SatoshiVM Overview

I. Introduction to Bitcoin ZK Rollup

> SatoshiVM is a decentralized Bitcoin ZK Rollup Layer 2 solution compatible with the Ethereum Virtual Machine (EVM) ecosystem, using native BTC as gas. SatoshiVM bridges the EVM ecosystem with Bitcoin, enabling the Bitcoin ecosystem to issue assets and develop applications.

SatoshiVM possesses the following technological features:

1. **ZK EVM**: SatoshiVM is a versatile ZK Rollup that employs EVM for off-chain computations. This implies that users can interact with SatoshiVM in a manner similar to interacting with Ethereum, and developers can build on top of SatoshiVM just as they would on Ethereum.
2. **ZK Rollup**: SatoshiVM utilizes Rollup technology to bundle multiple transactions into a single batch and validate them on the Bitcoin main network as a single transaction. This ensures the same level of security as the Bitcoin main network, guaranteeing data validity and availability.
3. **ZK Fraud Proofs**: SatoshiVM utilizes technologies such as Taproot and Bitcoin Script to perform on-chain verification of contracts without altering the consensus rules of the Bitcoin network, thereby accomplishing the computation of fraud proofs.
4. **Data Availability**: SatoshiVM must release transaction data on the Bitcoin main network, enabling anyone to verify the correctness of computations executed off the Bitcoin main network.
5. **BTC Native Gas**: SatoshiVM employs native BTC as gas for the EVM. Similar to ETH OP Rollup / ZK Rollup Layer 2 solutions that use ETH as gas for Layer 2, SatoshiVM utilizes BTC as the gas for EVM transactions.

II. SatoshiVM Design Philosophy

> SatoshiVM is built upon a robust design philosophy that encompasses four three aspects: Simplicity, Sustainability, and, of course, Proof of Work (POW). Understanding these principles is crucial as they significantly influence the overall design of SatoshiVM.

Simplicity

Designing with maximum simplicity is one of the primary objectives of SatoshiVM. Ideally, SatoshiVM should consist of the minimum number of components necessary for a secure, scalable, and flexible Layer 2 system. This simplicity provides SatoshiVM with several notable advantages compared to more complex Layer 2 implementations.

Simplicity reduces engineering overhead, allowing us to invest time in developing new features rather than recreating existing ones. SatoshiVM aims to leverage existing, battle-tested Bitcoin code and infrastructure whenever possible.

In the context of critical infrastructure, simplicity also translates to security. Every line of code we write has the potential to introduce unintended bugs. A simpler protocol requires less code, reducing the potential space for errors. This clean and minimal codebase is more accessible to external contributors and auditors. All of these efforts are geared towards maximizing the security and correctness of the SatoshiVM protocol.

Simplicity is also essential for SatoshiVM's long-term vision. By limiting the amount of code we write on Ethereum tools, we can devote most of our efforts to direct utilization of existing code libraries. Engineering contributions to SatoshiVM can directly benefit Bitcoin, and vice versa. As the SatoshiVM protocol solidifies, it becomes increasingly evident that existing resources can be redirected towards core Bitcoin infrastructure.

Sustainability

SatoshiVM is intended for long-term, sustainable operation. Application developers need to ensure that the platform they build remains viable and competitive over an extended period. The design process of SatoshiVM is centered around the idea of long-

term sustainability rather than taking shortcuts in scalability. Ultimately, a scalable system is meaningless if it doesn't sustain its ecosystem.

