We The People

A Bitcoin-Settled Peer-to-Peer Payment Network Using Agent-Centric Architecture, Zero-Knowledge Proofs, and Blinded Batch Settlement

Author: Race Dornan
Contact: RaceDornan@protonmail.com
Date: February 2026
Version: 0.1 (Draft)

Abstract

Current decentralized exchange infrastructure is designed for speculation. Automated market makers, order books, and the ecosystems surrounding them optimize for trader profit extraction, not for ordinary people exchanging value. The result is a financial landscape where bots extract billions through front-running and sandwich attacks, where bridges concentrate hundreds of millions in exploitable honeypots, where oracle manipulation distorts pricing, and where the people who most need accessible financial infrastructure — the unbanked, the underserved, and those in developing economies — are either priced out or exploited.

This paper proposes a different architecture. By combining Holochain's agent-centric distributed computing model, BitcoinOS's zero-knowledge proof verification on Bitcoin, UTXOracle's trustless price feed derived from Bitcoin's own transaction data, and a novel economic model based on over-collateralization and anti-speculative incentive design, we describe a peer-to-peer payment and exchange network that is:

- **Fair by construction** — architecture makes exploitation impossible rather than merely punishable
- **Bitcoin-settled** — inherits the hardest settlement finality available
- **Self-sovereign** — every participant controls their own keys, funds, and identity
- **Anti-speculative** — speculation is economically irrational within the system by design
- **Accessible** — participants can earn entry through mining labor, not just existing capital

The design is ahead of its infrastructure. Several component technologies are still maturing. We publish now so that the teams building these components can see how their work composes into something greater than the sum of its parts.

---

1. Problem Statement

1.1 DeFi Serves Speculators, Not People

The overwhelming majority of decentralized exchange volume is speculative. Automated market makers like Uniswap, Curve, and their derivatives were designed to facilitate trading — the buying and selling of assets for profit. The participant base reflects this:

- Approximately 60% of volume comes from speculators and traders
- Approximately 25% comes from arbitrage bots
- MEV extractors capture billions annually through transaction ordering exploitation
- Actual end users — people exchanging value for payments, remittances, or commerce — represent a small fraction of activity

This isn't a failure of these protocols. They are working as designed. The problem is that their design serves the wrong population for what the world actually needs: a fair, accessible way to exchange value.

1.2 Bridges Are Systemic Vulnerabilities

Cross-chain bridges represent some of the largest losses in crypto history. The Ronin bridge lost $625 million. The Wormhole bridge lost $320 million. These failures share a common root: **concentrated custody with centralized trust assumptions**. Whether through multisig committees, MPC schemes, or lock-and-mint contracts, traditional bridges create honeypots where a single exploit drains everyone's funds.

1.3 Oracles Introduce External Trust Dependencies

DeFi protocols overwhelmingly depend on external oracle networks — Chainlink, Pyth, and others — to source price data. Each oracle introduces trust assumptions about the honesty and reliability of data providers. Oracle manipulation has been a recurring attack vector, and oracle dependencies create failure modes external to the protocols that rely on them.

1.4 Bitcoin's Settlement Strength Is Underutilized

Bitcoin provides the strongest settlement finality in cryptocurrency. Its proof-of-work security, decentralization, and fifteen-year track record are unmatched. Yet Bitcoin's limited scripting language has historically prevented complex financial applications from settling directly on its base layer. Recent advances in zero-knowledge proof verification on Bitcoin — particularly through BitcoinOS and BitSNARK — are changing this, enabling complex computation to be verified on Bitcoin without modifying its consensus rules.

1.5 The Missing Middle

There exists a vast population of potential users who need none of what current DeFi offers and all of what it lacks:

- Remittance senders who need low-cost cross-currency exchange
- Merchants who need to convert between Bitcoin and stable value
- Individuals making peer-to-peer payments across currency boundaries
- Unbanked populations who need financial infrastructure without gatekeepers
- Communities in developing economies with abundant energy resources but limited financial access

These people don't need a trading platform. They need a **utility** — a fair exchange at a fair price with low fees and no one extracting value from them.

---

2. Design Principles

Before any technical detail, we state the principles that govern every architectural decision in this design. Where a technical choice conflicts with a principle, the principle prevails.

Utility over profit. The network exists to serve people who need to exchange value. It is not an investment vehicle, a trading platform, or a speculation engine.

