

Social Commodity Layer

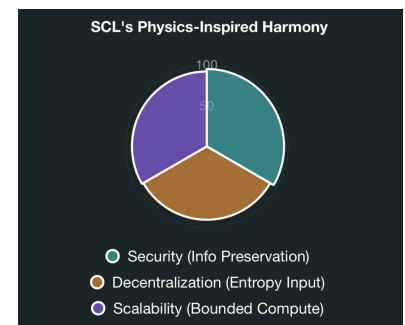
A Peer-to-Peer Quantum-Resistant Economic Engine

Abstract. Breakthroughs in quantum-computing have sparked recent debates over the glaring vulnerabilities imposed by classical cryptography. The potential impact on critical infrastructure, including, but not limited to, public sectors, private industries and national security interests, is an existential threat to its cryptographic foundation. Post-quantum (PQ) academic research aims to establish safeguards against such weaponization, though complex mathematical primitives often hinder applicable standards. With respect to cryptocurrencies, the blockchain *trilemma* quickly evolves into a far greater obstacle. As scalability decreases, these challenges multiply and exceed the domain of global network topology. Furthermore, while the double-spending problem has been solved, there are few incentives, if any, to spend digital assets as a competitive medium of exchange in accordance with the original ethos that defined early cash systems. The only conclusion to the widely-accepted “Digital Gold” narrative, which tempers organic adoption at scale, is a proof-of-concept with the power to influence a modern-day paradigm shift: a circular economy that champions access to a self-sustainable, finite, yet easily accessible payment vehicle.

1. Introduction

The blockchain *trilemma* [1], famously articulated by Vitalik Buterin in 2017, is not merely a computational constraint; it is a multidisciplinary representation of the “Spending Problem.” It posits that Security, Decentralization, and Scalability are effectively trade-offs that correlate respectively with three quintessential functions of virtual currency, defined by the Commodity Futures Trading Commission [2]: a Store of Value, a Medium of Exchange, and a Unit of Account. A virtual currency is only a true Store of Value when secure; a trusted Medium of Exchange when decentralized; and a preferred Unit of Account when scalable. To resolve the friction between these competing properties, the protocol introduces **Proof-of-Symmetry (PoS²)**, a consensus mechanism that integrates game-theoretic principles, quantum-derived randomness and the security assumptions of lattice-based hard problems.

Unlike algorithms that rely on external energy (Proof-of-Work) or static capital (Proof-of-Stake), PoS² is a dual-tier application featuring a Noether-inspired economic conservation model [3] that establishes transactional velocity as the primary mechanism for conserving systemic network capital. In this system, the synergy between market participants—whether human individuals, corporate entities, or autonomous AI agents—becomes the prerequisite for network security. By coupling consensus authority to economic velocity, the protocol incentivizes the circulation of capital, aligning personal profit motives with the collective health of the network.



6th century Greek classic “The Tortoise and the Hare” serves as the defining heuristic for this economic model:

The Hare (Staking): Represents "Liquid Authority." It offers leveraged validation weight, but at the cost of immediate liquidity (temporary inability to spend). Like the Hare, it is fast but volatile; if the stake is withdrawn, the authority vanishes.

The Tortoise (Spending): Represents "Compound Legacy." It sacrifices rapid accumulation to build a reserve of premium time credits. Like the Tortoise, it is slow and relentless; a user who consistently participates in the economy creates a substrate of authority that persists over time even if their balance fluctuates.

This enforces mathematical symmetry—including a transparent one-to-one (1:1) ratio between a node's validation rate and staked balance, compounded by their cumulative fee history. It gives rise to a self-correcting, free market where asset management is no longer restricted to hoarding value, but rather, optimizing the flow of liquidity. Spending becomes the “fuel” for the economic engine, ensuring that the network is secured not just by those who merely hold, but predominantly by those who proactively transact. In fact, the consensus mechanism grants the means of production to all of its validators, in return for efficient contributions towards its security budget.

2. Consensus

The security of PoS² stems from the design of the Game Audit mechanism, which continuously records the consumption of Validations in the form of Valuations. Valuations are individual vectors that contain, at most, ten (10) integers whose values are range bound from zero to ninety-nine (0 - 99). The game audit consists of a ten-by-ten (10 x 10) matrix that stores up to the last ten (10) confirmed valuations—sorted by timestamp—inside a circular buffer. Each valuation is explicitly linked to a hash of the corresponding user’s public key in addition to a timestamp denoting when it was added to the audit.

