Ɉ Montana: Time-Proven Human Temporal Currency

A Peer-to-Peer Quantum-Resistant Electronic Cash System

Version 4.2 Alejandro Montana alejandromontana@tutamail.com **December 2025**

Abstract

A purely peer-to-peer electronic cash system would allow online payments without relying on financial institutions. Existing cryptocurrency solutions—Proof of Work and Proof of Stake—scale influence through purchasable resources, inevitably concentrating power in the hands of capital owners. We propose Ɉ Montana ($MONT), a currency built on Proof of Time consensus where influence accumulates through time presence rather than resource expenditure. The network timestamps blocks through sequential computation that cannot be parallelized or accelerated. Nodes compete for 21 million minutes of temporal asset distributed over 131 years.

Version 4.0 additions: The Twelve Apostles trust system and EPOCHS dimension: - 12 Apostles — Each node chooses exactly 12 trust partners - EPOCHS replaces GEOGRAPHY — Bitcoin halvings survived (unfakeable) - Collective slashing — Attack the network, lose your friends - Bitcoin Oracle — Real-time BTC block verification via multiple APIs

Version 4.1 additions: The Hal Humanity System: - Graduated Trust Model — Hardware (3 Apostles) → Social (6) → Time-Locked (12) - Time-Locked Identity Proofs — Bitcoin halving anchored, unfakeable - Sybil Resistance — Proving humanity, not just cryptographic identity

Version 4.2: Documentation and integration completeness.

Time cannot be bought, manufactured, or transferred—only spent. Humanity cannot be faked across Bitcoin halvings—only proven.

Table of Contents

---

1. Introduction

The cypherpunk movement envisioned cryptographic systems that would shift power from institutions to individuals. Bitcoin delivered on part of that promise—a monetary system without central authority. But Bitcoin's consensus mechanism, while elegant, contains a flaw that becomes more apparent with time: influence scales with capital.

Proof of Work requires specialized hardware. A participant with capital purchases ASICs and controls hashrate proportional to investment. Proof of Stake makes this explicit—stake coins, receive influence. Both systems work. Both systems concentrate power.

What the cypherpunks sought was not merely decentralized currency, but decentralized power. True decentralization requires a resource that cannot be accumulated, purchased, or transferred.

Time is that resource.

A node operating for 180 days accumulates the same influence whether owned by a billionaire or a student. This time is irreversible. It cannot be bought on an exchange. It cannot be rented from a cloud provider. It can only be spent—by existing.

1.1 The Quantum Threat

Current cryptographic systems face an existential threat: quantum computers. Shor's algorithm breaks ECDSA, RSA, and X25519. Conservative estimates place cryptographically-relevant quantum computers 10-15 years away.

Ɉ Montana implements quantum-resistant cryptography to ensure long-term security.

1.2 The Humanity Problem

Version 4.0 introduced EPOCHS—proof that a node survived Bitcoin halvings. But TIME proves existence, not uniqueness. An attacker can create 100 keypairs, wait 4 years, and control a coordinated network.

The Hal Humanity System (v4.1) solves this: proving humanity, not just cryptographic identity.

Named after Hal Finney (1956-2014), who received the first Bitcoin transaction and understood Sybil resistance before anyone else.

---

2. The Plutocracy Problem

All existing consensus mechanisms suffer from the same fundamental weakness: resource dependence creates plutocratic capture.

In Proof of Work, hash rate is purchasable. In Proof of Stake, the problem is structural. Delegated systems (DPoS) merely add intermediaries.

The solution is not to redistribute resources more fairly within these systems. The solution is to build consensus on resources that cannot be unequally distributed.

- **Time** passes for everyone at the same rate. This is physics.
- **Humanity** cannot be multiplied. One person = one human.

---

3. Verifiable Delay Functions

A Verifiable Delay Function (VDF) is a function that requires sequential operations to compute, but whose output can be efficiently verified.

3.1 Post-Quantum Construction: SHAKE256 VDF

```
Parameters:
  H = SHAKE256 (extendable-output function)
  T = iteration count

Compute(x, T):
  state₀ = x
  for i = 1 to T:
    stateᵢ = H(stateᵢ₋₁)
  return state□

Verify(x, y, proof, T):
  return STARK_verify(x, y, proof, T)
```

Quantum Resistance: SHAKE256 security relies only on hash function properties.

3.2 VDF Synchronization

Each VDF proof depends on the hash of the previous block:

```
VDF_input = SHA3-256(prev_block_hash || height)
VDF_output = Compute(VDF_input, T)
```

Pre-computation is impossible because `prev_block_hash` is unknown until the previous block is finalized.