Sovereign agency. Every participant is self-custodial, self-responsible, and in full control of their identity, keys, and funds. No entity — including the network itself — can freeze, seize, or restrict an agent's assets.

Fair by construction. The system does not rely on rules that can be gamed or enforcement that can be evaded. Fairness is a property of the architecture itself — the mathematics, the cryptography, and the economic incentives make unfairness structurally impossible.

Bitcoin finality. All settlement inherits Bitcoin's security model. No weaker chain, no trusted committee, and no external validator set stands between an agent and the finality of their transactions.

Zero external dependencies. The oracle, settlement layer, and verification mechanism all derive from Bitcoin itself. The coordination layer runs on Holochain's peer-to-peer infrastructure. No external service, API, or trusted third party is required for any core function.

Anti-speculative. Speculation within the network is not prohibited — it is made economically irrational. The design removes all profit incentive for speculative behavior, causing speculators to self-select out of the network entirely.

Accessible. Entry to the network must not be gated exclusively by existing capital. Participants must be able to earn their place through labor, ensuring the network serves those who need it most, not only those who already have wealth.

---

3. Technology Foundations

This design composes four technologies that have not previously been connected. Each is summarized here with an honest assessment of its current maturity.

3.1 Holochain

Holochain is an agent-centric distributed computing framework. Unlike blockchains, which maintain a single global ledger validated by global consensus, Holochain gives each agent their own cryptographic source chain. Agents publish data to a distributed hash table (DHT) where neighboring peers validate entries against shared rules called Zomes, written in WebAssembly.

Key properties relevant to this design:

- **No global consensus bottleneck** — scales with the number of agents
- **Agent-centric identity** — each participant maintains their own signed hashchain
- **Peer validation** — integrity enforced locally by DHT neighbors, not by miners or validators
- **Eventual consistency** — no single canonical state, which introduces design challenges for financial applications

Maturity: Beta. Holochain has a functioning runtime and growing developer community but has not yet seen large-scale production deployment of financially critical applications.

3.2 BitcoinOS and BitSNARK

BitcoinOS enables zero-knowledge proof verification directly on Bitcoin's base layer without requiring any changes to Bitcoin's consensus rules. Using the BitSNARK protocol, complex computations can be performed off-chain, compressed into a succinct proof, and verified on Bitcoin. The first ZK proof was verified on Bitcoin mainnet at Block 853626.

Key properties relevant to this design:

- **General-purpose ZKP verification on Bitcoin** — not limited to specific computation types
- **No Bitcoin protocol changes required** — works within existing script capabilities
- **Enables complex settlement logic** — batch settlement, fraud proofs, and recovery mechanisms can all be expressed as ZK circuits

Maturity: Early stage. The core verification mechanism has been demonstrated on mainnet, but tooling, developer experience, and production hardening are ongoing.

3.3 UTXOracle

UTXOracle derives the BTC/USD exchange rate from Bitcoin's own transaction data. The insight is that people tend to transact in round fiat amounts — $100,500$, $1000 — and these round-number transactions create identifiable patterns in UTXO output values. By analyzing the distribution of transaction outputs near each block, the exchange rate can be inferred without any external data source.

Key properties relevant to this design:

- **Zero external trust** — the only assumption is "people use Bitcoin"
- **Publicly verifiable** — any agent can independently compute the price from public blockchain data
- **ZKP-compatible** — price computation can be proven correct in a zero-knowledge circuit
- **No oracle network, no data publishers, no external API**

Maturity: Experimental. The methodology has been demonstrated but is not widely deployed in production systems. Precision and latency are limited by Bitcoin's block time.

3.4 Component Integration

The complete technology stack:

```
┌───────────────────────────────────────┐
│        Bitcoin L1              │
│ Settlement · ZKP verification      │
│ UTXOracle price source           │
│ Agent collateral scripts          │
└─────────────────┬─────────────────────┘
       │
┌─────────────────┴─────────────────────┐
│      BitcoinOS / ZKP Layer        │
│ Batch settlement proofs           │
│ Oracle price computation proofs     │
│ Fraud proofs for slashing          │
│ Recovery verification            │
└─────────────────┬─────────────────────┘
       │
┌─────────────────┴─────────────────────┐
│    Execution Engine             │
│ Phase 1: Blind order book matching  │
│ Phase 2: AMM backstop             │
│ Pro-rata allocation              │
│ Batch computation                │
└─────────────────┬─────────────────────┘
       │
┌─────────────────┴─────────────────────┐
```

```
│     Holochain hApp Layer         │
│ Tiered agent pools               │
│ Blind order submission via DHT   │
│ Agent source chains              │
│ Threshold encryption/decryption  │
│ Validation rules (Zomes)         │
│ Bond-pool liquidity management   │
│ Bootstrap pool coordination      │
└──────────────────────────────────┘
```

---

4. Architecture

4.1 The Agent-as-Bridge Model

Traditional bridges concentrate all users' funds in a single contract or multisig, creating catastrophic single points of failure. This design eliminates bridges entirely by making each agent their own self-sovereign bridge.