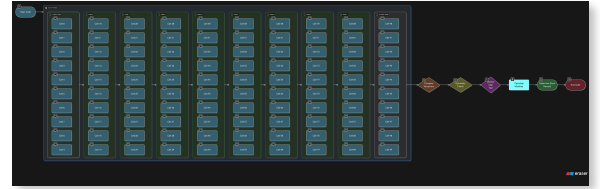
New valuations are always inserted into the game audit at the highest available index; if the audit is full, the default index is nine (9). Each previous row, from index eight (8) to the oldest valuation at index zero (0), is “overwritten” by the preceding valuation, extending the logical queue and thus removing the oldest valuation from the matrix. The latest valuation is then compared to all remaining valuations to determine if any of the audit’s columns, formed by concatenating data from each valuation’s corresponding index, contain at least one (1) symmetric set of numbers.

To avert front-running during any given audit state, a 7-qubit quantum register superposition is initialized and simulated via Google's qsim C++ statevector simulator (Appendix A) [4]. The resulting distribution is sampled deterministically during consensus; nodes are not required to execute quantum hardware or maintain a live quantum state. The circuit evaluates the normalized amplitudes of distinct values in each column, accumulated by their frequency f , such that the probability of measurement outcome i is:

$$p_i = \frac{f_i^2}{\sum_{j=1}^N f_j^2}, \text{ where } N \text{ is the column length.}$$

Presence of a target value is confirmed via M repeated measurements—dynamically scaled by active network nodes. The computational cost of evaluating an audit state is strictly proportional to network participation. This ensures that the expected block reward cannot be cheaply computed for arbitrary audit permutations, binding the evaluation cost to the network's own entropy.

A match occurs for every valid multiset if at least one collapse of the audit state aligns with the target and satisfies the current stage threshold, as long as it also passes the distinct user test.* The total number of matches per audit state, assuming a result greater than or equal to one, determines the final size of the block reward.



* = The distinct user test requires that, for a match to be considered valid, the set of valuations from which the matches are derived must consist of at least a threshold number of users equivalent to the current stage. This raises the cost of simple Sybil attacks from a single node filling the buffer to force a mint.

The stage threshold is calculated by adding the minimum number of matches required to propose a block, with an increment of one (1) for every two million coins issued, offset by a decrement of one (1) for every two million coins actively staked.

The formula for the current Stage Threshold is:

$$T = \max \left(2, T_{\text{base}} + \left\lfloor \frac{C_{\text{issued}}}{R} \right\rfloor - \left\lfloor \frac{C_{\text{staked}}}{R} \right\rfloor \right)$$

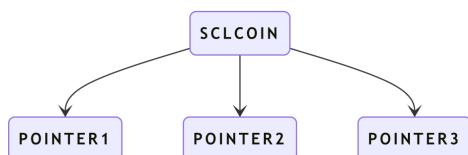
Where:

- **T** : The current **Stage Threshold**.
- **T_{base}** : The **Base Threshold** value, where $T_{\text{base}} = 2$.
- **C_{issued}** : The total number of coins that have been **issued**.
- **C_{staked}** : The total number of coins that are currently **staked**.
- **R** : A constant **Ratio** value, where $R = 2,000,000$.
- **[]** : The **floor function**, which means the result of the division is rounded down to the nearest whole number.

This formula enforces checks and balances on centralization, encouraging Nash equilibria without strict trust guarantees. For example, though it is technically possible for all **sixteen million (16,000,000) coins** to become staked at the same time, it is equally unlikely because it is not in the best interest of the network. The current stage dynamically adjusts towards a lower threshold as the number of staked coins increases, supporting higher minting odds and aligning with the expected spike in validation rates. Nodes should then react accordingly, as the negative pressure on market value will likely promote decreased staking to counter inflation.

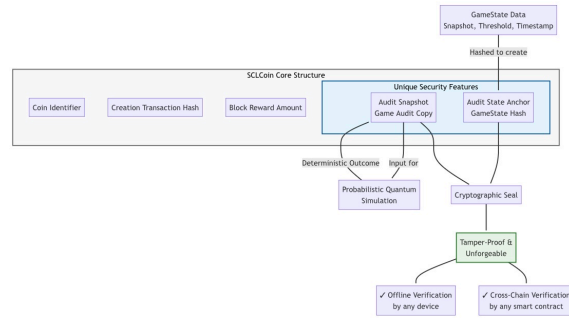
In the event that unrestricted supply saturates the market, the current stage dynamically adjusts towards a higher threshold as the number of staked coins decreases, lowering minting odds and indirectly reducing validator strength. This signals that the market should proceed to lock up enough capital to reduce the stage threshold, stabilize network throughput and reintroduce deflationary measures. Otherwise, market participants risk destroying the ability to trade, which lowers demand, erases liquidity and makes it virtually impossible to establish economic value by completely halting the blockchain. By setting soft limits on supply and demand, both greed and fear remain in sync for the prosperity of the economy. The long-term stability of this mechanism is intended to withstand the demands of agent-based economic activity, particularly under conditions of rapid staking, unstaking, and fee-market volatility.

