

Junkcoin Bridge: Production FROST MPC Custody, UUPS EVM Verification, and the Multichain PSOB Architecture

A Production-Deployed Cross-Chain Bridge Connecting Junkcoin (JKC) to EVM Networks and the Proof-of-Sibling-Block (PSOB) Trust-Minimized Roadmap

The Junkcoin Core Community & DeDoo Bridge Project

senasgr · senasgr@s3na.xyz

<https://junk-coin.com> · <https://dedoo.xyz> · <https://dedoo.xyz/research>

Version 2.2 · October 2026

Abstract

We present the architecture, security model, and multichain scaling roadmap of the **DeDoo Junkcoin Bridge**, a cross-chain system connecting the Junkcoin (JKC) Proof-of-Work / Auxiliary Proof-of-Work (AuxPoW) UTXO blockchain to EVM-compatible networks, with Arbitrum One as the initial mainnet production target, alongside Base and LitVM (LiteForge zkLTC Rollup) environments. In current production (Phase 1), the bridge operates a decentralized *2-of-3 FROST (Flexible Round-Optimized Schnorr Threshold)* multi-party computation (MPC) protocol over BIP-340 Taproot Schnorr outputs on the JKC chain, paired with an on-chain UUPS upgradeable smart contract architecture on Arbitrum One minting Wrapped Junkcoin (**wJKC**, 8 decimals, 54,000,000 supply cap). Validators communicate over an authenticated peer-to-peer mTLS QUIC network, indexed by distributed rindexer engines.

Building upon foundational research from dedoo.xyz/research, we formalize the **PSOB (Proof-of-Sibling-Block / Partial-Signature Output Bridge)** multichain architecture. PSOB leverages the AuxPoW consensus invariant—wherein multiple merged-mined chains share a unified Merkle commitment in the parent Litecoin coinbase—to achieve: (1) *fully trustless ZK peg-in minting* via an SP1 zero-knowledge guest circuit that validates Script PoW and sibling-block commitments without trusted relayers; (2) *universal multichain dispatching* enabling a single ZK proving circuit to route assets across Arbitrum One, Base, LitVM, and Ethereum L1; (3) *decentralized threshold peg-out settlement* using MuSig2/FROST key aggregation with deterministic leader rotation, on-chain claim boards, sign-once state enforcement, and optional ZK-proof finalization; and (4) *native consensus synergy with LitVM (Litecoin L2 zk-Rollup)*, where both Junkcoin AuxPoW and LitVM share the exact same Litecoin L1 parent anchor, granting coupled reorganization immunity and frictionless zero-oracle settlement. This paper details both the live Phase 1 production deployment and the complete engineering roadmap toward trustless PSOB verification.

Disclaimer: This document is provided for technical, architectural, and informational purposes only. It does not constitute financial or investment advice. Smart contract and cryptographic protocols involve risk.

Contents

1	Introduction	2
2	Background and Cryptographic Foundations	2
2.1	Junkcoin (JKC) Architecture and AuxPoW Consensus	2
2.2	BIP-340 Taproot Schnorr Signatures and FROST MPC	3
2.3	EVM Mint/Burn Token Model and Invariants	3
3	Phase 1: Production 2-of-3 FROST MPC Bridge	4
3.1	On-Chain Smart Contract Layer (UUPS Pattern)	4
3.2	Deposit Protocol (Junkcoin UTXO $\rightarrow$ EVM)	4
3.3	Withdrawal Protocol (EVM $\rightarrow$ Junkcoin UTXO)	5
4	The Multichain PSOB Architecture (dedoo.xyz/research)	5
4.1	The Proof-of-Sibling-Block (PSob) Invariant	5
4.2	ZK Peg-In Circuit (SP1 RISC-V zkVM)	6
4.2.1	Unified 352-Byte Journal Specification	6
4.3	Universal Multichain Routing	7
4.4	PSOB Peg-Out: MuSig2 Key Aggregation and Claim Board	7
4.5	PSOB and LitVM: Direct Consensus Synergy between AuxPoW and Litecoin L2	8
5	Security Analysis and Threat Model	9
6	Deployment Roadmap and Active Contracts	9
6.1	Phased Rollout Matrix	9
6.2	Production Contract Registry	9
6.2.1	Arbitrum One Mainnet (Chain ID: 42161)	9
6.2.2	Testnet Deployments	10
7	Conclusion	10