To participate in the network, an agent deposits over-collateralized funds into a Bitcoin script that they control. For every unit of trading capacity desired, the agent must deposit two units of Bitcoin:

- **1 unit: Trading collateral** — backs the agent's activity in the payment network
- **1 unit: Integrity bond** — slashable stake that ensures honest behavior

This creates a system where:

- **No honeypot exists.** There is no single contract or address holding pooled funds. Each agent's collateral sits in its own individual Bitcoin script. An attacker would need to compromise agents one by one.
- **Cheating has negative expected value.** If the maximum extractable profit from cheating equals the agent's trading capacity, and the cost of being caught is the integrity bond (equal to trading capacity), the expected value of cheating is zero before accounting for detection probability — which, given ZKP-based fraud proofs, is effectively certain.

- **No trusted third party.** The Bitcoin script enforces spending conditions. The Holochain DHT monitors behavior. ZKPs prove misbehavior. No committee, multisig, or validator set is required.

4.2 Tiered Agent Pools

Agents are separated into pools based on their collateral and trading capacity:

- **Whale Pool** — High collateral agents with large trading capacity
- **Shark Pool** — Medium collateral agents
- **Shrimp Pool** — Lower collateral agents and newly graduated bootstrap participants

This tiered structure serves multiple purposes:

Eliminates cross-class exploitation. In traditional AMMs, a whale and a retail user share the same pool. The whale's single trade can move the price more than all retail activity combined. Tiering ensures that every participant operates in an environment proportional to their size. A shrimp with 0.05 BTC gets the same relative trading experience as a whale with 500 BTC. This is weight classes — flyweights don't fight heavyweights.

Solves the thin pool problem. Pool depth is relative, not absolute. A pool is "thin" when individual trades are large relative to total depth. By restricting each pool to agents of similar size, no single agent can represent a disproportionate share of pool activity. Slippage characteristics become predictable and fair within each tier.

Preserves privacy. Combined with blind orders, tiers limit information leakage. An observer knows an agent's general tier but not their specific collateral amount. Within the tier, all orders are blind.

Tier boundaries are dynamically computed based on the actual distribution of agent collateral, using natural clustering rather than fixed thresholds. The hApp runs this clustering logic, and a ZKP proves the tier assignments were correctly computed.

4.3 Blind Orders With Threshold Encryption

All orders are encrypted before submission to the network. No agent can see any order other than their own until the batch settlement window closes and orders are collectively decrypted.

Order submission: An agent creates an order, encrypts it using a threshold encryption scheme, and generates a zero-knowledge proof attesting that:

- The order is within the agent's collateral limits

- The order follows valid format rules
- The agent's cumulative open orders do not exceed their bonded amount

The encrypted order and the validity proof are published to the DHT. Neighboring peers validate the proof — confirming the order is legitimate — without ever seeing the order contents.

Batch decryption: At the close of each batch window, a quorum of agents cooperates to decrypt all orders simultaneously. No single agent or minority group can decrypt orders early. The threshold is set high relative to pool size to prevent collusion.

Why this matters: Even with batch settlement eliminating sequential ordering attacks, visible orders leak information. Agents could analyze trading patterns, game batch boundaries by observing submitted volume before committing, or build behavioral profiles of other participants. Blind orders eliminate these vectors entirely. The system becomes a sealed-bid batch auction — a well-understood fair mechanism from traditional finance, implemented trustlessly through cryptography.

Order structure: Because all trading is anchored to the UTXOracle price (Section 4.5), orders require no price field. They specify only direction (buy BTC or sell BTC), amount, and a validity proof. This simplification means the order book isn't matching on price — it's matching on direction and amount, functioning more like a clearing house than a traditional order book.

