The Markovian Protocol

A Proof-of-Intelligence Protocol Powered by Markov State Transition

Version 0.8 - June 2026

Author: Colin Winter

Abstract

Bitcoin solved the double-spend problem. The security model is sound. The computation produces no output beyond the security it purchases. SHA-256 hashing generates heat and irreversible work, both of which are discarded the moment a block is verified. The network accumulates ledger entries. It accumulates nothing else.

The Markovian Protocol proposes a different work function.

Miners compute N transitions of a published 3x3 stochastic transition matrix M from a starting vector s derived deterministically from the previous block hash. The output is a probability distribution across three economic regime states: Accumulation, Markup, Distribution. Each valid block adds a verified data point to a permanent archive of market regime history. The work function is deterministic, independently verifiable, and difficulty-adjustable via the transition depth parameter N. The validity condition is not arbitrary. M is derived from fifteen years of price data via Hidden Markov estimation, ratified by Byzantine fault tolerant validator governance, and committed on-chain with cryptographic proof. The protocol defines what constitutes a valid transition. That definition is itself a claim about market structure.

The archive has commercial value. Quantitative funds, risk desks, and algorithmic trading systems require labeled regime history. The Markovian Protocol produces that data as a byproduct of consensus, with zero-knowledge proof of correctness and Merkle-rooted provenance on every block.

The coin is priced by the demonstrated intelligence of the network. Total dollar value of data produced divided by circulating supply and velocity produces a computable fundamental value. The valuation is anchored to real economic utility, not supply constraint.

The supply is dynamic. Emission is proportional to verified network output: governance cycle accuracy, archive depth, validator participation, and commodity volatility. The emission rate is a scoreboard, not a schedule.

The transition matrix M is a governance parameter. A Byzantine fault tolerant supermajority of validators, weighted by historical accuracy, proposes and ratifies matrix updates on a defined cycle. The protocol's model of market structure improves with the archive. The data compounds in accuracy. The coin follows.

1. The Problem

Nakamoto consensus is a well-understood mechanism. By making block production expensive and verification cheap, Bitcoin created a trustless ledger that has operated without interruption for over fifteen years. The security model is battle-tested.

The work is not reused. SHA-256 hashing produces nothing of value beyond the security it purchases. Miners burn electricity, generate heat, and discard the outputs. The system functions because the waste is expensive enough to make attacks uneconomical. The incentive structure is sound. The output is not.

Proof of Useful Work has been proposed as an alternative - mining that produces economically valuable computation as a byproduct of consensus. Prior attempts have failed on one of two grounds: the useful computation is not independently verifiable without trust, or the computation is insufficiently resistant to strategic gaming.

The Markovian Protocol addresses both constraints. Markov state transition is deterministic - identical inputs produce identical outputs on any hardware. Zero-knowledge proofs make verification trustless and instantaneous. Difficulty is adjustable via transition depth N. The output is a probability distribution across economic regime states derived from an empirically fitted governance model.

---

2. Prior Art

Prior attempts at useful proof-of-work fall into three categories, each failing on a common underlying constraint.

ML Training as PoW

PoGO (arxiv:2504.07540, 2025) uses quantized gradient descent with Merkle proofs to prove miners are training large-scale models. Verification cost is less than training cost. The approach has been demonstrated at GPT-3 scale, 175 billion parameters.

The constraint is non-determinism. The same gradient step on the same data produces different floating point outputs on different hardware due to parallel operation ordering. No single verifier can confirm a block is correct without re-executing it. PoGO falls back to probabilistic sampling, approximating rather than proving correctness. Blocks require hours. The system cannot operate at Bitcoin-cadence block times.

Non-determinism is not a defect in PoGO's implementation. It is a property of floating point arithmetic under parallel computation. There is no patch.

Validator-Scored Intelligence

Bittensor (TAO, 2022) is the largest deployed proof-of-intelligence network. 4,096 active nodes provide model outputs. Validators score those outputs using other models. Rewards distribute via Yuma Consensus, a sigmoid-threshold mechanism requiring more than 50% of network stake to confirm a ranking.