1. Introduction

Proof-of-Work (PoW) UTXO blockchains—originating with Bitcoin and continuing through Litecoin and its Auxiliary Proof-of-Work (AuxPoW) merge-mined family such as Junkcoin (JKC)—represent some of the most resilient, secure, and economically sound distributed ledgers. Their scripting environments, however, are deliberately restricted: complex smart contracts, automated market makers (AMMs), decentralized lending protocols, and ERC-20 composability cannot execute directly on the base UTXO layer.

Conversely, Ethereum Virtual Machine (EVM) networks and modern Layer-2 rollups (such as Arbitrum One and Base), alongside specialized zero-knowledge rollups anchored directly to Litecoin (such as LitVM LiteForge zkLTC Rollup), provide highly programmable execution environments but lack native introspection into the state or monetary supply of independent UTXO blockchains. Bridging Junkcoin to EVM networks unlocks vital utility: JKC holders can engage with decentralized liquidity pools, collateralized borrowing, and multi-asset trading, while preserving the 54,000,000 hard-capped supply and energy-backed security of native Junkcoin.

Historically, cross-chain bridges have suffered substantial security failures due to brittle custody architectures, compromised single-key hot wallets, or vague trust assumptions [1]. The core challenge in bridging UTXO chains to EVMs lies in *bidirectional state verification*:

1. **Peg-In (UTXO $\rightarrow$ EVM):** The EVM contract must definitively verify that native coins have been locked into bridge custody to a safe confirmation depth, without trusting a single centralized relayer.
2. **Peg-Out (EVM $\rightarrow$ UTXO):** The bridge must securely release native coins on the UTXO chain only after equivalent wrapped tokens are burned on EVM, without ever exposing a monolithic private key to any single host or process.

Contributions of this Paper. This paper presents the technical specification and operational model of the DeDoo Junkcoin Bridge system across its evolutionary phases:

- **Phase 1 (Live Production):** A 2-of-3 threshold FROST MPC custody scheme on Junkcoin utilizing BIP-340 Taproot Schnorr signatures, coupled with an audited UUPS upgradeable ERC-1967 smart contract suite on Arbitrum One Mainnet, monitored by an independent P2P validator network running rindexer.
- **The PSOB Multichain Architecture:** The mathematical and cryptographic specification of the Proof-of-Sibling-Block (PSOB) protocol from dedoo.xyz/research. PSOB harnesses AuxPoW parent-coinbase commitments to enable a single zero-knowledge circuit (SP1 RISC-V zkVM) to verify transactions across any AuxPoW sibling chain, routing wrapped assets across Arbitrum One, Base, LitVM, and Ethereum L1.
- **Consensus Synergy with LitVM (Litecoin L2):** An analysis of the symbiotic relationship between AuxPoW merge-mined chains and LitVM (Litecoin Layer-2 zk-Rollup). Because both share the Litecoin L1 parent anchor, PSOB achieves frictionless light-client verification and complete immunity to cross-chain reorganization desynchronization.
- **Security & Threat Model:** Formal threat analysis, sign-once non-equivocation rules, deterministic leader rotation, reorg resistance thresholds, and proof-backed peg-out finalization.

2. Background and Cryptographic Foundations

2.1 Junkcoin (JKC) Architecture and AuxPoW Consensus

Junkcoin is an established Scrypt-based Proof-of-Work blockchain, utilizing an auxiliary proof-of-work mechanism merged-mined with Litecoin (LTC). Its key parameters include:

- **Max Supply:** 54,000,000 JKC hard cap.
- **Base Unit:** 8 decimal precision (1 JKC = 10^8 satoshis).
- **Consensus:** Scrypt AuxPoW, anchored to parent Litecoin block mining.
- **Address Formats:** P2PKH (7...), P2SH (Q...), and native BIP-341 Taproot SegWit v1 (jc1p...).