4.4 Two-Phase Execution: Order Book Then AMM Backstop

Each batch settles in two sequential phases:

Phase 1 — Peer-to-Peer Order Matching. After batch decryption, buy orders are matched directly against sell orders at the exact UTXOracle price. No spread, no slippage, no bonding curve involved. When buy and sell flow is balanced — which in a payment network it frequently will be, since one person's need to sell BTC is another's need to buy — this phase handles the entire batch at perfect pricing.

If one side is oversubscribed, allocation is pro-rata. If a batch contains 10 BTC of buy orders and 7 BTC of sell orders, each buyer receives 70% of their order filled in Phase 1 at perfect oracle price, and all sellers are fully filled.

Phase 2 — AMM Backstop. Any volume unmatched in Phase 1 — the net imbalance between buy and sell flow — routes to an AMM backstop pool. This pool absorbs the imbalance at the oracle price plus or minus a small spread, bounded by the floating volatility band (Section 4.6).

The AMM's role is fundamentally different from a traditional AMM. It is not the market. It is a backstop that activates only when peers cannot match each other directly. This means the pool handles a fraction of total volume rather than all of it, keeping it smaller, less stressed, and more sustainable.

The bonding curve for the backstop can be simplified to near-flat pricing within the allowed band, with slippage increasing only as the pool absorbs more imbalanced flow within a single batch. This structure naturally incentivizes balanced flow — if recent batches have been sell-heavy, the slightly higher Phase 2 cost for sellers encourages some to wait for a more balanced batch.

4.5 UTXOracle Price Anchoring

All pricing in the network derives from UTXOracle — the BTC/USD exchange rate computed from Bitcoin's own UTXO transaction patterns. Every agent independently computes the same price from the same public Bitcoin data. No external oracle network, data publisher, or API is consulted.

A ZKP proves that the oracle price was correctly computed from UTXOs in the relevant block range. This proof accompanies each batch settlement, ensuring that all participants can verify pricing was derived honestly.

The network supports BTC/USD-stablecoin pairs exclusively. This constraint keeps the oracle dependency clean — UTXOracle natively provides BTC/USD, and all trading references this single price feed. This eliminates the need for additional oracle sources and keeps the system fully self-referential to Bitcoin.

4.6 Floating Volatility Band (X)

The allowable spread between execution price and oracle price is not fixed. It floats based on recent UTXOracle variance:

- **Low volatility** (BTC price moved less than 1% in the last 6 blocks): X = 0.5%
- **Medium volatility** (1–5% movement): X = 1.5%
- **High volatility** (greater than 5% movement): X = 3.0%

Phase 1 order matching always executes at exact oracle price regardless of X. The floating band governs Phase 2 AMM backstop spread only.

This prevents honest users from being penalized during legitimate market moves while keeping the band tight during stable conditions. The ZKP proving correct batch settlement includes proof that X was correctly computed from UTXOracle data for the relevant block range.

4.7 Anti-Speculation: Profit Confiscation Rule

Any agent whose Phase 2 trades execute outside the ±X% band and result in profit has those profits redirected to the governance pool. The agent retains their principal but gains nothing from off-band execution.

This rule does not prevent bad trades — agents who accept unfavorable pricing bear their own losses. It prevents **extractive** trades — no agent can profit from price dislocation.

Additionally, agents whose trading history over a rolling window shows systematic profit extraction from Phase 2 spread — consistently buying when spread is negative and selling when positive — are identified through ZKP analysis of their source chain history. Demonstrated pattern extraction results in profit confiscation to the governance pool.

The self-selection effect is the real mechanism. This rule doesn't just punish speculators after the fact — it makes the network economically irrational for speculative participation. Speculators, arbitrage bots, and MEV extractors have no incentive to be present. They self-select out entirely. The population that remains consists almost exclusively of people who actually need to exchange value — merchants, payment senders, remittance users, and ordinary economic participants.

This eliminates the adversarial population that every other DeFi protocol spends enormous engineering effort defending against. Sandwich attacks, front-running, and MEV extraction become irrelevant when there's no one attempting them.

4.8 Bond-Pool Passive Liquidity

The AMM backstop requires liquidity. Rather than requiring a separate class of liquidity providers — which would reintroduce speculative incentives — the integrity bonds already posted by every agent serve double duty.