3. Tokenomics



SCLCoin is the static virtual currency native to the Social Commodity Layer (SCL) 1 blockchain. While an SCLCoin structure can represent an arbitrary number of subunits, a single SCL is equivalent to **one hundred and sixty million (160,000,000) subunits** for calculation purposes. It can be described as a Post-Monetary Transaction Map (PTXM), a modular Unspent Transaction Output (UTXO) [5] that facilitates control over a fraction of a coin's total equity by mapping a specific range of subunits (Pointer) to its parent. This hierarchy allows for fine-grained, variable ownership, which enables scalable, peer-to-peer (P2P) transfers.

Each SCLCoin structure includes an amount relative to a block reward, a coin identifier (ID) and a creation transaction hash. What makes it unique, however, is the presence of an audit snapshot, as well as an audit state anchor. The audit snapshot is an exact copy of the Game Audit at the time of the minting ceremony that allows any validator on the network to reproduce the value of the reward by using a simulation of quantum superposition natively integrated with Google's qsim. The audit state anchor is a hash of the entire GameState when the reward was calculated—including the audit snapshot, stage threshold and audit timestamp.



Not only does the snapshot make it impossible to artificially inflate the value of a block reward, it ensures audit consistency and combines seamlessly with the tokenization of time. A counterfeit SCLCoin cannot be constructed because manipulating any aspect of the GameState breaks the cryptographic seal and renders the anchor unverifiable. Every contributed unit (Validation) must be immutably witnessed (Valuation) and frozen in a tamper-proof coin structure, avoiding retroactive alteration or forgery. Since new SCL can only be created through verifiable events, triggered only by accumulated time (validation rate), it is the first asset class that qualifies as a Store of Time.

Interoperability at the protocol-level seeks to reduce reliance on traditional oracle infrastructure for cross-chain bridging and enable efficient, zero-infrastructure payments. As it is self-verifying, SCLCoin can be exported to foreign blockchains without complex trusted intermediaries. By recomputing the deterministic probability distribution from the snapshot and verifying the deterministic sampling result, any smart contract can verify that the coin was minted according to the consensus rules of the network. Likewise, a receiver in a region with fractured or nonexistent internet connectivity can easily perform **offline authenticity** and state-anchor verification on any device using the embedded proof without querying full nodes. While global double-spend finality requires subsequent network synchronization, this provides robust decentralized verification of coin provenance and state validity.

PTXM's recursive architecture differentiates between simple or even complex timestamped logs by converting a record of time (Validation-to-Valuation) into fungible time. The audit is consensus-dependent and mutable until minted, at which point the most valuable human resource becomes a provably causal artifact that can then be distributed. It is plausible that the constraint of spending time leads to its accrual, with each phase of ownership strengthening an irreversible chain of custody; a low-level mandate for sophisticated resource allocation and a distributed basis for maintaining Universal Basic Income (UBI).

Fragmenting the mining process ultimately lowers the barrier to entry for financial inclusion. Aside from blockchain data storage limitations, mitigated by generating succinct proofs appended to a Merkle Mountain Range (MMR), a node can extract more value from a valuation compared to the low probability of a single hash, avoiding the high costs of electricity associated with Proof-of-Work. The centralization risks introduced by mining pools are eliminated, which were only later masked with the wealth concentration pitfalls of Proof-of-Stake. Instead, these flaws are permanently addressed by the heightened utility of a Validation, and democratized by its unapologetic availability to all practical entities.

4. Security

Core tenets of the blockchain are fortified ab initio using ML-DSA-87 (FIPS 204, standardized from CRYSTALS-Dilithium), targeting NIST Security Category 5 parameters. This provides robust hardness assumptions against known classical and quantum algorithms.

Except for initial peer discovery, all transport communication is encapsulated via ML-KEM-512 (FIPS 203, standardized from CRYSTALS-Kyber), providing Category 1 Post-Quantum Key Encapsulation. However, AES-256 in Galois/Counter Mode (GCM) is utilized to produce an authentication tag that provides both confidentiality and sovereign data integrity. Hash-based Message Authentication Code (HMAC) is derived during the handshake to support “defense-in-depth” extensibility. The specific implementation stack is integrated using the open-source liboqs C library developed by the Open Quantum Safe project [6].

Staked assets utilize a customized threshold variant of ML-DSA-87 [7] operating at the validator staking layer. For individual stake/unstake pointer operations, a client-dealer threshold signing scheme operates with parameters ($t = 2, n = 3$) to construct valid spending signatures across designated peer nodes without exposing the complete private key. (This is distinct from the dealerless, network-wide DKG beacon consensus detailed in Section 5). The final threshold Dilithium equation is:

$$\mathbf{z} = \sum_{i=0}^2 (\mathbf{y}_i + \mathbf{c} \cdot \mathbf{s}_{1,i}) \pmod{q}, \quad \|\mathbf{z}\|_{\infty} < \gamma_1 - \beta$$

The check $\|\mathbf{z}\|_{\infty} < \gamma_1 - \beta$ ensures that the secret shares are statistically hidden inside the added noise $\mathbf{y}_i$.

