Surge: A Composable MetaLayer on Bitcoin

tech@surge.build
September 2024

Abstract

Surge is a **Bitcoin-native Composable Metalayer** built to provide a scalable and secure network for launching Rollups, dApps on Bitcoin. By combining Bitcoin's robust security model with advanced cryptographic techniques like **Zero-Knowledge Proofs (ZKPs)**, Surge creates a flexible, privacy-enhanced environment to build high-throughput and cost-efficient dApps developments. This paper outlines the Surge Network's core architecture, including its zk-aggregation, and decentralized verification mechanisms that distinguish it as a unique solution for scaling Bitcoin.

1. Introduction

1.1 What is Surge?

Surge is a **Bitcoin-native Composable Metalayer** that allows developers to launch scalable, privacy-preserving rollups, protocols and dApps with the security of Bitcoin. Utilizing **Zero-Knowledge Proofs (ZKPs)**, Surge creates a flexible and modular network designed to address the inherent scalability challenges of the Bitcoin ecosystem, providing cryptographic assurances and computational efficiency.

1.2 Vision

The vision of Surge is to establish a metalayer network that extends Bitcoin's security model to decentralized applications, protocols, and rollups. Surge offers a composable environment where developers can deploy rollups that interoperate seamlessly while unlocking Bitcoin liquidity and scaling capabilities. Surge emphasizes zk-aggregation, decentralized verification, and an integrated signing and indexing architecture to unlock Bitcoin liquidity while expanding its scalability.

2. Overview

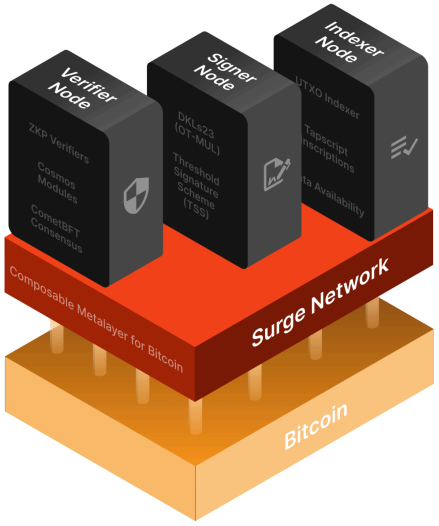
Surge network design is inspired by the Cosmos SDK's modular design to enable decentralized proof validation, while CometBFT ensures Byzantine Fault Tolerant (BFT) consensus across the network. The Rollup Development Kit (RDK) allows

developers to easily integrate ZKPs and deploy secure rollups with the security guarantees of Bitcoin

1. **Verifier Nodes**: Nodes dedicated to zk-proof verification, aggregation, and settling validated proofs on Bitcoin.

2. **Signer Nodes**: These nodes handle transaction signing and manage distributed cryptographic keys to govern BTC vault operations.

3. **Indexer Nodes**: Nodes responsible for indexing UTXOs and ensuring transparent, seamless data retrieval enabling real-time audits.



