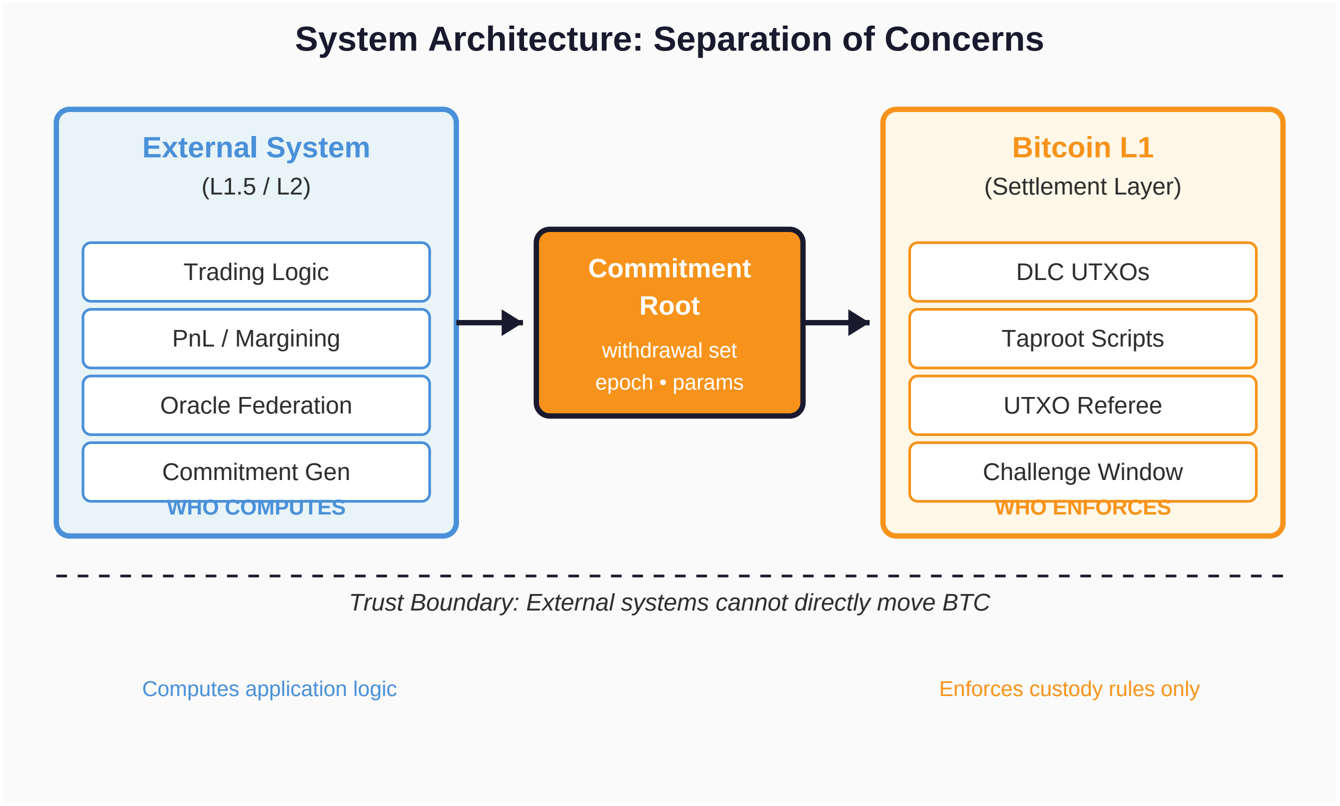
BitVM UTXO Referee

Minimal Settlement Enforcement for Bitcoin-Native Derivatives

Executive Summary

This proposal describes a **minimal BitVM-based enforcement kernel** for verifying UTXO settlement transactions on Bitcoin. Unlike general-purpose computation verification, this project deliberately constrains its scope to a single function: ensuring that sweep transactions follow committed settlement rules.