In the AuxPoW construction, a parent Litecoin miner includes a commitment to one or more auxiliary chains within the parent coinbase transaction. An auxiliary block header is valid if it contains a **CAuxPow** witness proving that the auxiliary block header hash is present in a Merkle tree committed inside the parent Litecoin coinbase, and that the parent Litecoin header meets the auxiliary chain's difficulty target (`expand_target(nBits)`).

2.2 BIP-340 Taproot Schnorr Signatures and FROST MPC

Legacy federated bridges on UTXO chains typically rely on BIP-16 P2SH multisig scripts (`OP_CHECKMULTISIG`), which expose all participant public keys and quorum thresholds on-chain, consuming elevated transaction fees and leaking validator topologies.

The Junkcoin Bridge utilizes BIP-340/341 Taproot. Under Taproot, any m -of- n threshold policy can be expressed off-chain using the **FROST (Flexible Round-Optimized Schnorr Threshold)** protocol [2]. During distributed key generation (DKG), n validators jointly establish a single 32-byte x-only aggregate group public key:

$$Q \in \mathbb{G}$$

Each participant $i \in \{1, \dots, n\}$ holds a private Shamir secret share s_i . When signing a payout transaction:

1. **Round 1 (Nonce Commitment):** Participating signers generate and exchange hiding and binding nonce pairs $(D_{i,j}, E_{i,j})$.
2. **Round 2 (Partial Signatures):** Given the transaction signature hash m , signers compute partial signatures $z_i = d_i + (e_i \cdot \rho_i) + \lambda_i s_i c \pmod{q}$, where $c = H_{\text{BIP0340}}(R \parallel Q \parallel m)$.
3. **Aggregation:** The coordinator aggregates partial signatures into a standard 64-byte BIP-340 Schnorr signature (R, s) .

On the Junkcoin blockchain, the resulting transaction spends from a standard Pay-to-Taproot output (jc1p...). To outside observers and block explorers, the spend appears identical to an ordinary single-key Taproot transaction, providing maximum privacy, lowest sat/vB fees, and strict non-reconstruction of the master key.

2.3 EVM Mint/Burn Token Model and Invariants

On the EVM layer, bridged assets are represented by the Wrapped Junkcoin (**wJKC**) contract. **wJKC** conforms to ERC-20 with 8 decimals, exactly matching the 8 decimals of native JKC.

Definition 1 (Global Reserve Invariant). *At any EVM block height h , let S_h denote the total minted supply of wJKC across all supported EVM chains, and let R_h denote the balance of native JKC held in the Taproot custody wallet with at least $k_{\min}$ confirmations on the JKC blockchain. Under honest execution:*

$$S_h \leq R_h$$

Equality holds asymptotically in the absence of fees. When bridge fees f are deducted during bridging operations, $S_h < R_h$, representing surplus reserves accumulated in the protocol treasury.

3. Phase 1: Production 2-of-3 FROST MPC Bridge

The Phase 1 bridge architecture is live in production connecting the Junkcoin native chain with Arbitrum One Mainnet, along with testnet deployments on Base Sepolia and LitVM LiteForge.

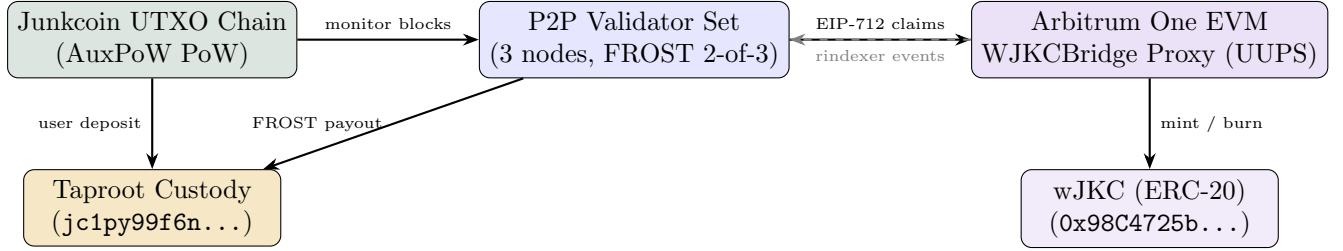


Figure 1: Phase 1 Production Architecture: Junkcoin to Arbitrum One FROST MPC Bridge.