Each agent's 1 BTC integrity bond, while remaining fully slashable, is passively deployed as AMM backstop liquidity during normal operation. The bonds collectively form the backstop pool. This means:

- **No new capital is required** — the liquidity already exists within the system
- **The pool scales automatically** — more agents means more bonds means deeper backstop

- **Security is preserved** — if an agent misbehaves, their bond is slashed first and removed from the pool
- **Fair compensation** — the small cost-recovery fee from Phase 2 fills is distributed to bond holders proportionally, compensating them for providing backstop utility without creating speculative returns

The governance pool (funded by slashing events, confiscated off-band profits, and transaction fees) provides additional backstop liquidity, creating a hybrid pool that grows more resilient over time.

4.9 Transaction Fees

The network charges a small flat fee per transaction — approximately 0.01–0.05% — sufficient to cover:

- Computational cost of ZKP generation
- Ongoing hApp development and maintenance
- Governance pool funding

This is a **cost-recovery fee**, fundamentally different from the 0.3% fees charged by traditional AMMs that exist to compensate speculative liquidity providers. It is the financial equivalent of an ATM charging a small fee to cover operational costs.

4.10 Settlement and Net Clearing

Not every transaction requires on-chain Bitcoin movement. The Holochain layer handles high-frequency matching and state tracking. Bitcoin handles periodic net settlement.

If two agents trade back and forth fifty times during a settlement period, only the net difference settles on Bitcoin. Over-collateralization covers interim risk, but actual on-chain transactions are minimal. This mirrors interbank net settlement — banks don't wire money for every transaction; they net out periodically.

ZKPs prove that all interim activity was conducted honestly, that net balances are correct, and that no agent exceeded their collateral at any point during the settlement period.

---

5. Slashing and Fraud Proofs

5.1 Detection

Holochain's validation architecture provides native misbehavior detection. Every action an agent takes is written to their source chain and validated by DHT neighbors against the Zome rules. If an agent attempts to spend beyond their collateral, forge a trade, double-commit funds, or corrupt their chain, the validation rules reject the entry or Holochain's warranting system detects the inconsistency.

5.2 Proof of Misbehavior

When misbehavior is detected, the evidence is compiled into a witness and used to generate a ZKP proving:

"Agent X committed to state A on Bitcoin but acted according to state B on Holochain"

This fraud proof is submitted to Bitcoin, where BitcoinOS verifies it. Upon verification, the agent's bond script executes its slash path, redirecting the integrity bond to the governance pool.

5.3 Bitcoin Script Spending Paths

Each agent's collateral and bond are held in a Bitcoin script with multiple spending paths:

- **Normal operation:** Agent spends using their key with standard verification
- **Slash path:** Verified fraud proof triggers bond transfer to governance pool
- **Emergency recovery:** Extended timelock allows agent to recover trading collateral minus outstanding obligations
- **Social recovery:** Recovery proof with medium timelock enables key rotation after guardian approval (see Section 7)

---

6. The Bootstrap Pool

6.1 The Access Problem

The over-collateralization model requires agents to deposit twice their desired trading capacity in Bitcoin. This creates an inherent barrier: the people who most need accessible financial infrastructure — unbanked populations, those in developing economies, and anyone whose wealth is in labor rather than financial assets — are excluded by the capital requirement.

A payment utility that serves only those who already have Bitcoin is not truly a public utility.

6.2 Proof-of-Labor Onboarding

The bootstrap pool provides an alternative entry path. An agent with no Bitcoin but access to mining hardware can **earn their way into the network** by contributing hashrate to Bitcoin's security.

Registration: The agent creates a Holochain identity, designates recovery guardians, and receives a bootstrap Bitcoin script address.

Mining phase: The agent directs mining hardware — either individually or as part of a cooperative — to any compatible mining pool, with payouts directed to their bootstrap script address. The hApp tracks contributions by verifying that funds at the bootstrap address originated from mining pool payout transactions traceable to coinbase rewards.

Accumulation: Mined satoshis accumulate in the bootstrap script. During this phase, accumulated funds serve dual purpose — they count toward the agent's graduation threshold while simultaneously being available as AMM backstop liquidity for the network.

Graduation: When the agent's accumulated mining rewards reach the threshold for shrimp pool entry (trading capacity plus bond collateral), a ZKP proves the threshold is met. The bootstrap script transitions to a standard agent collateral/bond script. The agent enters the shrimp pool as a full participant, identical in every way to a capital-entry agent.

