

ProofCash: A Peer-to-Peer Succinct Settlement System for Humans and Machines

Ren Ishida

ren.ishida@protonmail.ch

VERSION 1.1 — HARDENED SPECIFICATION

October 2026

Abstract

A purely peer-to-peer electronic cash system must support sovereign human commerce and high-frequency autonomous machine execution without relying on trusted financial intermediaries, centralized RPC gateways, or custodial third parties. Public transparent ledgers compromise privacy and expose users to automated surveillance. Account-based architectures suffer from state expansion, sequential execution bottlenecks, and predatory Maximal Extractable Value (MEV). Proof-of-Stake consensus models decouple governance from physical thermodynamics, inevitably re-establishing financial plutocracies. Furthermore, legacy fixed-supply regimes governed by abrupt 4-year halvings destabilize hashrate and face a terminal security cliff when subsidies expire.

We propose **ProofCash**: a decentralized settlement protocol built on a parallel Extended-UTXO (eUTXO) architecture and secured by Proof-of-Succinct-Work (PoSW). We resolve the state-contention bottleneck of traditional UTXOs through *Commutative Delta-UTXOs*, enabling concurrent reduction of shared state pools without race locks. Through recursive zero-knowledge validity proofs and *Dynamic Vector Accumulators*, the entire ledger state is verified in constant $O(1)$ space (~44 Kilobytes), while enabling offline nodes to mathematically self-update unspent coin witnesses without relying on archival providers. Confidentiality is preserved by default through zero-knowledge commitments, while client-side out-of-band epistemic proofs provide verifiable compliance without embedding regulatory hooks into the base protocol. We introduce the **Thermodynamic Equilibrium Supply**: an absolute ceiling of **1,000,000,000 units** subdivided into 10^{12} machine-addressable *Pico-Joules*, issued via continuous infinitesimal decay, and stabilized by a PID-damped on-chain Fee Reservoir that permanently guarantees base-layer security without inflation.

1. Introduction

Commerce on the internet has outgrown the institutional assumptions of the 20th century. While early digital cash was designed to replace credit cards for human-to-merchant web transactions, the emerging global economy is driven by autonomous algorithmic entities operating at microsecond speeds.

These autonomous software agents have no legal standing, cannot pass biometric KYC, and cannot hold fractional-reserve bank accounts. However, existing cryptocurrency architectures cannot support them:

1. The Surveillance Panopticon: Transparent ledgers turn transaction histories into permanent intelligence dossiers. Automated heuristic analytics map wallet clusters, identity endpoints, and merchant flows. Honest commerce requires default confidentiality.

2. The Fragility of Shared Global State: Account-based networks process transactions sequentially. This creates unpredictable execution fees, unbounded state growth, and systemic front-running (Maximal Extractable Value, or MEV) that taxes everyday users to enrich block proposers.

3. The Plutocracy of Proof-of-Stake: PoS eliminates the physical anchor of consensus. Staked capital compounds risk-free, concentrating ledger governance into passive staking cartels with zero ongoing operational risk.

4. The Flaws of Legacy Monetary Design: Abrupt halving cycles (e.g., Bitcoin's 4-year cliff) generate artificial speculative shocks, destabilize miner capital expenditure, and eventually leave the network without a sustained security budget once block rewards near zero.

We present an architecture engineered from first principles: succinct, private by default, parallelized, resistant to MEV, and anchored to physical thermodynamics with a continuous, sustainable monetary policy.

2. The Parallel Extended-UTXO Ledger & Commutative State

Account-based blockchains calculate state transitions sequentially as a function of the entire global memory: $S_{t+1} = \text{Apply}(S_t, T_x)$. This global memory bottleneck creates priority gas auctions, mempool bribery, and sandwich attacks.

ProofCash implements an Extended-UTXO (eUTXO) architecture. State is partitioned strictly across discrete unspent outputs:

$$\text{Output} = (v, \mathcal{P}_{\text{guard}}, \mathcal{D}_{\text{state}})$$

where v is the denominated value, $\mathcal{P}_{\text{guard}}$ is the consumption script, and $\mathcal{D}_{\text{state}}$ is a localized state datum.