3.1 On-Chain Smart Contract Layer (UUPS Pattern)

The EVM side is structured around the Universal Upgradeable Proxy Standard (UUPS, ERC-1967):

- **Wrapped Junkcoin (WrappedJunkcoin.sol):** ERC-20 token with `name = "Wrapped Junkcoin"`, `symbol = "wJKC"`, `decimals = 8`, and `MAX_SUPPLY = 54,000,000 * 108`. Minting and burning authority is restricted strictly to the authorized WJKCBridge proxy contract.
- **Bridge Proxy (WJKCBridge.sol):** UUPS proxy holding the bridge state, nonces, validator public key registry, and threshold logic. Upgrades require multisig owner authorization.
- **Arbitrum One Mainnet Deployment Parameters:**
 - Token Address: 0x98C4725bA9531eD1C2C40A2cEb29a016c014d9e1
 - Bridge Proxy Address: 0x42E455151c222Afd2A36E2dB94B43c304B3d17f5
 - Implementation Address: 0xd3A85c9e40fBfc29EAd2A3D15d89eB49157829D6
 - Taproot Custody Address: jc1py99f6nn90jgu2h8lygc4f7ysxrezd97tedegehmw967wgypcdqzsfrfcus
 - Protocol Fee: 100 bps (1.00%), adjustable via governance timelock.

3.2 Deposit Protocol (Junkcoin UTXO → EVM)

1. **Deposit UTXO:** The user broadcasts a transaction on the Junkcoin blockchain transferring V satoshis to the Taproot custody address (jc1py99f...), including an `OP_RETURN` output with the 20-byte recipient EVM address:

$$\text{scriptPubKey} = \text{OP_RETURN} \langle \text{EVM Recipient} \rangle$$

2. **Confirmation Gate:** Independent validator nodes monitor the JKC mempool and blocks via local Junkcoind nodes and Electrs RPC. Once the transaction achieves $k_{\min} \geq 6$ confirmations, each validator extracts the deposit tuple $(txid, vout, amountSats, recipient, chainId)$.
3. **EIP-712 Structured Attestation:** Each validator signs an EIP-712 typed digest:

$$\text{DEPOSITCLAIM}(txid, vout, amountSats, recipient, chainId, epoch)$$

4. **Registration & Minting:** The aggregated claim signatures (≥ 2 of 3) are submitted to WJKCBridge.sol. The contract verifies the signatures against the active validator set, enforces replay prevention via mapping `processedTxids`, deducts the 1% bridge fee into protocol reserves, and calls `wJKC.mint(recipient, netAmount)`.

3.3 Withdrawal Protocol (EVM $\rightarrow$ Junkcoin UTXO)

1. **Initiation & Burn:** The user calls `initiateWithdrawal(amountSats, jkcAddress)` on the bridge contract on Arbitrum One. The contract immediately burns `amountSats` of `wJKC` and emits a `WithdrawalInitiated` event containing a unique `withdrawalId`.
2. **Indexing via rindexer:** Distributed `rindexer` instances running on each validator node index the burn event in real time and notify the local bridge daemon.
3. **Deterministic Leader Election:** To avoid mempool collision, a round leader is elected deterministically from on-chain event data:

$$\text{leader} = H(\text{withdrawalId} \parallel \text{round}) \bmod 3$$

4. FROST Round Execution:

- The leader constructs an unsigned UTXO transaction paying $(\text{amountSats} - \text{networkFee})$ to the user's `jkcAddress`, adding an `OP_RETURN` containing the EVM burn transaction hash.
 - Validators inspect the UTXO inputs, outputs, and amount against their local verified EVM state.
 - The validators execute FROST Round 1 (nonce generation) and Round 2 (partial signing) across their P2P network.
 - The leader aggregates the partial signatures into a single 64-byte BIP-340 Schnorr signature and broadcasts the raw transaction to the Junkcoin network.
5. **Confirmation on EVM:** Once the UTXO transaction is confirmed on Junkcoin, any validator submits `confirmWithdrawal(withdrawalId, jkcTxid)`, marking the withdrawal completed on-chain.