6.3 Dynamic Bootstrap Threshold

The graduation threshold responds to network conditions:

- **If AMM backstop is underfunded:** Lower threshold — the network benefits from more bootstrapping agents whose mining simultaneously fills the backstop
- **If network conditions are standard:** Standard threshold
- **If network is at capacity:** Higher threshold — slows entry to maintain pool quality

6.4 Community Cooperative Bootstrapping

Bootstrap agents can form cooperatives. A group of people with shared mining hardware direct their combined hashrate toward shared bootstrap addresses. As individual thresholds are reached, agents graduate one by one into the shrimp pool. Graduated agents can immediately serve as recovery guardians for those still bootstrapping, building social recovery networks that mirror real-world community relationships.

This mirrors the historical formation of credit unions and cooperative banks — communities pooling resources to create financial infrastructure for their members.

6.5 Post-Graduation Mining

Graduated agents who continue mining can direct rewards toward:

- **Personal collateral** — building toward graduation to higher tiers
- **AMM backstop** — earning a proportional share of Phase 2 cost-recovery fees
- **Governance pool** — earning governance weight and funding network development

6.6 Incentive Alignment

The bootstrap pool creates circular reinforcement:

- Mining secures Bitcoin
- Mining rewards fund the AMM backstop
- The AMM backstop enables the payment network
- The payment network serves the agents
- The agents are the miners

Every participant's self-interest aligns with the network's collective interest and with Bitcoin's security model simultaneously.

---

7. Key Management and Social Recovery

7.1 Single Seed Derivation

Despite the system spanning multiple cryptographic domains — Bitcoin (secp256k1), Holochain (Ed25519), and threshold encryption — all keys derive from a single BIP-39 mnemonic. The user manages one seed phrase. All derivation complexity is invisible.

Derivation paths:

- Bitcoin keys: standard BIP-84 derivation for collateral control and settlement signing
- Holochain agent keys: custom derivation path, converted to Ed25519 for source chain signing and DHT identity
- Threshold encryption keys: custom derivation path, rotatable per batch window
- BitcoinOS interaction: uses Bitcoin keys directly, no separate domain required

7.2 Social Recovery via Shamir's Secret Sharing

Upon joining the network, each agent designates guardians — other agents they personally trust. The agent's seed is split using Shamir's Secret Sharing into N shares with a threshold of K required for reconstruction. Each share individually reveals nothing about the seed.

Guardian shares are encrypted to each guardian's Holochain agent key and stored as private entries on both the agent's and guardians' source chains. The DHT provides redundancy through validation receipts.

7.3 Recovery Process

If an agent loses access to their seed:

1. Agent installs the app on a new device, generating a temporary keypair
2. Agent contacts K guardians through any channel — phone, in person, encrypted messaging
3. Each guardian verifies the agent's identity through human means and approves recovery in their app
4. Guardian apps decrypt their shares, re-encrypt to the agent's temporary key, and transmit via DHT
5. Agent's app reconstructs the original seed and re-derives all keys
6. A ZKP proves the recovery was authorized by a valid guardian quorum
7. After a timelock period (allowing time for fraud challenges if the recovery is illegitimate), the agent regains full access to their Bitcoin collateral and network identity

7.4 Guardian Health Monitoring

The hApp periodically verifies guardian availability through encrypted DHT pings. If a guardian becomes inactive, the agent is notified and can designate a replacement, triggering a re-split of the seed that renders the inactive guardian's old share useless.

7.5 User Experience

The complexity of multi-curve derivation, secret sharing, and script-based recovery is entirely hidden. The user experience consists of:

- **Setup:** Write down 24 words and choose trusted guardians
- **Daily use:** Open app, authenticate with biometric or PIN
- **Recovery:** Contact guardians, wait for approvals, wait for timelock

No user is required to understand key derivation, elliptic curves, or zero-knowledge proofs. They manage relationships, not cryptography.

---

8. Economic Model

8.1 Game-Theoretic Security

The over-collateralization model creates a clear economic disincentive for misbehavior. An agent with 1 BTC trading capacity has 1 BTC at stake as a bond. The maximum extractable value from cheating equals the trading capacity. The cost of being caught — with near-certain detection via Holochain validation and ZKP fraud proofs — is the entire bond. Expected value of cheating: negative.

8.2 Governance Pool

The governance pool receives funds from three sources:

- Slashed integrity bonds from misbehaving agents
- Confiscated profits from off-band trading
- Transaction fees from normal network operation

This creates counter-cyclical funding: during volatile or stressed periods, more off-band events occur, growing the pool exactly when insurance reserves are most needed. The governance pool funds:

- AMM backstop supplementation
- Protocol development and audits
- Compensation for edge-case attack victims
- Network upgrades governed by staked agents

8.3 No Impermanent Loss, No LP Extraction

Because every agent provides their own collateral for their own use — rather than passively depositing into a shared pool for others to trade against — the concepts of impermanent loss, LP fee optimization, liquidity mining incentives, and vampire attacks do not apply. There is no separate liquidity provider class to compensate or exploit.