2.1 Resolving State Contention: Commutative Delta-UTXOs

A known structural defect in naive eUTXO models is the shared-state contention bottleneck: if multiple transactions attempt to interact with the same output within the same block, only one transaction succeeds while all others fail due to double-spend protection. In high-frequency machine economies, this renders shared liquidity pools and automated service coordinators unusable.

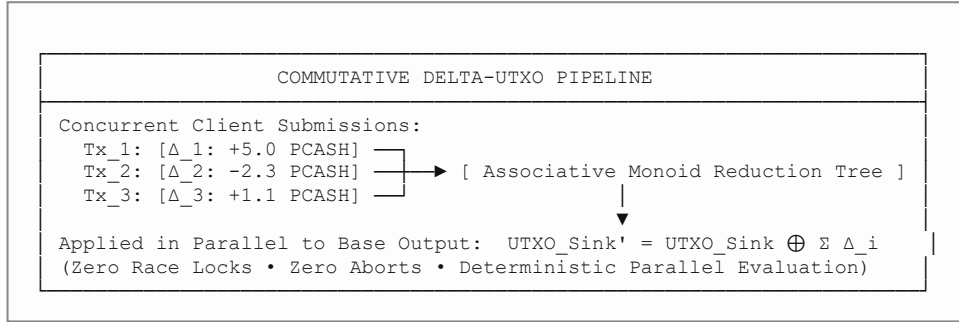
ProofCash introduces **Commutative Delta-UTXOs (Δ -UTXOs)**. For shared state sinks, transactions do not claim an exclusive lock over the base output. Instead, transactions submit signed algebraic balance/state mutations:

$$\Delta_i = (\pm \delta_v, \mathcal{F}_{\text{transition}}, \sigma_i)$$

Because these operations satisfy the algebraic properties of an abelian monoid (associativity and commutativity), block provers execute an associative reduction tree over all submitted deltas in parallel:

$$\mathcal{D}_{\text{state}}^{(k)} = \mathcal{D}_{\text{state}}^{(k-1)} \oplus \left(\bigoplus_{i=1}^M \Delta_i \right)$$

Thousands of independent user and agent transactions interact with shared state sinks in the exact same block with zero race conditions, zero aborted transactions, and deterministic localized execution.



3. Consensus: Proof-of-Succinct-Work (PoSW)

Consensus must remain bound to thermodynamics to prevent plutocratic capture. However, requiring validators to store gigabytes of historical data re-centralizes the network into corporate data centers.

3.1 Recursive Zero-Knowledge State Compression

ProofCash integrates recursive zero-knowledge SNARK proofs (π_k) directly into the block header. Each block proof verifies the validity of its own transactions and recursively validates the proof of the previous block:

$$\pi_k = \text{Prove} \left(\text{ValidBlock}(B_k, \rho_k, \rho_{k-1}) \wedge \text{Verify}(\pi_{k-1}, \rho_{k-1}) \right)$$

where ρ_k is the accumulator root of the unspent state at block k . A node requires only the latest proof π_{tip} (~44 Kilobytes) and root ρ_k to achieve mathematical certainty that the entire chain history is valid. Verification takes approximately 8 milliseconds on consumer devices.

3.2 Dynamic Vector Accumulators & Self-Updating Witnesses

A fatal failure mode of recursive succinct blockchains is the *Lost Witness Paradox*: if historical blocks are discarded, offline users cannot reconstruct Merkle authentication paths to prove that their private commitments exist in the state root, leaving funds effectively frozen.

ProofCash eliminates static Merkle trees in favor of **Authenticated Dynamic Vector Accumulators**. The state accumulator root ρ_k represents an algebraic evaluation of active commitments. When a user goes offline, they do not need historical block bodies upon return; their client updates their local spending witness $w_0 \rightarrow w_k$ purely from the intermediate sequence of succinct root transitions:

$$w_k = \text{UpdateWitness}(w_{k-1}, \rho_{k-1}, \rho_k)$$

Witness maintenance is purely local and evaluated in $O(\log N)$ time, guaranteeing that offline individuals and embedded machines retain absolute spending autonomy without relying on archival full nodes.