4. The Multichain PSOB Architecture (dedoo.xyz/research)

While the Phase 1 FROST MPC bridge provides robust security, it relies on an honest-majority federation of operators. Based on the theoretical framework published in dedoo.xyz/research, we specify the **PSOB (Proof-of-Sibling-Block / Partial-Signature Output Bridge)** architecture. PSOB transforms the bridge into a trust-minimized, multichain settlement protocol spanning Arbitrum One, Base, LitVM (LiteForge zkLTC Rollup), and Ethereum L1.

4.1 The Proof-of-Sibling-Block (PSob) Invariant

Auxiliary Proof-of-Work chains (Junkcoin, Craftcoin, Librecoin) share parent Litecoin miners. When a Litecoin block merges-mines N auxiliary chains, the miner constructs a Merkle tree of auxiliary block headers. The root of this tree is committed inside the Litecoin coinbase transaction using a standardized 4-byte marker:

0xfa 0xbe 0x6d 0x6d $\parallel$ chain_merkle_root (32 bytes reversed) $\parallel$ tree_size (4 bytes) $\parallel$
nonce (4 bytes)

Property 1 (PSob Consensus Invariant). *Let B_{LTC} be a valid Litecoin parent block with header H_{LTC} satisfying $\text{scrypt}(H_{LTC}) \leq \text{target}_{LTC}$. If auxiliary blocks B_{JKC} and B_{AUX} are committed within the same parent coinbase T_{coinbase} , then B_{JKC} and B_{AUX} are **Sibling Blocks**. Their validity and proof-of-work can be simultaneously verified by proving their respective Merkle membership paths back to the single Merkle root in T_{coinbase} .*

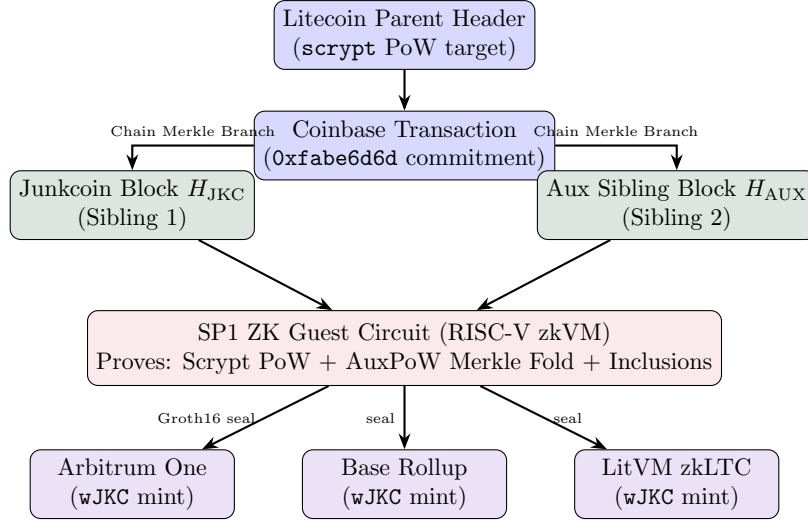


Figure 2: Multichain PSOB Architecture: Single ZK-Circuit Verification across Sibling Chains.

4.2 ZK Peg-In Circuit (SP1 RISC-V zkVM)

Under PSOB, peg-in requires **zero relayer trust**. A user or permissionless prover generates a Succinct SP1 RISC-V zero-knowledge proof proving the deposit directly from raw block bytes:

1. **Chain Continuity:** Verifies a sequence of auxiliary headers $[H_{\text{checkpoint}+1}, \dots, H_{\text{deposit}}]$ satisfying $\text{sha256d}(H_{i-1}) == H_i.\text{prev_hash}$.
2. **Script PoW Verification:** Verifies that each parent header satisfies the Script hash target: $\text{script}(H_i.\text{parent}) \leq \text{expand_target}(H_i.\text{nBits})$.
3. **Double Sibling-Merkle Verification:**
 - *Sibling-TX Proof:* Merkle fold branch proving the coinbase transaction is included in the parent Litecoin header’s Merkle root.
 - *Sibling-Block Proof:* Merkle fold branch proving the Junkcoin header hash is included in the sibling chain tree committed in the parent coinbase script.
4. **Structure-Aware Deposit Parsing:** Walks the raw deposit transaction bytes, parses output scriptPubKeys, confirms payment to the Taproot custody address, and decodes the 20-byte EVM recipient from the OP_RETURN payload.