8.4 Reputation-Based Collateral Tiers

Over time, agents with clean trading histories — verifiable through their Holochain source chains and provable via ZKPs — may qualify for reduced collateral ratios:

- New agent (no history): 200% collateral
- Agent with 6 months clean history: 175% collateral
- Agent with 1 year and high volume: 150% collateral
- Governance-approved veteran: 125% collateral

This rewards long-term honest participation without compromising security for new or unknown participants.

---

9. Security Analysis

9.1 Attacks That This Design Eliminates

Sandwich attacks and front-running. Blind orders prevent anyone from seeing pending transactions. Batch settlement eliminates sequential ordering. There is no mempool to observe and no "front" to run. Combined with the anti-speculation rule that makes extraction unprofitable, the entire MEV attack class becomes irrelevant.

Bridge exploits. No bridge exists. Each agent's collateral sits in its own Bitcoin script. There is no pooled custody to exploit.

Oracle manipulation. UTXOracle derives from Bitcoin's own transaction data. Manipulating it would require manipulating Bitcoin itself — a cost measured in billions of dollars of hashrate.

Price manipulation via AMM. The bonding curve's convex slippage makes brute-force price manipulation costly on any AMM. In this design, the profit confiscation rule additionally ensures that even successful manipulation yields no benefit.

9.2 Attacks That Are Mitigated

Sybil attacks across tiers. A whale could create many shrimp-tier agents to dominate a smaller pool. The 200% collateral requirement makes this expensive (capital is locked, not just spent), and hApp validation rules can incorporate social vouching or activity-pattern analysis to detect synthetic identities.

Threshold decryption collusion. The quorum that decrypts blind orders at batch settlement could collude to decrypt early. Mitigation: set the threshold high relative to pool size, use verifiable delay functions, and rotate quorum membership.

Liveness attacks. An agent could go offline and refuse to settle. The timelock paths in Bitcoin scripts ensure funds are always recoverable, and the bond compensates counterparties for delays.

Timing attacks on batch boundaries. Even with blind orders, the timing of submission is observable. Mitigation: randomized batch close times or mandatory early commitment windows with random delay jitter on order propagation.

Thin pool exploitation. Tiered pools ensure that pool depth is always proportional to maximum individual trade size within each tier, making relative exploitation impractical regardless of absolute pool size.

9.3 Remaining Open Risks

Cross-tier rebalancing complexity. If tiers develop persistent price discrepancies despite UTXOracle anchoring, the cross-tier rebalancing mechanism must be carefully designed to prevent gaming.

UTXOracle latency. Bitcoin's 10-minute block time creates inherent price staleness. During rapid price movements, the oracle may lag significantly behind true market price. The floating X band partially addresses this, but extreme volatility events could still create challenges.

Regulatory risk. Depending on jurisdiction, the network's exchange functionality could be classified as money transmission, requiring licensing or compliance measures that conflict with the decentralized architecture. This risk is external to the design but relevant to deployment.

---

10. Technology Readiness

This design is honest about its dependencies on maturing technologies:

Component	Current Status	Required For	Risk Level
Bitcoin L1	Production (15+ years)	Settlement, oracle data	None
UTXOracle	Experimental	Price feed	Medium
BitcoinOS/BitSNARK	Early stage	ZKP verification on Bitcoin	High

Component	Current Status	Required For	Risk Level
Holochain	Beta/Maturing	Agent coordination, DHT	Medium
Threshold encryption	Well-understood cryptography	Blind orders	Low
Shamir's Secret Sharing	Well-understood cryptography	Social recovery	Low
ZK circuit design for WASM	Active research	Proving hApp computation	High
WASM to ZK compilation	Early research	End-to-end verification	High

The highest-risk dependencies are BitcoinOS's maturation for production ZKP verification on Bitcoin and the compilation pipeline from Holochain's WASM-based Zomes into zero-knowledge circuits. Both are areas of active development with multiple teams contributing.

This design is ahead of its infrastructure. We present it now so that teams building these individual components can see how they compose into something greater than the sum of parts, and so that infrastructure development can be informed by compelling end-to-end use cases.