The protocol's own paper states the constraint directly: "The blockchain does not trust rankings from any individual peer on the network, but rather trusts the collective rankings from all participating peers."

Output quality is subjective by construction. Useful embeddings for one validator may score poorly with another depending on training alignment. There is no objective standard. There is no single verifier. Correctness is consensus of opinion.

Proof of Useful Intelligence (PoUI, arxiv:2504.17539, 2025) is structurally identical. Workers execute AI tasks, validators score outputs, rewards flow from agreement. Output correctness is not cryptographically verified. Social consensus is substituted for cryptographic proof.

General Optimization PoUW

The formal security literature (arxiv:2405.19027, Cao et al., 2024) identifies the core trade-off explicitly: useful problems have structure that can be exploited. The more computationally valuable the work function, the harder it is to construct as a secure puzzle. Useful work and hard puzzles are in tension. The literature proposes matrix multiplication as a candidate work function. The construction is technically correct. The anchor to real-world data is absent.

Common Failure Mode

Every prior system arrives at one of two failure modes.

Non-determinism: useful computation produces different outputs on different hardware. Probabilistic approximation is not proof. Blocks cannot be independently verified. The network is required to trust validators, miners, or statistical sampling. Trust re-enters the system.

Subjective scoring: output quality is evaluated by other models or validator committees. Correctness is consensus of opinion rather than mathematical fact. The protocol cannot distinguish a competent miner from a colluding coalition without sufficient voting stake.

PoGO is non-deterministic by construction. Bittensor and PoUI are subjective by design. Neither constraint is resolvable within the respective architecture.

---

3. The Protocol

3.1 The Transition Matrix

The Markovian Protocol is built around a 3x3 stochastic transition matrix M encoding the probability of transitions between three economic regime states:

- State 0: Accumulation - capital accumulation phase, low volatility, range-bound price action
- State 1: Markup - directional price discovery, expanding participation, trend confirmation
- State 2: Distribution - late-cycle behavior, deteriorating market internals, capital rotation out

M is published by the protocol. Every node holds an identical copy. Every miner uses the same matrix. There is no ambiguity in the validity condition.

3.2 The Starting Vector

The starting state vector s is derived deterministically from the previous block hash. A defined extraction function maps the 256-bit hash to a valid 3-dimensional probability simplex - a vector of three non-negative values summing to 1.0. The starting vector requires no external data, no oracle, and no off-chain input. It is derived entirely from the chain state.

3.3 The Work Function

Given M and s, the miner:

1. Computes s_N = M^N * s, applying the transition matrix N times to the starting vector
2. Combines s_N with a nonce value to produce an input string
3. Hashes the input string
4. Checks whether the hash meets the current difficulty target

The miner iterates nonce values until a valid hash is found. This is the search problem - identical in structure to Bitcoin mining, different in computation. The protocol targets a 60-second block time. Initial transition depth N is 1,000 steps at genesis.

3.4 Dual Proof-of-Work

The Markovian Protocol supports two parallel mining paths. The primary path uses the Markov state transition work function described above. The secondary path uses RandomX, a CPU-optimized algorithm resistant to ASIC specialization. Both paths produce valid blocks. Both earn Kovs at protocol-defined rates. Bitcoin miners may merge mine MKV via Auxiliary Proof of Work - no additional hardware, no additional electricity.

The dual architecture distributes security across hardware types and mining communities, reducing concentration risk at the consensus layer.

3.5 Zero-Knowledge Verification

Every valid block submission includes a ZK proof demonstrating:

- The miner used the canonical matrix M
- The miner used the correct starting vector s derived from the previous block hash
- The miner computed exactly N transition steps
- The output vector s_N is correctly derived

The proof system uses BN128 elliptic curve Pedersen commitments combined with Schnorr sigma proofs. Commitment binds the computation to the canonical inputs. The sigma proof demonstrates correct execution without revealing intermediate state. Any node can verify the proof in milliseconds without re-executing the computation.