---

4. Network Architecture

4.1 Dual-Layer Consensus

Layer 1 — Proof of History (PoH)

Sequential SHA3-256 chain for transaction ordering:

```
PoH_n = SHA3-256(PoH_{n-1} || tx_hash || timestamp)
```

Layer 2 — Proof of Time (PoT)

VDF checkpoints every 10 minutes provide finality.

4.2 Leader Selection

Every 10 minutes, one node is selected to produce a block using ECVRF:

```
VRF_input = SHA3-256(prev_block_hash || height)
(π, β) = VRF_prove(secret_key, VRF_input)

# Node is leader if: β < threshold × 2^256
```

---

5. The Five Fingers of Adonis

The Adonis reputation system uses five-dimensional assessment.

5.1 THUMB: TIME (50%)

The dominant factor. Saturates at 180 days of continuous uptime.

```
f_time(t) = min(t / K_TIME, 1.0)
# where K_TIME = 180 days
```

5.2 INDEX: INTEGRITY (20%)

No protocol violations. Decreases with misbehavior.

5.3 MIDDLE: STORAGE (15%)

Percentage of chain history stored.

5.4 RING: EPOCHS (10%) — v4.0

NEW: Bitcoin halvings survived. Unfakeable through time manipulation.

```python
def compute_epochs_score(node):
    epochs_survived = get_epochs_survived(node.first_btc_height)
    return min(epochs_survived / 4.0, 1.0)  # Saturates at 4
        halvings (16 years)
```

Why EPOCHS replaces GEOGRAPHY: - You can fake location with VPN. You cannot fake surviving a Bitcoin halving. - 210,000 BTC blocks (~4 years) = 1 epoch - Maximum score after 4 halvings (16 years of existence)

5.5 PINKY: HANDSHAKE (5%)

Mutual trust bonds via the 12 Apostles system.

---

6. The Twelve Apostles

New in v4.0: Each node chooses exactly 12 trust partners.

6.1 Design Philosophy

```
Trust Manifesto:
Before forming a handshake, ask yourself:

  Do I know this person?
  Not an avatar — a human.

  Do I trust them with my time?
  Willing to lose if they fail?

  Do I wish them longevity?
  Want them here for years?

  If any answer is NO — do not shake.
```

6.2 Why Twelve?

- **Dunbar's inner circle:** Humans maintain ~12-15 close relationships
- **Manageable responsibility:** You can truly know 12 people
- **Game-theoretic limit:** Prevents trust dilution

6.3 Seniority Bonus

Older nodes vouching for newer nodes carries more weight:

```
def compute_handshake_value(my_number, partner_number):
    if partner_number < my_number:
        # Older partner vouching for me
        return 1.0 + log10(my_number / partner_number)
    return 1.0

# Node #1000 shakes #50: value = 1 + log10(1000/50) = 2.30
# Node #1000 shakes #999: value = 1 + log10(1000/999) = 1.00
```

6.4 Collective Slashing

Attack the network, lose your friends:

```
ATTACKER_QUARANTINE_BLOCKS = 180_000  # ~3 years

# Penalties:
# - Attacker: TIME=0, INTEGRITY=0, quarantine
# - Vouchers: -25% integrity (those who vouched for attacker)
# - Associates: -10% integrity (those vouched by attacker)
```

All handshakes dissolved. Trust network damaged.

This isn't punishment—it's accountability.

---

7. The Hal Humanity System

New in v4.1: Proof of Human, Not Just Proof of Time.

7.1 The Sybil Problem

Montana v4.0 proves TIME (via EPOCHS). But TIME proves existence, not uniqueness.

Attack Scenario:

```
Year 0: Attacker creates 12 keypairs
Year 4: All 12 survive halving, EPOCHS=0.25
        Form mutual handshakes (no humanity check)
Result: One person controls full 12-Apostle network
```

Sybil Cost in v4.0: $0 (just wait)

7.2 The Hal Solution

Time-Locked Identity + Graduated Trust = Unfakeable Humanity Proof

Named after **Hal Finney** (1956-2014), who understood Sybil resistance before anyone else.

7.3 Graduated Trust Model

```
┌─────────────────────────────────────────────────────────────┐
│  TIER 3: TIME-LOCKED (Ultimate - 12 Apostles max)           │
│  ═══════════════════════════════════════════                │
│  • Survived 1+ Bitcoin halvings with valid commitment       │
│  • UNFAKEABLE - requires actual time passage                │
│  • Weight: 1.0                                              │
├─────────────────────────────────────────────────────────────┤
│  TIER 2: SOCIAL (Bridge - 6 Apostles max)                   │
│  ═══════════════════════════════════════                    │
│  • Built social graph through handshakes                    │
│  • Sybil cost: real human connections over time             │
│  • Weight: 0.6                                              │
├─────────────────────────────────────────────────────────────┤
│  TIER 1: HARDWARE (Bootstrap - 3 Apostles max)              │
│  ═══════════════════════════════════════                    │
│  • TPM/Secure Enclave/FIDO2 attestation                     │
│  • Sybil cost: physical device ($50-500)                    │
│  • Weight: 0.3                                              │
└─────────────────────────────────────────────────────────────┘
```