---

11. Open Questions

The following questions require further research, formal analysis, or implementation experience to resolve:

Batch window duration. What is the optimal batch window length? Shorter windows improve responsiveness but reduce the probability of balanced order flow. Longer windows improve matching but introduce latency. The optimal point likely depends on network size and activity patterns and may differ by tier.

Threshold encryption quorum sizing. How large must the decryption quorum be relative to pool size to make collusion impractical? What is the performance cost of larger quorums?

Floating X parameter calibration. The volatility thresholds and corresponding X values proposed in this paper (0.5%, 1.5%, 3.0%) are illustrative. Formal analysis using historical UTXOracle data and simulation is needed to determine optimal parameters.

Cross-tier rebalancing. If virtual liquidity transfers between tiers are needed to maintain price consistency, what mechanism prevents gaming of the rebalancing process?

Bootstrap mining verification. The system must verify that bootstrap funds originated from mining rather than direct deposits. The proposed coinbase-tracing approach needs formal specification and testing against edge cases.

Governance structure. How are governance pool decisions made? What voting weight do agents receive? How are protocol upgrades proposed, debated, and implemented?

WASM to ZK circuit compilation. Can Holochain Zome validation logic be feasibly expressed in ZK circuits for end-to-end verification? What are the performance characteristics?

Regulatory navigation. How can the network operate in jurisdictions with money transmission laws? Is the decentralized architecture sufficient to avoid classification as a money services business, or are compliance mechanisms needed?

Agent onboarding UX. The over-collateralization requirement and confirmation waiting time create onboarding friction. What guided experiences, progressive trust models, or social onboarding mechanisms can reduce this without compromising security?

Emergency scenarios. What happens during a sustained Bitcoin network congestion event that prevents timely settlement? What happens if UTXOracle produces anomalous readings due to unusual transaction patterns?

---

12. Implementation Covenant

This design describes a public utility. Its value derives from being a commons — owned by no one, available to all, extractive to none. This principle extends to implementation.

Any implementation of this design should be released under a strong copyleft license such as the GNU Affero General Public License (AGPL-3.0) or the Cryptographic Autonomy License (CAL), ensuring that:

- All source code remains publicly available
- Network-deployed instances cannot hide proprietary modifications

- Users retain sovereignty over their data and their participation
- No entity can create a proprietary fork that captures the network's value

An implementation that violates these principles may be technically possible but is philosophically incompatible with the design's purpose. A proprietary implementation of a public utility is a contradiction.

We release this design under Creative Commons Attribution-ShareAlike 4.0 International (CC-BY-SA 4.0) and recommend AGPL-3.0 or CAL for all implementation code. We further recommend that any reference implementation repository explicitly state these licensing expectations in its README and CONTRIBUTING files.

---

13. Call to Collaboration

This design requires expertise spanning Bitcoin script programming, zero-knowledge proof systems, Holochain hApp development, mechanism design, distributed systems engineering, cryptographic protocol design, and user experience research. No single team is likely to possess all of these capabilities.

We publish this as an invitation to the builders working on these individual technologies to consider how their work could compose into a system that serves ordinary people rather than extracting from them.

Specifically, we invite engagement from:

- **The Holochain team and developer community** — this design is built on Holochain's philosophical and technical foundations and represents a concrete use case for agent-centric architecture in financial infrastructure
- **The BitcoinOS team** — this design demonstrates a compelling application for ZKP verification on Bitcoin beyond speculative DeFi
- **The UTXOracle community** — this design places UTXOracle at the center of a complete financial system, validating the trustless oracle concept at scale
- **Bitcoin developers and researchers** — this design settles entirely on Bitcoin and strengthens Bitcoin's network through bootstrap mining
- **Mechanism designers and economists** — the anti-speculative incentive model and tiered pool structure merit formal game-theoretic analysis

- **Open source and public goods advocates** — this design is explicitly structured as a commons and needs community stewardship

The problems this design addresses — extractive financial infrastructure, inaccessible exchange mechanisms, and the capture of decentralized systems by speculative interests — are not solved by technology alone. They require a community committed to building financial infrastructure that serves people rather than extracting from them.

We believe that community exists. This paper is our invitation to find each other.

Contact: Race Dornan — RaceDornan@protonmail.com

Citation: Dornan, R. (2026). *We The People: A Bitcoin-Settled Peer-to-Peer Payment Network Using Agent-Centric Architecture, Zero-Knowledge Proofs, and Blinded Batch Settlement*. Version 0.1.