researchgate.net/publication/2558744_Revisiting_the_Distributed_Key_Generation_for_Discrete-Log_Based_Cryptosystemss
11. Babylon: https://docs.babylonchain.io/papers/btc_staking_litepaper.pdf
12. CometBFT (Tendermint): https://tendermint.com/static/docs/tendermint.pdf
13. ABCI++: https://docs.cometbft.com/v0.37/spec/abci/
14. https://github.com/nghuyenthevinh2000/bitcoin-playground/blob/main/benchmark/frost_test.go
15. https://mempool.space/tx/d11508595752c972585eba0657d0cf432eb01a0f7f486575c19c3c55986403e8
16. https://mempool.space/block/000000000000000002b5eda69ec783b4ad48bddcf11ed2aa6ebb02c111a6a6