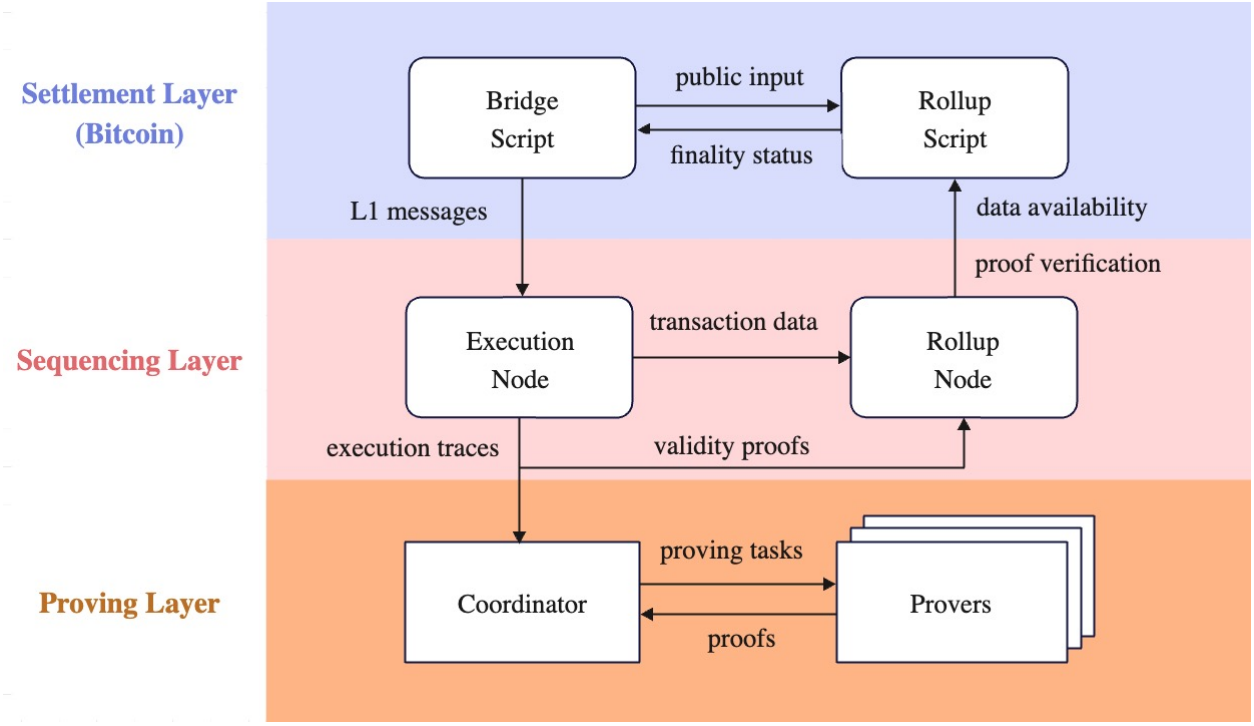
Sustainability positively impacts our protocol design and aligns with our philosophy of simplicity. The more complex the codebase, the harder it becomes for individuals outside the core development team to make meaningful contributions. By keeping our codebase simple, we can foster a larger contributor community that can assist in the long-term maintenance of the protocol.

POW (Proof of Work)

Of course, none of this would be possible without POW. The spirit of Satoshi Nakamoto, represented by Bitcoin, propels this project forward. We believe in a future for Bitcoin where we can redesign the relationships between individuals and organizations.

While SatoshiVM may appear as a standalone blockchain, it is ultimately designed as an extension of Bitcoin. This fact is kept in mind every time we create new features or attempt to simplify existing ones. SatoshiVM can achieve the same level of security as Bitcoin while maintaining simplicity without altering the consensus rules of the Bitcoin network. This is not only for practical reasons but also because the presence of POW enables the success of Bitcoin. We hope that when you examine the design of SatoshiVM, you can see the influence of this philosophy.

III. SatoshiVM Architecture



As depicted in the diagram, the SatoshiVM chain comprises three layers:

1. **Settlement Layer**: This layer provides data availability, ordering, and validation of proofs for the SatoshiVM chain. It allows users and dApps to send messages and assets between Bitcoin and SatoshiVM. Bitcoin serves as the settlement layer, and bridges and rollup scripts are deployed on the Bitcoin network.
2. **Sequencing Layer**: This layer consists of an execution node responsible for executing transactions submitted to the SatoshiVM sequencer and transactions submitted to the L1 bridge script, generating L2 blocks. It also includes a Rollup node that handles batched transactions, publishes transaction data and block information to Bitcoin to ensure data availability, and submits validity proofs to Bitcoin for finality.
3. **Proving Layer**: This layer comprises a coordinator, which assigns proof tasks to provers and relays the generated proofs to the Rollup node to complete finality verification on Bitcoin. It also includes a prover pool, responsible for generating validity proofs that verify the correctness of L2 transactions.

IV. Rollup Protocol

The crucial concept that makes SatoshiVM possible is ZK Rollup. We'll briefly explain how ZK Rollup works at a high level and then delve into why SatoshiVM is built as a ZK Rollup and why we believe it's the best choice for meeting all our design goals.

ZK Rollups

SatoshiVM is a "ZK Rollup," which is a clever way of describing a blockchain whose security relies on another "parent" blockchain's consensus mechanism (e.g., PoW or PoS), rather than providing its own consensus mechanism. In the case of SatoshiVM's mainnet, the parent blockchain is Bitcoin.

Block Storage

In SatoshiVM, L2 blocks are saved with minimal Layer 1 Gas fees to the Bitcoin blockchain. These blocks are submitted as transaction data on Bitcoin, and once these "transactions" are included in blocks with sufficient proofs, no one can alter or censor this transaction data. This is how SatoshiVM's mainnet inherits the availability and integrity guarantees of Bitcoin.