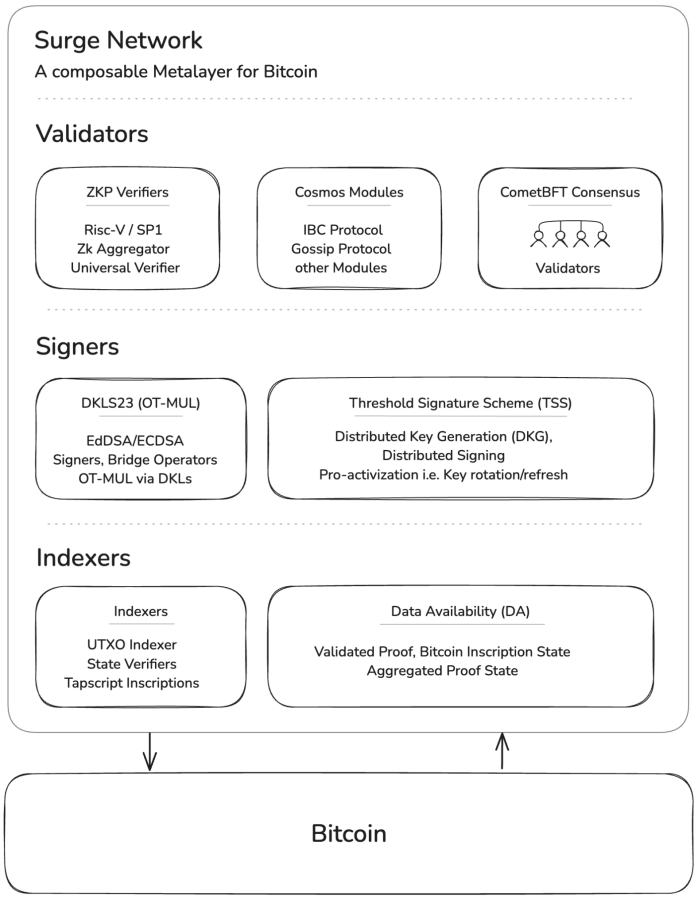
Surge Network Nodes

2.1 Key Components

1. **Verification Layer**: The verification layer processes and validates ZKPs submitted by Rollups. It ensures each proof complies with the network's standards and provides cryptographic verification before settling proofs on Bitcoin.

2. **Cosmos SDK and CometBFT**: Surge leverages the Cosmos SDK's modular architecture alongside CometBFT's **Byzantine** fault-tolerant consensus to create a highly secure, decentralized and scalable proof validation network.

3. **Rollup Development Kit (RDK)**: The **RDK** simplifies the process of deploying rollups on the Surge network. It provides tools to submit Zero-Knowledge Proofs (ZKPs) and facilitates the development of chains that inherit Bitcoin's security properties through ZKP integration.



Network Architecture Overview

3. Bitcoin-Native Metalayer

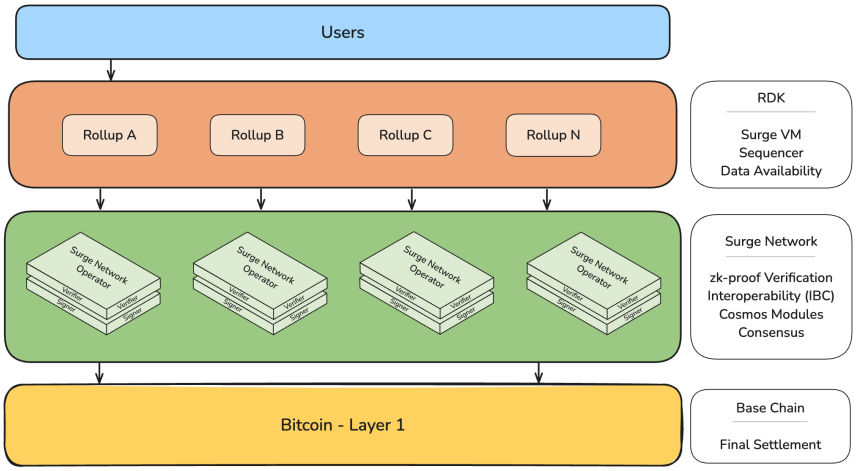
3.1 Why Bitcoin?

Bitcoin's robust **proof-of-work (PoW)** security, immutability, and unparalleled trust model make it an ideal base layer for decentralized applications. Surge anchors rollup & Defi data onto Bitcoin, ensuring that rollups and applications benefit from the economic and cryptographic guarantees Bitcoin provides, creating a robust environment for scalable applications.

3.2 Composability and Interoperability

Surge enables **composable Infrastructure**, allowing rollups to interoperate and share liquidity across chains. Through **IBC (Inter-Blockchain Communication)**, Surge supports seamless communication with other blockchain ecosystems, enabling the transfer of assets, data, and zk-proofs. This composability allows developers to combine functionalities from various blockchains into a single application, increasing flexibility and interoperability.

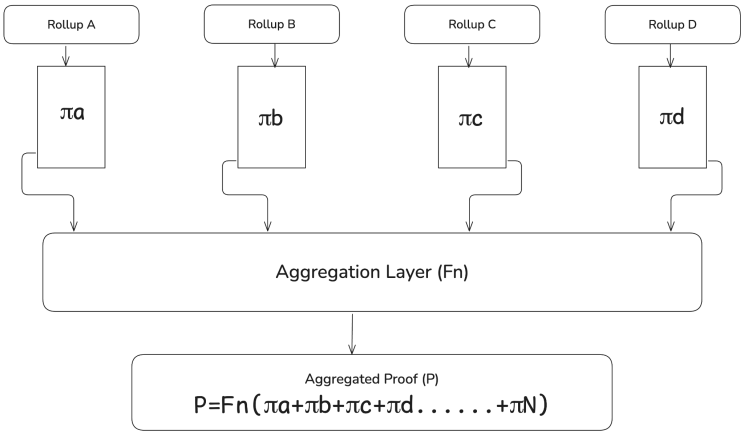
Surge's architecture enables developers to leverage Bitcoin's security while constructing rollups that interact with other chains, creating a multi-layered, interoperable environment where developers can compose novel decentralized solutions.