Where:

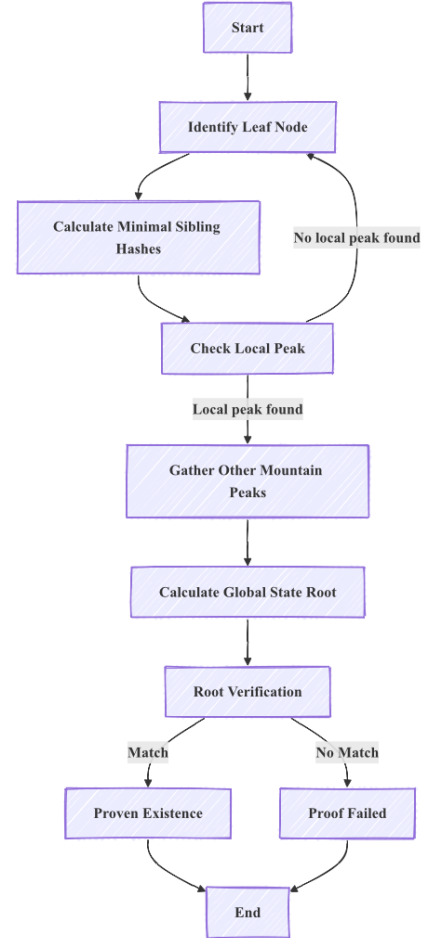
- $\mathbf{z}$: aggregated response vector, computed as the sum of all parties’ partial responses.
- $\mathbf{y}_i$: ephemeral masking vector generated locally by party i .
- $\mathbf{c}$: challenge polynomial (coefficients in $\{-1, 0, 1\}$, Hamming weight $\tau = 60$).
- $\mathbf{s}_{1,i}$: secret-key share of party i .
- q : ring modulus.
- $\|\cdot\|_{\infty}$: infinity norm (max absolute coefficient).
- γ_1 : rejection-sampling bound.
- β : bound on $\|\mathbf{c} \cdot \mathbf{s}_1\|_{\infty}$ ($\tau \times \eta \times 3 = 60 \times 2 \times 3$).

On the other hand, transferred assets are complemented by a novel interpretation of a Zero-Knowledge Pseudo-Random Function (ZK-PRF) [8] layer based on the Short Integer Solution (SIS) problem and MLWE, achieving the fundamental hardness assumptions of Kyber and Dilithium. It binds a user's secret PRF key to a transaction ID and the MMR root without revealing the key itself. Since the PRF is deterministic, it proves that an input context was signed by a specific user, deterring double-spending attempts as well as replay attacks. Verifiers must enforce that the transaction identifier and MMR root are uniquely bound to the proof context to prevent cross-context replay. The formula for calculating the PRF key is:

$$y = (A \cdot s) \pmod{p}$$

Where:

- y : The deterministic output vector containing the lower 8 bits of the computation.
- A : A public matrix expanded from the transaction seed.
- s : The user's sparse, ternary secret vector with coefficients in $\{-1, 0, 1\}$.
- p : The projection modulus, which reduces the ring element to a single byte per coordinate.



Utilizing a non-interactive, Stern-like protocol with 219 repetitions, the proof attains 128-bit PQ security as well as a drastic reduction in overall bandwidth. Instead of transmitting the large matrix A , the prover transmits 32-byte seeds so that the verifier can regenerate the matrices using SHAKE256 (XOF). Since the secret s is ternary (-1, 0, 1) and inputs are binary, they are compressed significantly before hashing to keep proof sizes manageable. If the user tries to create a different proof for the same transaction index/root, the deterministic PRF output would force a collision or fail verification. This transparent approach is essential for enabling private, unforgeable pointers within the transaction graph, which protects the anonymity of the asset flow while maintaining regulatory compliance.

5. Scalability

Addressing core challenges in post-quantum blockchain design requires tightening the interplay between security and decentralization. Central to this is the data availability problem: ensuring every network participant can access and independently verify the transaction data in each new block, thereby preserving the integrity of the protocol. The MMR supports this goal by enabling efficient light-client verification.

An MMR proof comprises the minimal set of sibling hashes needed to ascend from a leaf to its local peak, together with the remaining mountain peaks required to recompute the global state root. When the recomputed root matches the root in the block header, both the existence of the coin and its assigned value are cryptographically confirmed. Because new blocks update the structure incrementally, the cost remains $O(\log n)$ rather than requiring a full $O(n)$ re-hash of historical data.