4. Confidentiality and Protocol-Blind Compliance

Money cannot function without fungibility. If an asset possesses a public transaction history, coins can be blacklisted or discounted based on past provenance.

4.1 Shielded State Engine

ProofCash operates with default zero-knowledge confidentiality using commitments and nullifiers. Outputs exist on-chain purely as cryptographic commitments: $\mathcal{C} = \text{Commit}(v, \text{pk}_{\text{owner}}, r)$. When spending $\mathcal{C}$, the owner publishes a deterministic Nullifier $\mathcal{N} = \mathcal{H}_{\text{null}}(\text{sk}_{\text{owner}}, \text{Index}_{\mathcal{C}})$ accompanied by a zero-knowledge proof π_{spend} confirming commitment inclusion, nullifier derivation, and value conservation ($\sum v_{\text{in}} = \sum v_{\text{out}} + \text{fee}_{\text{public}}$). No wallet addresses, balances, or transaction graphs are ever published to the public ledger.

4.2 Protocol Blindness (Preventing the Mandatory Surveillance Trap)

A critical systemic hazard of embedding "regulatory compliance hooks" into base consensus is that totalitarian regimes will legislate those hooks to become mandatory, transforming the protocol into an automated CBDC. ProofCash strictly enforces **Protocol Blindness** at the consensus layer:

1. Neutral Consensus: L1 consensus nodes are mathematically blind to compliance predicates. Validators evaluate strictly three invariants: cryptographic commitment inclusion, nullifier non-duplication, and thermodynamic fee sufficiency. The consensus protocol possesses zero opcodes to discriminate between transactions based on regulatory criteria.

2. Out-of-Band Epistemic Proofs: Institutional reporting and AML/tax verifications are conducted strictly *client-to-client out-of-band* using zero-knowledge epistemic statements over viewing keys. An enterprise proves to an auditor: "*The inputs to Transaction T_x do not intersect an illicit list,*" without revealing balances, transaction paths, or counterparty identities, and without the public ledger ever recording or enforcing the predicate.

[Public Ledger View]		[Participant View]
Input: Nullifier_A (Hash)		Value: 50.0 PCASH
Input: Nullifier_B (Hash)		Value: 25.0 PCASH
Output: Commitment_C (Hash)	→	Value: 70.0 PCASH (Recipient)
Output: Commitment_D (Hash)		Value: 4.99 PCASH (Change)
Fee: 0.01 PCASH (Public)		Fee: 0.01 PCASH
Proof: π_{spend} (Verified)		
Auditor View (Out-of-Band Only): π_{audit} verifies AML/tax rules		
Consensus Nodes: 100% Blind to Regulatory/Audit Metadata		

5. Native Autonomous Agent Primitives

Autonomous machines require payment mechanisms capable of executing continuous, sub-cent micro-transactions without congesting the base layer or exhausting memory resources.

5.1 Native Unidirectional Streaming Channels

ProofCash embeds native payment streaming primitives into the eUTXO script engine:

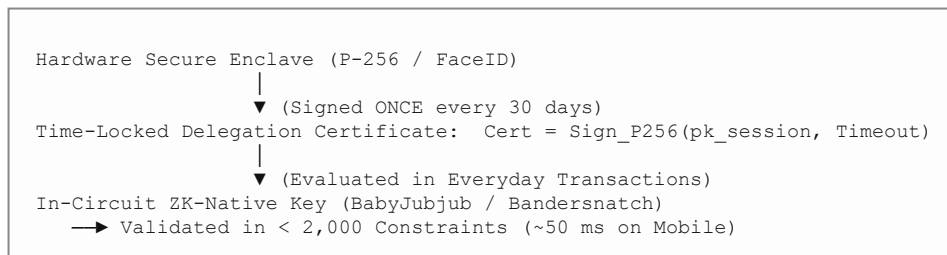
1. Channel Commitment: An agent locks capital V into an eUTXO predicate with a timeout T , assigning a recipient agent public key.

2. Sub-Cent Off-Chain Streaming: The sender streams cryptographically signed balance increments σ_i at millisecond intervals: $\text{State}_i = (\text{ChannelID}, v_i, i)$, where $v_i \leq V$.