4. Verifier Nodes

4.1 ZK Aggregation and Verification Layer

Surge uses **zk-aggregation** to address the limitations of traditional rollups, where each individual transaction needs to be submitted to the base layer. Instead, multiple proofs (n proofs) are aggregated into a single concise proof (a proof), significantly reducing the computational overhead and fees associated with on-chain transactions. This aggregation also enhances throughput by enabling the verification of entire histories of rollups using a single cryptographic proof.



Proof Aggregation

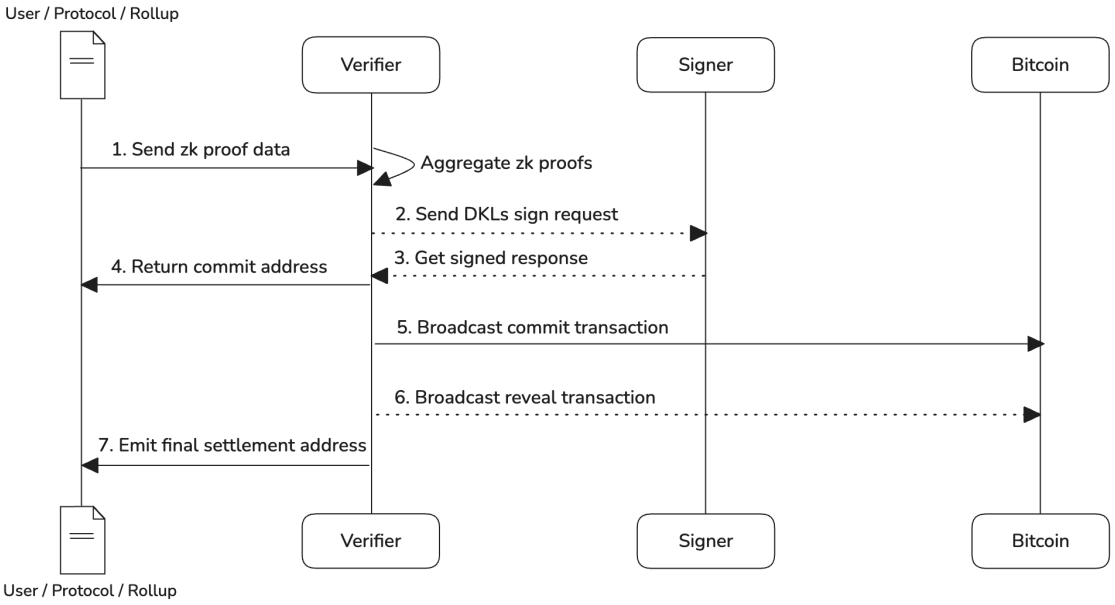
4.2 Proof Verification

The **Surge Network** begins its verification process when a Layer-2 rollup submits a **ZKP** on-chain. Each proof batch contains a cryptographic commitment to the state transition of the rollup. The Surge network leverages **Cosmos SDK** with **Proof-of-Stake (PoS)** consensus to validate that the ZKP is correct and adheres to the predefined conditions. Upon successful verification, the proof is aggregated into the **zk-State Root** and prepared for inscription onto Bitcoin.

4.3 Commit-Reveal Framework

Surge uses **commit-reveal transactions** to inscribe validated ZKPs into individual satoshis of Bitcoin. The commit-reveal process for Bitcoin inscriptions involves,

- Wrapping data into a Taproot script
- Injecting the script into the witness portion of a Bitcoin transaction
- Broadcasting the inscribed sat to the network in two transactions: a commit transaction and a reveal transaction



Commit-Reveal framework journey

4.3.1 Commit Phase

The commit phase starts with a transaction that contains a specific output type, such as **Taproot (P2TR)**, **Pay-to-Witness Script Hash (P2WSH)**, or other compatible outputs like **P2SH-P2WSH**. These types of outputs require that a script is revealed at a later stage in order to unlock the funds.

4.3.2 Reveal Phase

The reveal phase is triggered once the commit transaction has been included in a block, and it involves a follow-up transaction, which must meet the following conditions:

- It must use an unspent output from the commit phase as its first input.
- The first output must include an OP_RETURN output that carries a small payload. This payload contains:
    - The first two bytes identify the network
    - The following byte signifies the ZKP operation code and whether the script is part of the witness data or redeem script.
    - The version number for the script is included to define the type of transaction.