To put it in perspective, a leaf is not just a transaction hash. Cryptographic commitments to the values held in SCLCoin structures transform the MMR into a verifiable state machine. It can be described as a deterministic composition of a sequence of binary trees of varying heights, which allows the network to commit the entire history of the economy into a 32-byte string. The ease of proving that a coin not only exists, but also holds a specific amount of equity at a particular block height, helps detect inflation bugs or unauthorized value tampering without mobile devices or IoT participants needing to scan the full ledger.

Even as the blockchain grows to petabytes of history, the proof size for any given coin remains logarithmic. However, the MMR pivots to long-term resilience in its support for aggressive pruning. Once a “mountain,” or an isolated timeframe of economic activity, is complete and buried under sufficient confirmations, the peak hash verifies future states and the internal leaves of finalized mountains can, in theory, be discarded. Simply put, a receiver in a low-bandwidth environment (offline/mesh networks) can verify a payment using only the block header and the compact MMR proof without trusting a third-party node.

Correlations between Scalability and Security form a recurring technical moat throughout the protocol. For example, a ZK-PRF proof binds the MMR root directly to the user’s PRF key. To achieve high-throughput scaling without sacrificing security, the protocol disentangles block proposal generation from block finality. Block finality is governed by a 67% supermajority BFT voting consensus among active network validators. Consensus extends threshold ML-DSA to establish an unforgeable epoch signing key via a decentralized, dealerless Distributed Key Generation (DKG) ceremony among the active validator committee ($t = 2, n = 10$), distinct from client-level stake authorization. Rather than requiring heavy signature propagation for individual state checks, this threshold signing ceremony produces a 256-bit unforgeable Randomness Beacon. This beacon seeds the qsim quantum statevector simulator. For consensus execution, nodes deterministically sample that distribution using the classical procedure in Appendix A. This suppresses grinding attacks while ensuring absolute, IEEE 754 cross-platform determinism across all participating nodes.

These relationships are further cemented by the fee pool mechanism linking post-cap sustainability to the blockchain’s longevity. For every unspent fee pointer that is burned, a new pointer is created using a hash of the recycled amount for the creation transaction hash. Although the issuance logic relies on consistent transaction fees to sustain velocity, it remains impervious to block-level dynamics and the overarching economic state.

In terms of the least viable economic outcome where all coins are effectively staked ($\sim 1.57\%$ fee), approximately $\sim 251,327$ SCL will remain eligible for recirculation on a fluid basis through PoS² consensus. This conservation of internal energy, so to speak, is balanced by the Stage Threshold formula as it is exponentially difficult to increase supply when fees are neglected. Contrary to the traditional ideologies that govern “Digital Gold” phenomena, low opportunity costs are indefinitely preserved throughout all phases of adoption, reimagining sound money principles without the gravity of perpetual dilution or the caveats of a strictly deflationary virtual currency.

6. Risks

A Sybil attack where a malicious actor spawns thousands of peer instances to dominate validation weight and centralize the economy is relatively cheap, compared to the economic costs associated with 51%-style attacks. At the same time, however, game theory is intended to reduce the window of opportunity to the last ten (10) audit inputs. Every additional competing node—whether honest or malicious in nature—introduces minimal entropy, despite the presence of collusion between a group of mutually interested parties.

Reliance on quantum amplitude encoding introduces a unique 'Constructive Interference' attack surface. Since the probability of a state collapse scales quadratically rather than linearly, a coalition controlling a simple majority of the audit window achieves a disproportionately higher probability of dictating the block reward. This “rich-get-richer” physics phenomenon implies that while the game-theoretic window is small, the incentive for short-term collusion is mathematically amplified compared to classical probability distributions. The distinct user test, stage threshold adjustment, and fee-history weighting are intended to raise the economic cost of such collusion; however, formal simulation of coalition dynamics should be performed before mainnet.

The protocol is designed to inherit resistance from the “chain reaction” effect, so as to expect that at least ten (10) unique motivations will be present during transitions where the perceived value of the network is greatly influenced by the random diversity of the node pool. Even if a single entity were to accumulate a large majority of the total supply, the **static fee rate of $\sim 3.14\%$** for all self-fulfilling transactions ensures that wash trades feed the network by remaining prohibitively expensive in practice.

Custom PQC extensions also amplify exposure to subtle gaps in security, such as side-channel leaks, forgery and implementation bugs. Polynomial vector operations over ring elements and rejection sampling are subject to formal verification, with errors in share distribution leading to invalid or compromised signatures. Constant-time operations mitigate risks, but reductions may vary. To guarantee memory safety, the architecture incorporates compiler-level thread and memory sanitizers (ASAN/TSAN) alongside aggressive fuzzing harnesses that explicitly validate cross-architecture struct alignments and simulate replay attacks against sliding bitmask windows.