4.2.1 Unified 352-Byte Journal Specification

The SP1 guest program outputs an invariant 352-byte ABI-encoded **Journal** struct:

Listing 1: PSOB 352-Byte Journal Layout

```

1 struct Journal {
2     bytes32 depositBlockHash;    // sha256d of deposit header
3     bytes32 depositTxid;        // sha256d of raw deposit tx
4     uint256 amountSats;         // satoshis transferred to custody
5     address recipient;          // destination EVM recipient
6     uint32 confirmations;       // confirmation block depth
7     bytes32 checkpointHash;     // trusted anchor checkpoint
8     uint32 powLimitBits;        // consensus powLimit gate
9     uint256 provenWork;         // cumulative script work sum
10    bytes20 custodyHash160;      // custody pubkey hash
11    uint256 custodyEpoch;        // custody rotation epoch

```

```

12     uint32    chainId;                // target destination chain ID
13 }

```

4.3 Universal Multichain Routing

A transformative advantage of the PSOB design is that a **single zero-knowledge circuit** (one `imageId`) serves all destination EVM chains and all AuxPoW sibling chains:

Listing 2: Multichain PSOB Routing Logic

```

1  function pegInWithZkProof(bytes calldata seal, bytes calldata
   journalBytes) external {
2      // 1. Verify Groth16 zk-SNARK against immutable SP1 imageId
3      verifier.verify(seal, psobProgramImageId, sha256(journalBytes));
4
5      // 2. Decode Journal
6      Journal memory j = abi.decode(journalBytes, (Journal));
7
8      // 3. Verify destination chain and parameters
9      require(j.chainId == block.chainid, "Invalid_destination_chain");
10     require(j.confirmations >= minConfirmations, "Insufficient_depth");
11
12     require(j.provenWork >= minProvenWork, "Insufficient_work");
13     require(!processedDeposits[j.depositTxid], "Already_claimed");
14
15     processedDeposits[j.depositTxid] = true;
16     wjkcToken.mint(j.recipient, j.amountSats - calculateFee(j.
       amountSats));
17 }

```

By deploying this identical logic across Arbitrum One, Base, and LitVM, wrapped token minting becomes instant, auditable, and entirely free of federated committee trust.

4.4 PSOB Peg-Out: MuSig2 Key Aggregation and Claim Board

For peg-out operations (EVM $\rightarrow$ UTXO), where native UTXO scripts cannot yet cheaply verify EVM zero-knowledge proofs on-chain, PSOB introduces a trust-minimized **Decentralized Publisher Set**:

- **MuSig2 Key Aggregation:** Up to $n = 20$ publishers collaboratively form the Taproot aggregate public key. Unlike legacy MPC schemes, MuSig2 requires only two simple signing rounds and offers rigorous algebraic security without complex zero-knowledge proofs of secret shares.
- **On-Chain Claim Board:** When a user initiates a withdrawal on EVM, the request is registered on an on-chain *Claim Board*. Any publisher can stake a bond to claim the right to lead the payout batch.
- **Sign-Once Non-Equivocation Enforcement:** Each publisher maintains an append-only local SQLite database. A publisher strictly refuses to generate or countersign any UTXO transaction with inputs or outputs conflicting with the verified EVM burn record.
- **Proof-Backed Finalization:** Once the payout is broadcast, a prover submits an SP1 ZK proof of the UTXO confirmation back to the EVM Claim Board, automatically unlocking the publisher's bond and releasing protocol fee rewards.