Block Generation

SatoshiVM blocks are primarily produced and managed by a party called the "sequencer," which assists the SatoshiVM Layer 2 network by:

1. Providing transaction confirmations and state updates.
2. Constructing and executing L2 blocks.
3. Submitting user transactions to L1.

In SatoshiVM, the sequencer has a mempool, similar to Bitcoin's L1, but the mempool is private to prevent opportunities for MEV. On the SatoshiVM mainnet, blocks are generated every 3-60 seconds (exact time to be determined), whether they are empty (with no transactions), filled up to the block's gas limit, or somewhere in between.

Transactions reach the sequencer in two ways:

1. Transactions submitted on L1 (whether with assets attached or not, known as deposits) are included in the corresponding L2 block's chain. Each L2 block is

identified by an "epoch" (corresponding to an L1 block, typically occurring a few minutes before the L2 block) and its sequence number within that epoch. The first block of that epoch contains all the deposits that occurred in the corresponding L1 block. If the sequencer attempts to ignore a valid L1 transaction, it will end up with a state inconsistent with the validator, just as if it tried to fake the state by other means. This provides L1 Bitcoin-level censorship resistance to the SatoshiVM mainnet.

2. Transactions submitted directly to the sequencer. The cost of submitting these transactions is much lower than L1 (as there is no separate L1 transaction fee), but the sequencer is the only entity aware of these transactions, making them non-resistant to censorship.

Initially, the SatoshiVM Foundation is the sole block producer on the SatoshiVM mainnet, but it will gradually transition to a decentralized sequencer.

Block Execution

The execution engine receives blocks using two mechanisms:

1. The execution engine can update itself using a peer-to-peer network similar to how L1 execution clients synchronize state across networks.
2. Rollup nodes (implemented as SatoshiVM-Node components) derive L2 blocks from L1. This mechanism is slower but resistant to censorship.

Asset Bridging between Different Layers

SatoshiVM's design enables users to send arbitrary messages between L2 and L1, making asset transfers between the two networks possible. The exact mechanism of this communication depends on the direction of message transmission.

In standard bridging on the SatoshiVM mainnet, this feature will be used, allowing users to deposit assets from the Bitcoin network into the SatoshiVM mainnet and extract the same assets back to the Bitcoin mainnet.

Assets Moving from Bitcoin to SatoshiVM Mainnet

In SatoshiVM terms, transactions from Bitcoin (L1) to the SatoshiVM mainnet (L2) are referred to as deposits, even if they involve no additional assets.

Deposit transactions become part of the blockchain of the first L2 block of an "epoch," corresponding to the L1 block in which the deposit occurred. This L2 block is typically created a few minutes after the corresponding L1 block.

Assets Moving from SatoshiVM Mainnet to Bitcoin

Withdrawals (this term is used for any message from SatoshiVM mainnet to Bitcoin, regardless of whether additional assets are involved) have three stages:

1. Initialization of the withdrawal with an L2 transaction.
2. Waiting for the next output root to be submitted to L1 (this can be seen on the SDK), then submitting the withdrawal proof. This new step allows for off-chain monitoring of withdrawals, making it easier to identify incorrect withdrawals or output roots. This can protect SatoshiVM mainnet users from a class of potential bridge vulnerabilities.
3. Completing the exit after the challenge period ends (takes several hours), finalizing it.

Fraud Proofs

In ZK Rollup, state commitments are published on L1 (Bitcoin in the case of SatoshiVM's mainnet) without any direct proof of the validity of these commitments. Instead, these commitments are considered pending for a period called the "challenge window." If a proposed state commitment goes unchallenged during the challenge window (usually around 2 hours), it is considered final. Once a commitment is deemed final, Bitcoin's smart scripts can safely accept withdrawal proofs based on the SatoshiVM mainnet's state derived from that commitment.

When a state commitment is challenged, it can be invalidated through a "fraud proof" process. If the commitment is successfully challenged, it will be removed from the state commitment chain and eventually replaced by another proposed commitment. It's important to note that successful challenges do not roll back the SatoshiVM mainnet itself, only the published commitments about the state of the chain. The order of transactions and the state of the SatoshiVM mainnet are not altered by fraud proof challenges.

zkVM

SatoshiVM is a generic ZK Rollup that uses a compatible Ethereum virtual machine for off-chain computations. The execution layer of SatoshiVM functions similarly to Ethereum—transactions are batched into blocks and then executed based on the EVM specification. This means that users can interact with SatoshiVM in the same way they interact with Ethereum. It also means that developers can build on SatoshiVM just as they do on Ethereum.

However, achieving compatibility with Ethereum and ZK Rollup presented significant challenges. Rollup must be able to generate proofs that demonstrate off-chain EVM computations were executed correctly. This is the essence of "zkVM": zkVM is a system that proves the correct execution of the EVM.