The intentional use of deterministic bindings serve to increase the computational infeasibility of collisions across vectors. Furthermore, to mitigate volumetric Denial-of-Service (DoS) and Distributed-Denial-of-Service (DDoS) attacks at line rate, the network utilizes fast-path eBPF/XDP shims to evaluate the SipHash-2-4 algorithm against cryptographic Connection IDs (CIDs). A dedicated 8192-slot asynchronous logging queue avoids bottlenecking consensus threads during these events.

P2P communication mirrors User Datagram Protocol (UDP) while implementing a custom QUIC-Parity L4/L5 transport layer designed explicitly for PQ network environments. To accommodate the elevated payloads, the underlying kernel UDP buffers are strictly configured to 8,388,608 bytes for both receive and send allocations, utilizing highly advanced shims to process packets at the kernel level before passing them to user-space. Because standard UDP is stateless, the shim introduces connection multiplexing by mapping peers to unique 20-byte Maglev-style CIDs and securely caching stateless reset tokens to manage peer disconnections and recovery seamlessly. The protocol allocates a strict 16 MB Out-of-Order memory ceiling per connection with a dynamic back-pressure limit at 8 MB, paired with a 4096-bit sliding replay window.

Connection establishment is negotiated through a rigorous 2-RTT PQ Handshake. To absorb CPU-scheduling jitter under intense signature verification, the Minimum Probe Timeout is intentionally elevated to 1000 ms for full nodes. Light-client and mobile profiles should expose a reduced-timeout, read-only synchronization mode where full block production and verification are not required. Initial handshake transmissions (Flight 1) are padded to 1232 bytes to prevent amplification attacks and stabilize MTU expectations. The system relies on a decayed Bandwidth-Delay Product limit of 32 KB, which prevents cryptographic thread lockups during cold-starts or heavy packet loss. Furthermore, the L5 layer features a built-in anti-spam and anti-DDoS mechanism utilizing a PoW cookie challenge.

When prompted by a server RETRY packet, connecting nodes must generate a 10-bit PoW before retransmitting Flight 1. Under complex network conditions, the protocol dynamically escalates to a Hybrid RETRY, issuing a 10-bit primary and 12-bit secondary PoW challenge. Once Flight 2 is processed and the payloads are successfully decrypted, the connection achieves an ESTABLISHED state. Throughout this process, the client continually monitors network topology, running internal classifications to identify Open Internet or Full Cone NAT traversal to optimize discovery and subsequent blockchain synchronization.

7. Roadmap

At peak maturity, and subject to independent security audit, SCL anticipates the structural realization of a Quantum-Commodity Hybrid architecture. While the *Guiding and Establishing National Innovation for U.S. Stablecoins Act (GENIUS Act)* [9], enacted on July 18, 2025, codifies rigid federal frameworks for digital assets tethered exclusively to static reserves, SCL proposes a radically dynamic “Commodity-Input” protocol. Within this horizon, the finite velocity of tangible physical artifacts is harnessed not as inert collateral to be vaulted in perpetuity, but as latent kinetic potential to be active and consumed—serving as an alternative, foundational energy source for network vitality. This structure is intended as a protocol-level input mechanism, not a custodial promise, redemption obligation, or static reserve arrangement; its regulatory treatment may vary by jurisdiction.

This orchestration derives its mechanics from simulated superpositions of the Game Audit, bridging the gap between abstract mathematical probability and concrete economic utility. To democratize this transmutation beyond software interfaces, the protocol introduces the **Centinel (¢)**. Centinel functions as a private UTXO system designed to recirculate the U.S. penny following the conclusion of its mint production run on November 12, 2025. The protocol treats the estimated 300 billion one-cent pieces left remaining in global circulation [10] as untapped fuel cells. By exchanging them to underwrite the network’s single atomic unit of work, the planned Centinel extension converts real-world physical commodity inputs into private on-chain utility represented by the same BDLOP commitments, PRF nullifiers, and MMR accumulator state machine.

Here, classical wave-particle duality finds a physical and macroeconomic analogue:

The Particle (Input: The Penny): An object of absolute, discrete value representing the **Tortoise**. It functions as a localized stake of historical labor that bootstraps the decentralized digital economy.

The Wave (Output: The SCLCoin): A quantum-secure cryptographic byproduct representing the **Hare**. It functions as a fluid, probabilistic waveform of systemic equity, measured by the natural fluctuation of capitalistic ideals.

The Centinel resonates as an evolutionary bridge across the Newtonian-quantum divide, aiming to eliminate the cognitive friction associated with traditional cryptocurrency on-ramps.

By abstracting complex private key management and remote remittances into a ubiquitous, tactile exchange—the conversion of a commodity for an on-chain digital service—SCL levels the playing field for unbanked populations. This empowers mesh-network participants to swap a penny’s stable face value for a secure representation of its intrinsic quantum state, effectively crowd-sourcing network security from demographics previously alienated by cryptographic opacity.