4.5 PSOB and LitVM: Direct Consensus Synergy between AuxPoW and Litecoin L2

A unique and profound architectural synergy exists between AuxPoW chains (such as Junkcoin) and **LitVM (LiteForge zkLTC Rollup)**. LitVM is an EVM-compatible Layer-2 zero-knowledge rollup settled directly on the **Litecoin L1** blockchain. Because Junkcoin is an AuxPoW chain merge-mined with Litecoin, both systems share the same foundational consensus anchor:

1. **Shared Root of Trust (Litecoin L1 Anchor):** Traditional cross-chain bridges connect two entirely independent consensus domains (e.g., Bitcoin to Ethereum), requiring either an expensive on-chain light client re-verifying foreign PoW headers or an external relayer federation. In stark contrast, **Junkcoin AuxPoW and LitVM share the exact same physical mining root: Litecoin L1**. LitVM's sequencer and verification contracts natively track and verify Litecoin block headers. Because Junkcoin block hashes are already committed inside the parent Litecoin coinbase transaction via the AuxPoW Merkle commitment (0xfabe6d6d), LitVM does not need an external oracle or an independent PoW header verifier. Proving a Junkcoin deposit on LitVM only requires proving the sibling Merkle branch within a parent Litecoin header that LitVM **already natively knows and trusts**.
2. **Coupled Reorganization Immunity (Fate-Sharing):** The primary existential vulnerability in cross-chain bridges is *asymmetric chain reorganization*—a deep reorg on the source chain invalidating a deposit after tokens have already been minted on the destination chain. When bridging between Junkcoin AuxPoW and LitVM, ***both chains share the exact same Litecoin L1 fork-choice clock***:
 - A reorganization on Litecoin L1 rewinds both the LitVM rollup settlement batches AND the parent blocks committing Junkcoin AuxPoW mining.
 - It is cryptographically impossible for an attacker to reorg Junkcoin's parent commitment without simultaneously reorganizing Litecoin L1 itself, which would rewind LitVM state in lockstep.
 - This fate-sharing eliminates the window of vulnerability that plagues cross-chain bridges on heterogeneous networks.
3. **LitVM as the Unified AuxPoW Liquidity Hub:** Litecoin L1 lacks expressive smart contract capability. LitVM brings high-speed zkEVM execution to the Litecoin ecosystem. By deploying PSOB on LitVM, LitVM becomes the natural **decentralized finance hub for all merge-mined AuxPoW chains**. Junkcoin gains immediate composability with native zkLTC, decentralized exchanges, and lending protocols directly within its home parent ecosystem, while LitVM aggregates deep cross-chain liquidity across all Litecoin AuxPoW sibling chains through a single, provably sound ZK verification gate.

5. Security Analysis and Threat Model

Threat Vector	Phase 1 (FROST MPC)	Phase 2 (PSOB ZK)
Validator Collusion (t -of- n)	Colluding majority (2/3) could construct unauthorized payout. Mitigated by geographic separation & timelock.	Peg-in is immune (soundness of ZK circuit). Peg-out protected by on-chain claim bonds & proof finalization.
Deep Chain Reorganization	Mitigated by $k_{\min} \geq 6$ confirmations before EIP-712 attestation.	Circuit enforces <code>provenWork</code> $\geq W_{\min}$ across parent Script headers (coupled reorg on LitVM).
Double Minting / Replay	On-chain deduplication mapping <code>processedTxids[txid]</code> .	Journal commits <code>depositTxid</code> ; contract enforces single-execution replay gate.
Front-Running Deposit Claims	Mitigated: <code>recipient</code> is bound in <code>OP_RETURN</code> and verified on-chain.	Recipient address is committed in the ZK Journal; proof can only mint to intended address.
Leader Equivocation / Censorship	Leader rotates deterministically: $H(id \parallel round) \bmod 3$.	Decentralized claim board allows any bonded publisher to finalize withdrawals.

Table 1: Comparative Threat Model: Phase 1 FROST MPC vs. Phase 2 PSOB Architecture.

6. Deployment Roadmap and Active Contracts

6.1 Phased Rollout Matrix

Phase	Architecture	Trust Assumptions	Status / Target
Phase 1	FROST 2-of-3 MPC	Honest 2-of-3 validator quorum	Live on Arbitrum One
Phase 1.5	P2P Gossip + rindexer	Distributed monitoring + automated fallback	Live on Testnets (Base, LitVM)
Phase 2	PSOB ZK Peg-In	SP1 cryptographic soundness	Testnet integration (Q4 2026)
Phase 3	Full PSOB Multichain	Trustless peg-in + bonded MuSig2 claim board	Mainnet expansion (2027)

Table 2: Bridge Evolution Roadmap.