Layered Proof Architecture

The proof system is structured in four independent layers. Each layer is verifiable independently of the others.

Layer 1 - Matrix provenance: GENESIS_M is committed via Pedersen commitment against a deterministic hash of 29,795 market observations across five instruments spanning 2000 to present. The training hash is published. Any party can reproduce it from the same dataset. The commitment cannot be opened to a different matrix without invalidating the proof.

Layer 2 - Computation correctness: each Markov transition step carries a Schnorr sigma proof on BN128. One proof per output component, three per step, N proofs per block. Verification does not require re-computation. A single invalid transition invalidates the block.

Layer 3 - Input provenance: signal synthesis commits to its inputs before execution. Gate state, price data, regime vector, and agent outputs are hashed and committed prior to synthesis. The input commitment is linked to the output Merkle root in a single provenance record. The two cannot be constructed independently.

Layer 4 - Miner credibility: regime predictions are committed on-chain prior to resolution via SHA256(address, ticker, predicted regime, target block, nonce). At resolution the chain compares the committed prediction against the actual observed regime. Governance weight is derived from on-chain prediction history, not declared by the participant.

The entire system rests on a single security assumption: discrete logarithm hardness over BN128. This is the same assumption underlying Ethereum's ZK-EVM. Breaking any layer of the Markovian proof system requires solving a problem that has resisted the cryptographic community since 1976.

3.6 Difficulty Adjustment

The network targets a 60-second block time. Every 2,016 blocks, the protocol measures actual block time against target and adjusts N, the transition depth, accordingly. Higher N requires more computation per mining attempt. Lower N requires less. The adjustment window mirrors Bitcoin's two-week retarget cycle, adapted to the 60-second target.

3.7 What Proof-of-Intelligence Means

Proof-of-intelligence is not a claim about the miner. It is a claim about the validity condition.

In Bitcoin, a hash is valid if it has sufficient leading zeros. The target is arbitrary. It carries no information about markets, regimes, or economic structure. The work is discarded the moment it is verified. The network state after the block is identical to its state before, except for the ledger entry.

In the Markovian Protocol, a hash is valid only if it encodes a state transition consistent with the canonical matrix M. M is not an arbitrary target. It is an empirically derived model of market regime dynamics, fitted to fifteen years of price data, updated through Byzantine fault tolerant governance, and committed to the chain with zero-knowledge proof. The validity boundary is derived from structured reasoning about market dynamics.

Miners do not perform the reasoning. They search for valid transitions within a boundary the protocol has defined. The reasoning is in the matrix. The matrix is the product of the governance process.

This is the precise sense in which the work is proof-of-intelligence: not that computation is intelligent, but that the standard of correctness is derived from an intelligence process rather than an arbitrary numerical target. The miner proves it found a true transition. The protocol determined what true means.

Proof as a Function of Intelligence

Validator-scored intelligence networks define useful output, then route that definition through consensus. The validator vote becomes the proof. Consensus of opinion does not constitute a cryptographic guarantee. It is subject to the same failure modes as any social mechanism: coordination, capture, and drift under adversarial pressure.

The Markovian Protocol separates the two claims. The intelligence is encoded in M - an empirical model derived from market data and ratified by governance. The proof verifies that M was applied correctly to the inputs in evidence. The miner's intent is not evaluated. The computation either satisfies the proof or it does not.

No deployed proof-of-intelligence network submits this claim to cryptographic verification. The Markovian Protocol does.

---

4. The Coin

Name: Markoin

Ticker: MKV

Base unit: Kov (1 MKV = 100,000,000 Kovs)

The supply is dynamic. There is no cap. There is no halving schedule. Emission is tied to verified network output - governance cycle accuracy, archive depth, validator participation, and commodity volatility. High volatility and active governance cycles produce higher emission. Stable, low-governance periods produce lower emission. The supply reflects the demonstrated output of the network.

Each Kov in circulation represents a unit of verified, ZK-proven, Merkle-rooted computation that the network produced and the governance layer ratified. Emission is evidence of work performed, not a schedule imposed in advance.