Through this fusion, industrial base metals are reappropriated into numismatic quanta whose components (zinc/copper) are quantified at a maximum fixed rate of approximately ~\$0.037, extracted from the remnants of the fiat age. This figure is used as a protocol-level economic constant and does not constitute a market price guarantee for physical pennies. As the Centinel supply fluctuates, it emulates both the scarcity and abundance required to facilitate frictionless trade at the reserve level. Ultimately, this framework harmonizes the raw profit motive with physical resource conservation. It stands as the true realization of sound money fundamentally reimagined—structurally independent from, yet intrinsically compliant with the forward-looking spirit of, modern financial innovation.

8. Conclusion

Throughout history, we often see money expressed as a function of time. A worker receives compensation for their time and labor, or the perspective that the value of money changes over time. Yet, Ben Franklin’s 1748 declaration [11] that time *is* money tends to favor philosophical discussion rather than literal argument. PoS² explores his later thoughts, remembering that credit is [preferred] money; essentially, it’s how new money is created. Born from an understanding that blockchain technology—an innovation based solely on the decentralized custody of value—transcends recalcitrant maximalism towards banking and regulation, it moves to establish a relevant use case [12] where interest is earned on time, not necessarily money.

The goal of PoS² is to distribute access to storing time. It does not preemptively slash its participants for egregious behavior because the protocol is designed to dynamically adjust risk and reward. It believes that diversity only exists when the target of speculation is the pinnacle of universally shared value. Most importantly, it definitively states that a scalable, decentralized security sandbox—Store of Time as the symbol of a quantum handshake—stands as a formal contribution to the cypherpunk community and the entire technological landscape.

References

- [1] Vitalik Buterin. 2017. Sharding FAQ. https://vitalik.eth.limo/general/2017/12/31/sharding_faq.html
- [2] Commodity Futures Trading Commission. 2015. Order Instituting Proceedings Pursuant to Sections 6(c) and 6(d) of the Commodity Exchange Act, Making Findings and Imposing Remedial Sanctions. *CFTC Enforcement Actions*. <https://www.cftc.gov/sites/default/files/idc/groups/public/@lrenforcementactions/documents/legalpleading/enfcoinfliprorder09172015.pdf>
- [3] Emmy Noether. 1971. Invariant Variation Problems. *Transport Theory and Statistical Physics*, 1(3), 186–207. <https://doi.org/10.1080/00411457108231446>
- [4] Google Quantum AI. 2024. qsim: A C++ High-Performance Quantum Circuit Simulator. Google Cloud Platform. <https://github.com/quantumlib/qsim>
- [5] Satoshi Nakamoto. 2008. Bitcoin: A peer-to-peer electronic cash system. <https://bitcoin.org/bitcoin.pdf>
- [6] Douglas Stebila and Michele Mosca. 2017. Post-quantum key exchange for the Internet and the Open Quantum Safe project. In *Selected Areas in Cryptography – SAC 2016*, Roberto Avanzi and Howard Heys (Eds.). Springer, Cham, 1–24. <https://openquantumsafe.org>
- [7] Rafaël del Pino and Guilhem Niot. 2025. Finally! A Compact Lattice-Based Threshold Signature. In *Proceedings of the 2025 International Conference on Public-Key Cryptography (PKC ‘25)*, Springer, Cham, 169–199. <https://eprint.iacr.org/2025/872.pdf>
- [8] Benoît Libert, San Ling, Khoa Nguyen, and Huaxiong Wang. 2017. Zero-Knowledge Arguments for Lattice-Based PRFs and Applications to E-Cash. *Cryptology ePrint Archive*, Paper 2017/856. <https://eprint.iacr.org/2017/856.pdf>
- [9] U.S. Congress. 2025. Guiding and Establishing National Innovation for U.S. Stablecoins Act (GENIUS Act). Public Law 119-27. *Government Publishing Office*, Washington, DC. <https://www.congress.gov/119/plaws/publ27/PLAW-119publ27.pdf>
- [10] United States Mint. 2025. United States Mint Hosts Historic Ceremonial Strike for Final Production of the Circulating One-Cent Coin. *U.S. Department of Treasury Press Release*. <https://www.usmint.gov/news/press-releases/united-states-mint-hosts-historic-ceremonial-strike-for-final-production-of-the-circulating-one-cent-coin>
- [11] Benjamin Franklin. 1748. Advice to a Young Tradesman. In *The Papers of Benjamin Franklin*, Vol. 3, Leonard W. Labaree (Ed.). Yale University Press, 304–308. <https://founders.archives.gov/documents/Franklin/01-03-02-0130>
- [12] Tobias Adrian, et al. 2025. Understanding Stablecoins. *IMF Departmental Paper No. 2025/001*. International Monetary Fund, Washington, DC. ISBN 979-8-229-02407-5. <https://www.imf.org/-/media/Files/Publications/DP/2025/English/USEA.pdf>
- [13] Google Quantum AI. 2024. Google Quantum Engine API. Google Cloud Platform. <https://quantumai.google/>

Appendix A

The routine below is not a quantum execution engine. It samples a precomputed probability distribution, probs, generated from the corresponding Cirq circuit. The linear congruential generator is used solely to provide a lightweight, deterministic, cross-platform sampler for consensus execution. It is not intended as a general-purpose cryptographic random number generator.