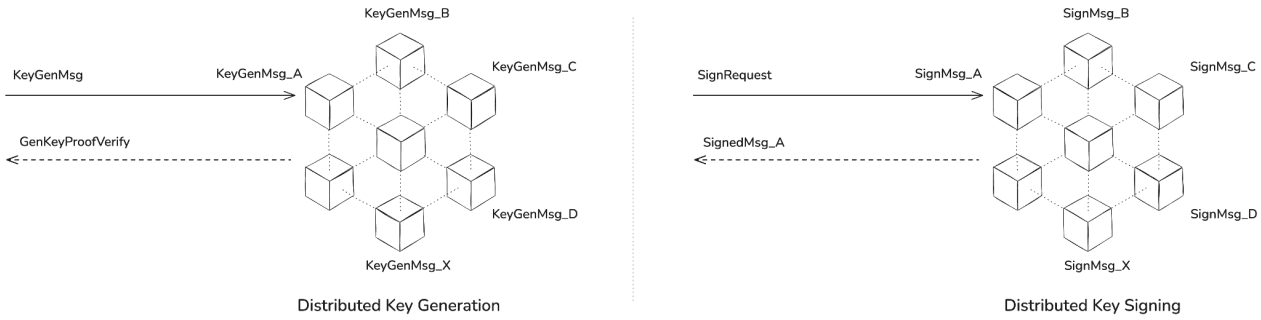
Once the reveal transaction is confirmed, the ZKP data committed in the original transaction becomes visible on-chain. The system detects this reveal, allowing the proof to be processed according to the operation code specified in the reveal transaction.

5. Signer Nodes

5.1 Decentralized Signing and Inscription

The **Signer Node** manages decentralized signing operations using the **DKLS23 EdDSA/ECDSA** scheme, ensuring secure, decentralized management of cryptographic keys for inscribing ZKPs into Bitcoin.

- **Decentralized Key Management**: A **quorum-based signing process** of nodes is required to sign a transaction, ensuring a decentralized approach to key management.



DKLs key management flow

- **BTC Vault Operations**: Signer nodes manage BTC vault operations, enabling secure peg-in and peg-out transactions for rollups. BTC is secured and governed through decentralized signing mechanisms.

5.2 Security and Resilience via TSS Network

The Signer Node operates within a standalone **TSS (Threshold Signature Scheme) network**, ensuring that no single node controls the signing process and utilizes **M of N consensus** to securely manage key shares and signatures.

- **Distributed Key Generation (DKG)**: Key generation is distributed across the TSS network, ensuring that no single party has access to the entire private key.

- **Pro-activization, Key rotation**: The system includes regular key rotation and refresh processes, ensuring that keys are never permanently exposed and remain secure even in the face of node compromise..

6. Indexer Node

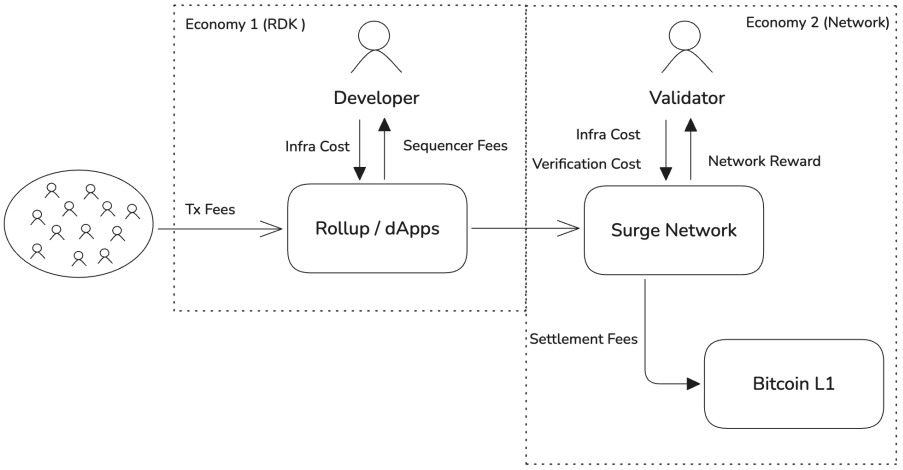
6.1 Tracking and Auditing

The **Surge Indexer** tracks UTXOs that contain ZKP data, providing immutable records and real-time data access that enables efficient auditing and compliance for decentralized applications.

- **Efficient Data Access**: Provides fast querying of transaction data across rollups.
- **Immutable Record**: Once a proof is inscribed on Bitcoin, the indexer guarantees its immutability, ensuring long-term integrity and availability for auditing and verification.

7. Network Economics

Validators are required to stake SURGE tokens to secure the network, with slashing mechanisms in place to penalize malicious behavior or downtime. As the network grows, SURGE staking ensures that the security scales in tandem with transaction throughput.