6.2 Production Contract Registry

6.2.1 Arbitrum One Mainnet (Chain ID: 42161)

- **Token (wJKC):**
0x98C4725bA9531eD1C2C40A2cEb29a016c014d9e1
- **Bridge Proxy (UUPS):**
0x42E455151c222Afd2A36E2dB94B43c304B3d17f5
- **Bridge Implementation:**
0xd3A85c9e40fBfc29EAd2A3D15d89eB49157829D6
- **Taproot Custody Address:**
jc1py99f6nn90jgu2h8lygc4f7ysxrezd97tedegehnmw967wgypcdqzsfrfcus
- **Taproot Group Pubkey:**
0x969cf5c6a1a9c58f57046dfc49cc7524ff00feb73b170ee2f0f6df7f8fddcb60

6.2.2 Testnet Deployments

- **Base Sepolia (Chain ID: 84532):**
Proxy: 0x22BEc33348037cF4200813A96f1981A70e37A489
Token (wJKC): 0x613352f33E8900246f619B7CBD6C3dC80E919122
- **LitVM LiteForge (zkLTC Rollup):**
Proxy: 0x42E455151c222Afd2A36E2dB94B43c304B3d17f5
Token (wJKC): 0x98C4725bA9531eD1C2C40A2cEb29a016c014d9e1

7. Conclusion

The DeDoo Junkcoin Bridge establishes a robust, mathematically sound, and scalable bridge connecting the Junkcoin AuxPoW ecosystem to EVM networks. By deploying a 2-of-3 FROST MPC Taproot custody architecture on Arbitrum One Mainnet, the system delivers immediate production utility, privacy, and low transaction overhead.

Simultaneously, the integration of the Proof-of-Sibling-Block (PSOB) protocol provides a clear, uncompromising roadmap toward trustless multichain settlement. By unifying Script AuxPoW verification inside an SP1 zero-knowledge guest circuit, PSOB eliminates operator trust for cross-chain deposits and establishes native consensus coupling with LitVM (Litecoin L2 zk-Rollup). This shared Litecoin L1 anchor unlocks coupled reorganization immunity, minimal verification overhead, and transforms LitVM into the premier decentralized liquidity hub for the entire AuxPoW ecosystem.

References

- [1] Rekt News, *Bridge Hacks: A Running Total of Exploit Vectors and Lost Capital*, 2023. <https://rekt.news>
- [2] C. Komlo and I. Goldberg, *FROST: Flexible Round-Optimized Schnorr Threshold Signatures*, Selected Areas in Cryptography (SAC), 2020. <https://eprint.iacr.org/2020/852>
- [3] J. Nick, T. Ruffing, and Y. Seurin, *MuSig2: Simple and Two-Round Schnorr Multi-Signatures*, CRYPTO 2021. <https://eprint.iacr.org/2020/1261>
- [4] P. Wuille, J. Nick, and T. Ruffing, *BIP 340: Schnorr Signatures for secp256k1*, 2020. <https://github.com/bitcoin/bips/blob/master/bip-0340.mediawiki>
- [5] P. Wuille, A. Poelstra, et al., *BIP 341: Taproot: SegWit version 1 spending rules*, 2020. <https://github.com/bitcoin/bips/blob/master/bip-0341.mediawiki>
- [6] senasgr, *Dedoo Bridge: ZK Proof-of-Sibling + Threshold Signing Architecture for the Litecoin AuxPoW Ecosystem*, DeDoo Research Paper, August 2026. <https://dedoo.xyz/research>
- [7] Succinct Labs, *SP1: A Performant 100% Open-Source RISC-V Zero-Knowledge VM*, 2024. <https://succinctlabs.github.io/sp1>
- [8] Q. Xie et al., *zkBridge: Trustless Cross-chain Bridges Made Practical*, ACM Conference on Computer and Communications Security (CCS), 2022.