Standard UTXO model. Wallets hold Kovs. Transactions are signed, broadcast to the network, and included in blocks by miners.

4.1 Commodity Anchoring

The Markovian Protocol models economic regime transitions driven by real commodity flows. Oil supply shocks trigger distribution regimes. Gold accumulation signals risk-off markup. Copper expansion precedes economic growth phases. The protocol measures these transitions directly.

The Markoin derives value from signal accuracy on real-world commodity flows, not supply constraint. As the governance model improves, the signal becomes more accurate. As the signal becomes more accurate, the archive becomes more useful. Utility drives demand. Demand prices the coin.

4.2 Token Economics - Fisher's Equation of Exchange

The Kov economy is governed by Irving Fisher's Equation of Exchange:

MV = PT

- **M** - Kovs in circulation. Dynamic supply tied to verified network output. Emission rate is a governance parameter.
- **V** - Velocity. The rate at which Kovs change hands to access archive data.
- **P** - Price. Cost of archive access denominated in Kovs.
- **T** - Transactions. Archive queries, data pulls, governance votes.

As the archive deepens, T grows. Archive depth is the primary demand driver. Fisher's equation requires V and P to absorb the difference.

In most token systems, velocity collapses as holders accumulate. The Markovian Protocol addresses this structurally. Archive access requires Kovs to be staked for the duration of the session. Staking is not optional - it is the access mechanism. This creates a structural floor on V that is independent of market sentiment.

4.3 The BTC Settlement Layer

Settlement currency: Bitcoin.

Data buyers pay in BTC. Bitcoin is the universal settlement layer - the most trusted, most liquid, most decentralized asset in existence. The protocol requires no fiat bridge. It uses the one that already exists.

When a data sale occurs, BTC enters the protocol. Network participants who contributed verified work - miners, validators, contributors - receive a proportional share of archive access fees based on Kovs staked. The distribution mechanism is fee-sharing, not issuance. Kovs are the access and governance instrument. BTC is the settlement instrument.

The closed loop:

1. Archive depth grows with every block mined.
2. Archive utility drives T upward.
3. BTC from archive access fees flows to network participants.
4. Staking demand increases to participate in fee distribution.
5. V stabilizes. Fisher holds. P reflects genuine demand.

Early miners earn Kovs at the lowest acquisition cost. The archive has not yet proven its depth. When the first institutional buyer pays BTC for access, every participant with staked Kovs receives a share of that fee immediately. The incentive is a fee stream, not a narrative.

MKV is priced by demand. Settled in Bitcoin. Earned by verified work.

---

5. The Governance Layer

5.1 Matrix Governance

The transition matrix M is a protocol parameter subject to update. Validators propose revised matrix weights based on validated historical signal data. The network votes. A BFT supermajority of 2/3 or more approves or rejects. Ratified updates activate after a mandatory time-lock delay.

The matrix improves with the archive. Each governance cycle has access to more verified regime history than the last. The protocol's model of market structure converges toward accuracy over time.

5.2 Byzantine Fault Tolerance

Governance uses BFT consensus. Up to 1/3 of validators may be malicious or offline without compromising the result. As long as 2/3 of validators are honest, correct matrix updates pass and malicious proposals fail.

5.3 Commit-Reveal Voting

Validators submit a cryptographic hash of their vote in the commit phase. Votes are revealed only after the commit window closes. No validator can observe the emerging consensus and adjust their position accordingly. Strategic position changes are not possible.

5.4 Time-Lock Activation

Ratified matrix updates do not activate immediately. A mandatory delay of 2,016 blocks provides the network with a review window. Any node that identifies a compromised matrix update can exit or fork before activation.

5.5 Validator Credibility

Validator voting weight is proportional to historical accuracy. Nodes whose proposed matrix updates improved network performance earn greater governance influence. Credibility is derived from on-chain prediction history. It cannot be purchased or declared.

---

6. Multi-Chain Architecture

The protocol uses three independent chains, each serving a distinct function.

Bitcoin - the anchor.