Network Economics

7.1 Economic Incentives for Verifiers

The Surge Network is secured by a **Proof-of-Stake (PoS)** consensus mechanism, with a fixed maximum number of validators that will be defined as the network evolves. Validators are responsible for verifying transactions and securing the network through cryptographic validation.

7.2 Surge Token as Gas and Incentive

The **Surge Token** is the core utility and gas token of the network. All transactions within the Surge Network, including rollup interactions, are paid for using SURGE. A native swapping mechanism allows users to pay gas fees in **BTC**, which is

automatically converted into SURGE for settlement. Gas fees are collected and distributed among validators and stakers, incentivizing network participation and security.

7.3 Staking and Validator Incentives

Validators must stake SURGE tokens to participate in consensus, ensuring they are economically aligned with network security. Validators are rewarded with SURGE from transaction fees and block rewards. However, malicious, such as downtime or validating incorrect proofs, can lead to slashing or removal from the active set. Reliable performance is key to maximizing rewards and maintaining validator status.

8. Security & Trust Assumptions

8.1 Zero-Knowledge Proof Security

Surge relies on **ZKPs** to ensure the correctness of off-chain computations while preserving privacy. Each rollup submits ZKPs summarizing batches of transactions, which the network verifies before inscribing them on Bitcoin. This guarantees secure state transitions without compromising scalability.

8.2 Threshold Signature Scheme (TSS) for Decentralized Signing

The **Threshold Signature Scheme (TSS)** distributes signing authority across multiple nodes, ensuring no single node controls the entire key. This decentralized approach prevents key compromise by requiring a quorum of nodes to generate valid signatures for rollup transactions and BTC vault operations.

8.3 Byzantine Fault Tolerance (BFT) in Verifier Nodes

Byzantine Fault Tolerant (BFT) consensus ensures that even if up to one-third of nodes act maliciously, the network remains operational. This is complemented by a zero-trust model, where no single participant is inherently trusted, and security is continuously enforced through cryptographic proofs.

8.4 Pro-Activation for Key Security

Regular key rotation in the **TSS network** prevents long-term exposure of cryptographic materials. Even if a subset of nodes is compromised, new key shares are frequently generated, maintaining security across the network.

8.5 Slashing for Malicious Actors

Malicious behavior, such as submitting incorrect ZKPs or failing to participate in consensus, results in slashing of staked tokens. This economic penalty deters dishonesty, ensuring verifiers act in the best interests of the network.

8.6 Trust Minimization with Zero-Trust Architecture

Surge follows a **zero-trust architecture**, where no single participant is trusted by default. Security is maintained through cryptographic proofs and decentralized consensus, reducing reliance on central authorities.

9. Conclusion

Surge Network establishes itself as the **Bitcoin-native composable metalayer**, designed to scale Bitcoin by enabling rollups with zk-aggregation, decentralized verification, and robust signing and indexing modules. Surge provides developers with a high-performance framework, utilizing Bitcoin's unparalleled security for building scalable decentralized applications. By combining the scalability and privacy of zk-rollups with Bitcoin's security, Surge is positioned to be the preferred platform for developers seeking to build advanced, scalable solutions.

10. Future Work and Research required

As Surge continues to evolve as the Bitcoin native metalayer, several areas of future work and research have been identified to further enhance the capabilities and reach of the ecosystem. These areas represent both the immediate next steps and the long-term vision for Surge, aiming to address current limitations and unlock new possibilities for decentralized applications.

1. **Cross-Rollup Communication:** Develop enhanced **Inter-Rollup Communication Protocols (IRC)** to enable seamless data and asset transfers between rollups on the Surge network, improving interoperability and liquidity sharing.

2. **Advanced zk-Proof Aggregation Techniques:** Continue enhancing **zk-proof aggregation** to reduce on-chain costs, exploring recursive proof techniques that further optimize scalability for high-throughput applications.

3. **Decentralized Sequencing:** Explore **decentralized sequencing** methods to optimize transaction ordering and throughput, reducing reliance on

centralized entities and improving network resilience against censorship.

4. **Bitcoin Security:** Leverage Bitcoin's robust **proof-of-work security** to further enhance Surge's settlement layer, ensuring that all rollup activities inherit Bitcoin's immutability and trust model.

5. **Minimizing Trust Assumptions:** Develop mechanisms to **minimize trust assumptions**, focusing on cryptographic guarantees and zero-trust architecture to eliminate reliance on centralized entities while ensuring decentralized validation and security.