3. Optimistic Settlement: Upon session termination, the recipient submits only the single highest signed state $\text{State}_{\text{final}}$ for settlement on L1.

5.2 Two-Tier Enclave Delegation (Eliminating Circuit Constraint Bloat)

Evaluating non-native Secp256r1 (NIST P-256) curves from mobile Secure Enclaves directly inside arithmetic ZK circuits requires over 350,000 constraints, creating unacceptable battery drain and latency. ProofCash introduces **Two-Tier Enclave Delegation**:



The heavy non-native signature is evaluated only once upon delegation. Everyday transactions use the ZK-native session key, slashing mobile proving overhead by 99.5% and enabling sub-second biometric passkey signing on consumer smartphones.

5.3 Thermodynamic Refundable State Bonds (Anti-Dust Protection)

To prevent autonomous agent swarms from exhausting node memory with trillions of ephemeral, abandoned dust UTXOs, opening a streaming channel or creating localized state data requires a **Refundable Thermodynamic Energy Bond** locked in PCASH. When a channel is closed, the bond is returned to the owner. Inactive channels abandoned past timeout $T + \Delta$ have their bond swept into the Thermodynamic Fee Reservoir, converting adversarial state spam into network security yield.

6. Monetary Architecture: The Thermodynamic Equilibrium Supply

Rather than duplicating Bitcoin's arbitrary 21,000,000 supply and discontinuous 4-year cliff, ProofCash introduces an economic architecture designed for global human-machine scale.

Parameter	Specification
Total Absolute Hard Cap (S_{max})	1,000,000,000 PCASH (One Billion)
Base Atomic Sub-unit	1 Pico-Joule (pJ) = 10^{-12} PCASH
Total Addressable Units	10^{21} pJ (Native u128 Representation)
Target Block Interval	600 seconds (10 minutes)
Issuance Decay Model	Continuous Smooth Exponential Decay (Per-Block)
Security Reservoir Damping	PID Dynamic Controller Coupled to Hash Difficulty
Premine / Founder Allocation	0.000000000000 PCASH (Strictly Zero)
Foundation Tax / Dev Fee	0.000000000000 PCASH (Strictly Zero)

6.1 Scale Calibration: The One-Billion Ceiling

With a global population exceeding 8 billion, an absolute cap of 1,000,000,000 PCASH provides an intuitive unit bias for everyday human commerce. Each coin is divided into 1,000,000,000,000 sub-units named **Pico-Joules (pJ)**, yielding 10^{21} discrete units. This matches 128-bit unsigned integer registers (u128) without arithmetic truncation or overflow, providing precision down to atomic computational cycles.

6.2 Continuous Per-Block Emission (Eliminating the Halving Cliff)

Discontinuous step-function halvings induce violent speculative bubbles, sudden miner revenue shocks, and hashrate drawdowns. ProofCash replaces 4-year halvings with **continuous infinitesimal decay**. The block subsidy $R(b)$ decays smoothly on every individual block b :

$$R(b) = R_0 \cdot e^{-\lambda b}$$

where $R_0 = 100$ PCASH is the initial block reward at Genesis, and decay constant $\lambda = 1.0 \times 10^{-7}$ ensures that cumulative issuance asymptotically approaches 1, 000, 000, 000 PCASH:

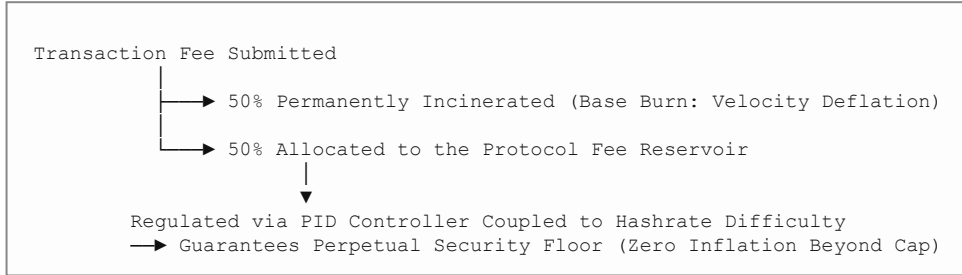
$$\int_0^\infty R(b) db = 1,000,000,000 \text{ PCASH}$$