Every block the Markovian Protocol produces is committed to Bitcoin via OP_RETURN. A 28-byte Merkle root, prefixed with the MKV magic bytes, is embedded permanently in the Bitcoin chain. Every regime classification the protocol has ever produced can be independently verified against the Bitcoin chain, in sequence, by any party, permanently. The archive cannot be rewritten. The provenance cannot be forged.

Direct BTC deposits are accepted. Send BTC to the protocol address with an OP_RETURN containing `MKV:YOUR_ADDRESS`. The deposit watcher detects the transaction. After three confirmations, Kovs are issued at the protocol rate.

Ethereum - the oracle.

The Markovian Protocol's regime classifications, ZK proofs, and Merkle roots are posted on-chain via the MarkovianOracle smart contract. Any Ethereum application can query live regime state for any covered instrument and build regime-gated logic against it.

The oracle contract accepts `updateBatch()` calls from the protocol's reporter address. The BN128 Pedersen commitment hash is included in every batch. The data is cryptographically linked to the source computation.

`lockAndMint()` is the Ethereum-native participation path. Lock ETH in the oracle contract for a defined period. The contract issues Kovs proportional to the amount locked and the lock duration. The lock expires, ETH returns. The Kovs remain.

Monero - the private settlement rail.

Bitcoin transactions are permanently visible. Ethereum is public by design. Institutional buyers who acquire regime signal data on a covered instrument are revealing a position. That purchase is alpha-sensitive. It will not be made on a public ledger.

Monero's ring signatures hide the sender. Stealth addresses hide the recipient. RingCT hides the amount. An outside observer observes that a transaction occurred, and nothing further. This is not a feature added to Monero. It is the architecture.

XMR payments are routed via subaddresses, one per user. The payment watcher detects the incoming transaction, confirms the amount, and issues Kovs to the registered address. No account, no email, no identity. Cryptographic proof of payment is the only credential required.

Unified Kov Issuance.

All Kov issuance routes through a single ledger regardless of origin chain. BTC deposit, ETH lock-and-mint, XMR payment, and direct mining all record to the same table. The source chain transaction hash serves as the deduplication key. Double issuance is not possible.

Issuance rates reflect the properties of each instrument. Rates are governance parameters, adjustable by validator supermajority as the network matures.

---

7. Security Model

Precomputation Attack: the nonce challenge changes every block. Precomputed tables are invalid across block boundaries.

Approximation Attack: ZK proofs commit to the exact computation. Any shortcut produces a proof that fails verification.

Guessing Attack: the PoW output is a continuous probability vector over three states. The valid search space is not enumerable.

Matrix Manipulation Attack: ZK proofs commit to the canonical M at the committed training hash. Any substitution produces an invalid proof.

51% Attack: same as Bitcoin. Mitigated by mining economics.

Governance Capture: BFT supermajority threshold, commit-reveal voting, time-lock activation, and fork rights operate as independent defenses. All four must be circumvented simultaneously for a

governance attack to succeed.

---

8. Roadmap

Phase 1 - Specification: white paper, math specification, protocol v1.0, community formation.

Phase 2 - Testnet: PoW function, ZK proof system, P2P networking, block explorer.

Phase 3 - Governance: validator registration, first matrix governance cycle, BFT implementation.

Phase 4 - Mainnet: genesis block, wallet release, MKV issuance begins.

Phase 5 - Data Marketplace: archive API launch, BTC settlement layer live, institutional access tiers, fee distribution to network participants begins.

---

9. Conclusion

Bitcoin answered the double-spend problem with energy expenditure. The work is real. The output is a secure ledger.

The Markovian Protocol answers the same security requirements with structured computation - the mathematics of state transition, verified by zero-knowledge proofs, governed by Byzantine fault tolerant consensus, and compounding in accuracy over time as the archive deepens.

The work is not discarded. It is verified, archived, and accumulated. The protocol grows more accurate with every block it produces. The supply reflects the cumulative output of the network.

The smallest unit is a Kov. Every Kov is cryptographic evidence that the network produced a verified regime classification. The protocol determined what valid means.