The quantum evaluation engine natively integrates Google's qsim C++ library [13] (with a strictly standardized IEEE 754 statevector fallback) for 7-qubit register simulation ($2^7 = 128$ state amplitudes). By binding the PRNG seed directly to the 256-bit DKG-derived Randomness Beacon Seed via SHA-256 and utilizing a standardized Cumulative Distribution Function (CDF) scan, measurement sampling achieves deterministic consensus equivalence across operating systems and hardware architectures (ARM64/x86_64). Monotonically scaling the measurement count M with network participation bounds computational complexity while maintaining low-latency consensus.

```
#include "quantum_qsim.h"
#include <cmath>
#include <cstring>
#include <iostream>
#include <random>
#include <vector>
#include <openssl/sha.h>

// If qsim is installed in the build environment, include its headers
#if __has_include(<qsim/simulator.h>)
#define HAS_QSIM_LIB 1
#include <qsim/simulator.h>
#include <qsim/state_space.h>
#endif

extern "C" {

int quantum_check_number_in_column_qsim(int target_number, const int *column_numbers, size_t N, size_t active_nodes,
const uint8_t *beacon_seed) {
    if (N == 0 || target_number < 0 || target_number >= 100) return 0;

    // Calculate M (Monotonically scales upward with network participation)
    int base_M = 15000;
    int scale_factor = 50;
    int M = base_M + static_cast<int>(active_nodes) * scale_factor;
    if (M < 1000) M = 1000;
    if (M > 50000) M = 50000;

    // Derive deterministic 64-bit seed from DKG Beacon
    uint64_t strict_seed = compute_dkg_beacon_seed_64(target_number, column_numbers, N, active_nodes, beacon_seed);

#ifdef HAS_QSIM_LIB
    // Execute natively utilizing Google qsim
    using Simulator = qsim::Simulator<qsim::For>;
    using StateSpace = Simulator::StateSpace;

    StateSpace state_space(7); // 7 qubits = 128 states
    StateSpace::State state = state_space.CreateState();
    state_space.SetStateZero(state);

    double norm_factor = 1.0 / std::sqrt(static_cast<double>(N));
    for (size_t k = 0; k < N; k++) {
        int num = column_numbers[k];
        if (num >= 0 && num < 100) {
            // Directly inject amplitudes
            state_space.SetAmpl(state, num, norm_factor, 0.0);
        }
    }

    std::mt19937_64 prng(strict_seed);
    std::vector<uint64_t> results = state_space.Sample(state, M, prng);

    for (uint64_t res : results) {
        if (static_cast<int>(res) == target_number) return 1;
    }
    return 0;
#else
    // Strict IEEE 754 Deterministic Fallback C++ Implementation
    std::vector<double> amplitudes(128, 0.0);
    std::vector<double> probs(128, 0.0);
    double norm_factor = 1.0 / std::sqrt(static_cast<double>(N));
```

```

double total_prob = 0.0;

// 1. Accumulate amplitudes linearly (Constructive Interference)
for (size_t k = 0; k < N; k++) {
    int num = column_numbers[k];
    if (num >= 0 && num < 100) {
        amplitudes[num] += norm_factor;
    }
}

// 2. Square amplitudes to derive quadratic probability (f_i^2 / N)
for (int i = 0; i < 128; i++) {
    probs[i] = amplitudes[i] * amplitudes[i];
    total_prob += probs[i];
}

if (total_prob <= 0.0) return 0;

// 3. Normalize probabilities
for (int i = 0; i < 128; i++) {
    probs[i] /= total_prob;
}

if (probs[target_number] <= 0.0) return 0;

// Deterministic CDF sampling replacing implementation-defined std::discrete_distribution
std::vector<double> cdf(128, 0.0);
double cumulative = 0.0;
for (int i = 0; i < 128; i++) {
    cumulative += probs[i];
    cdf[i] = cumulative;
}
cdf[127] = 1.0;

std::mt19937_64 prng(strict_seed);
const double to_unit = 1.0 / static_cast<double>(std::mt19937_64::max());
for (int m = 0; m < M; m++) {
    double sample_val = static_cast<double>(prng()) * to_unit;
    int sampled_idx = 0;
    while (sampled_idx < 127 && sample_val > cdf[sampled_idx]) {
        sampled_idx++;
    }
    if (sampled_idx == target_number) {
        return 1;
    }
}
return 0;
#endif
}

} // extern "C"

```