Bitcoin remains the final arbiter of custody, while external systems (trading engines, oracles, margin calculators) provide signed commitments that are checked against narrowly scoped, auditable rules. The **commitment root** is a Merkle root over (recipient, amount, asset/series, epochId) leaves plus global parameters (cap, expiry, dispute window). This separation keeps circuit complexity low, audit surface minimal, and dispute costs bounded.

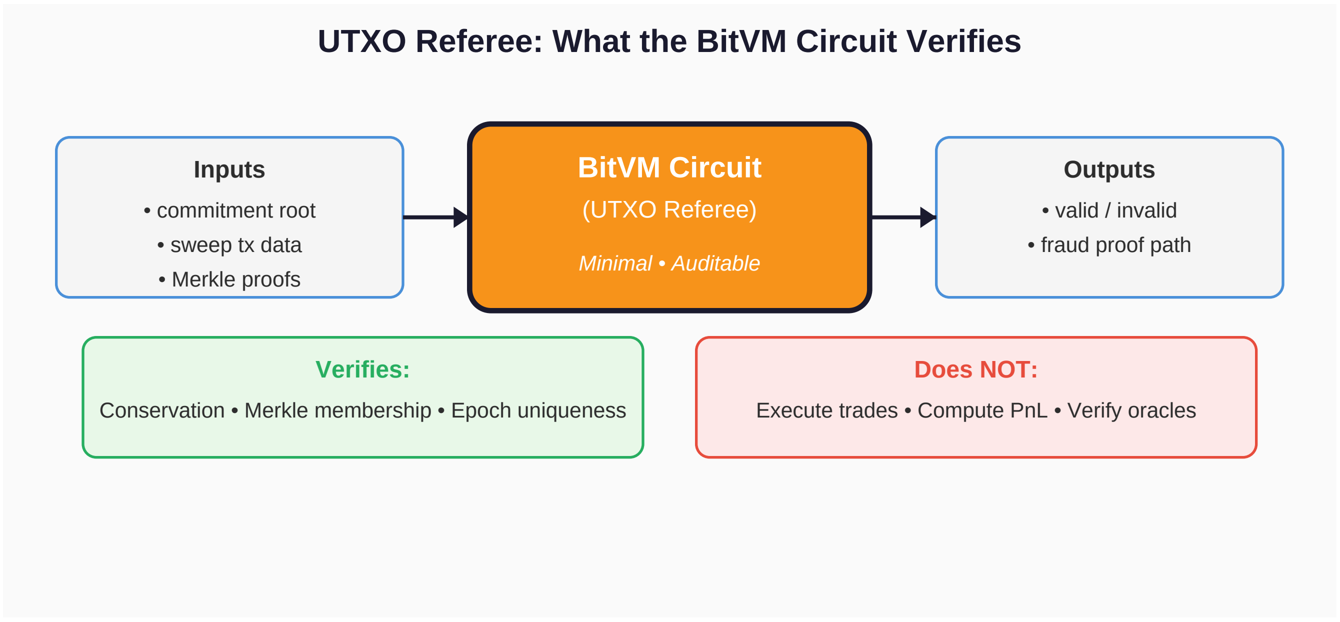


Scope Definition

This Project IS	This Project IS NOT
Minimal UTXO settlement referee	General-purpose BitVM / VM
Fixed-format input verification	Arbitrary computation verifier

Merkle membership + conservation checks	Full RISC-V opcode coverage
Targeted fraud proof for sweeps	Rollup or execution layer