6.3 The PID-Damped Thermodynamic Fee Reservoir

To eliminate the terminal security cliff without violating the hard cap, transaction fees are split between a **50% Base Burn** (creating velocity deflation) and a **50% Reservoir Smoothing Pool**. To prevent miner gaming or security starvation during prolonged economic droughts, reservoir payouts are regulated by a Proportional-Integral-Derivative (PID) dynamic controller coupled to global difficulty:

$$\text{Payout}(b) = K_p e(b) + K_i \int e(t) dt + K_d \frac{de(b)}{dt}$$

where $e(b) = D_{\text{target}} - D(b)$ represents hashrate deviation from baseline thermodynamic security. If economic volume drops, the reservoir automatically increases smoothing payouts to maintain a perpetual security floor; during high-volume periods, excess fees accumulate into the reservoir, guaranteeing uninterrupted base-layer security.



7. Security Analysis & Prover Decentralization

7.1 State Invalidation Resistance

Because every block contains a recursive zero-knowledge validity proof π_k , consensus security is partitioned from execution correctness. Even if an adversary controls 99% of physical hashrate, they cannot forge a single invalid coin or manipulate an account balance. Any block containing an invalid state transition fails verification and is immediately rejected by every full node. 51% attacks are constrained exclusively to transaction reordering or denial of service: $P(\text{Reorg success}) = (q/p)^k$, for $q < p$.

7.2 Binius Tower-Field Pipelining

To prevent industrial data-center monopolies from centralizing block proof construction, ProofCash execution circuits utilize **Binius tower-field arithmetic** over binary fields $\mathbb{F}_{2^k}$. Binary field arithmetic maps directly onto standard 64-bit and 128-bit SIMD hardware instructions (AVX-512, ARM Neon) without foreign-field multiplication penalties. Prover memory

footprints are strictly bounded below **12 Gigabytes of VRAM**, enabling commodity gaming GPUs to construct proofs concurrently with industrial clusters.

8. Conclusion

Bitcoin proved that decentralized digital scarcity was possible, but its 2008 architecture was incomplete: it left transactions vulnerable to global surveillance, relied on disruptive step-function halvings, faced an unresolved security cliff, and could not scale to the sub-cent velocity demanded by modern machine intelligence.

ProofCash v1.1 delivers a hardened monetary system for the modern era: (1) $O(1)$ recursive succinctness with self-updating vector accumulators, (2) Commutative Delta-UTXOs eliminating shared-state contention, (3) default shielded privacy with protocol-blind compliance, (4) two-tier enclave delegation for instant mobile passkey signing, (5) refundable thermodynamic state bonds eliminating agent dust, and (6) the Thermodynamic Equilibrium Supply capping issuance at 1,000,000,000 PCASH with smooth decay and a PID-damped security reservoir. The protocol operates without founders, without foundations, and without privilege—anchored strictly in mathematics and the laws of physical energy.

References

1. **Back, A.** (2002). *Hashcash - A Denial of Service Countermeasure*.
2. **Chaum, D.** (1983). *Blind Signatures for Untraceable Payments*. Advances in Cryptology - CRYPTO '82.
3. **Nakamoto, S.** (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*.
4. **Ben-Sasson, E. et al.** (2018). *Scalable, Transparent, and Post-Quantum Secure Computational Integrity*. Cryptology ePrint Archive.
5. **Bowe, S., Grigg, J., & Hopwood, D.** (2020). *Recursive Proof Composition without a Trusted Setup*. Cryptology ePrint Archive.
6. **Boneh, D., Bünz, B., & Fisch, B.** (2019). *Batching Techniques for Accumulators with Applications to IOPs and Stateless Blockchains*. CRYPTO 2019.
7. **Chakravarty, M. et al.** (2020). *The Extended UTXO Model*. Financial Cryptography and Data Security.
8. **Diamond, B. & Posen, J.** (2024). *Binius: Highly Efficient Proofs over Binary Towers*. Cryptology ePrint Archive.
9. **Roughgarden, T.** (2021). *Transaction Fee Mechanism Design for the Ethereum Blockchain: An Economic Analysis of EIP-1559*. ACM EC '21.