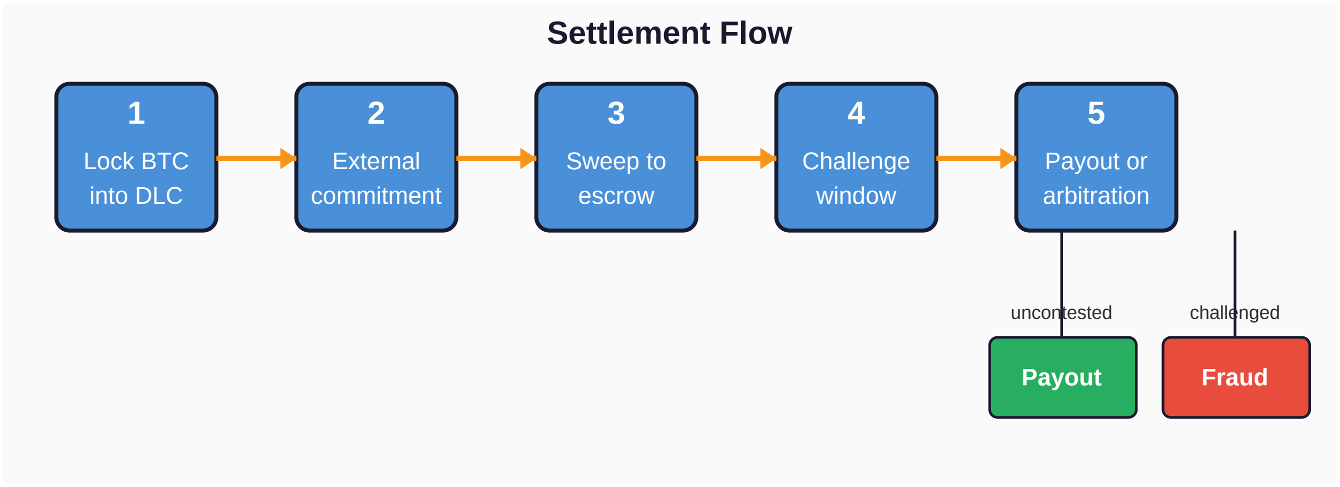
UTXO Referee Circuit



The BitVM circuit verifies a single statement: **"This sweep transaction follows the committed settlement rules."** Specifically:

- **Conservation of funds:** No BTC created or destroyed; outputs ≤ committed cap
- **Withdrawal-set membership:** Each payout corresponds to a leaf in the committed Merkle root
- **Epoch uniqueness:** No double settlement or replay across epochs
- **Residual handling:** Rounding remainders routed to predefined escrow paths

Settlement Flow



Users lock BTC into DLC templates. The external system issues tokenized receipts for collateral use. On a rolling period, the system commits to a settlement root. BTC is swept into escrow UTXOs with a challenge window. If uncontested, funds pay out automatically. If challenged, a BitVM fraud proof redirects funds to the arbitration path—a predefined dispute-resolution UTXO (e.g., bonded adjudicator or re-run settlement under a higher-assurance process).

Technical Approach

- **Commitment anchoring:** Settlement root published on-chain via OP_RETURN or witness commitment
- **Merkle proof generation:** Sparse tree for withdrawal-set membership verification (off-chain prover, on-chain verifiable structure)
- **Timelock dispute window:** Standard CLTV/CSV scripts—challenged settlements route to arbitration before timeout
- **Taproot script paths:** Uncontested payout vs. challenged arbitration branch (no OP_CAT dependency)
- **TypeScript/JavaScript reference implementation:** Accessible toolchain, integration with existing TradeLayer channel infrastructure

Budget & Timeline

Phase	Duration	Amount	Deliverables
Implementation	Months 1-2	$20,000	Commitment infrastructure, Merkle proofs, timelock disputes, testnet
Audit & Hardening	Months 3-6	$13,000	Security review, pentesting, documentation, mainnet prep

Total: $33,000 — Full-time development integrated with TradeLayer deployment activities.

Implementation scope: Reference implementation with commitment infrastructure, Merkle proof generation, and timelock-based dispute windows using standard Taproot scripts. Full BitVM circuit verification (on-chain fraud proof execution) is contingent on covenant opcode activation (OP_CAT or similar); the architecture is designed for seamless upgrade when Bitcoin consensus supports it.

Team

Patrick Dugan, founder of TradeLayer (Bitcoin-native derivatives, since 2014). 10+ years protocol development, production clearing engines.

Contact: patrick.buchanan.dugan@gmail.com • github.com/patrickdugan/tradelayer.js/tree/L2Scaling