7.4 Time-Locked Identity Proofs

Core mechanism anchored to Bitcoin halvings:

```python
# COMMITMENT PHASE (at epoch N)
def create_identity_commitment(pubkey, secret, btc_block_hash):
    """
    Create commitment anchored to Bitcoin halving.
    secret must be stored securely by user.
    """
    commitment_hash = SHA3_256(pubkey + secret + btc_block_hash)
    return IdentityCommitment(
        pubkey=pubkey,
        commitment=commitment_hash,
        epoch=current_epoch,
        btc_height=epoch * 210_000
    )

# PROOF PHASE (at epoch N+1, after halving)
def create_time_locked_proof(commitment, secret,
        current_btc_hash):
    """
    Create ZK proof of commitment after halving.
    """
    # STARK proof that:
    # 1. I know secret such that commitment = SHA3(pubkey ||
        secret || old_hash)
```

```
    # 2. old_hash is from epoch N
    # 3. current_hash is from epoch N+1

    proof = STARK_prove(
        public_inputs=[commitment.commitment, current_btc_hash],
        private_inputs=[secret, commitment.btc_block_hash],
        statement="commitment_valid_and_epoch_passed"
    )
    return TimeLockProof(commitment=commitment, stark_proof=proof)
```

7.5 Sybil Economics

Creating N fake identities requires:

Tier	Sybil Cost per Identity
HARDWARE	$50-500 (physical device)
SOCIAL	Months/years (real connections)
TIME-LOCKED	4+ years (Bitcoin halving)

At Tier 3: 100 fake identities = 400 years of waiting.

This is the Hal Finney vision realized.

7.6 Network Evolution

```
Year 0-4:  Hardware + Social bootstrap (enough to launch)
Year 4+:   Time-Locked activates (first real proofs)
Year 8+:   Time-Locked dominates (2 halvings = high trust)
Year 16+:  Pure Time-Locked (Hardware/Social deprecated)
```

7.7 Hardware Attestation (Tier 1)

Three attestation methods supported:

TPM 2.0:

```
# Quote mechanism proves:
# 1. Device has genuine TPM
# 2. PCR values match expected state
# 3. Montana pubkey is bound via TPM key
```

Apple Secure Enclave:

```
# DeviceCheck/App Attest API proves:
# 1. Device has genuine Secure Enclave
# 2. Attestation key is hardware-bound
# 3. Montana pubkey is bound via enclave key
```

FIDO2/WebAuthn:

```
# FIDO2 authenticator proves:
# 1. Device has genuine FIDO2 authenticator
# 2. User verified via biometric/PIN
# 3. Montana pubkey is bound via credential
```

7.8 Social Graph Verification (Tier 2)

Custom social graph built from handshakes:

```
# Sybil detection via graph analysis
class SocialVerifier:
    def verify_social_proof(self, proof):
        # Check minimum connections
        if proof.connection_count < 3:
            return False, "Need 3+ connections"

        # Check clustering coefficient
        cc =
        self.graph.compute_clustering_coefficient(proof.pubkey)
        if cc > 0.8:  # Too clique-like
            return False, "Clustering too high"

        # Check temporal correlation
        tc = self.graph.get_temporal_correlation(proof.pubkey)
        if tc > 0.5:  # Batch created
            return False, "Temporal correlation too high"

        return True, "Social proof valid"
```

7.9 Handshake Integration

Humanity tier limits Apostle count:

```
def can_form_handshake(my_profile, target):
    # Tier-based limits
    max_allowed = my_profile.max_apostles  # 3, 6, or 12
    if len(my_apostles) >= max_allowed:
        return False, f"Tier {my_profile.humanity_tier} limited
        to {max_allowed}"

    # Humanity score check
    if target.humanity_score < 0.3:
        return False, "Target humanity score too low"

    return True, "Requirements met"
```

---

8. Anti-Cluster Protection

Defense against the “Slow Takeover Attack.”

8.1 Behavioral Correlation Detection

```python
class ClusterDetector:
    def compute_correlation(self, node_a, node_b):
        # Timing correlation (50% weight)
        timing_corr = count_simultaneous() / total_actions

        # Action distribution (30% weight)
        dist_corr = cosine_similarity(actions_a, actions_b)

        # Block height patterns (20% weight)
        height_corr = jaccard_similarity(heights_a, heights_b)

        return 0.5*timing + 0.3*dist + 0.2*height
```

8.2 Global Cluster Cap

No cluster can exceed 33% of network influence.

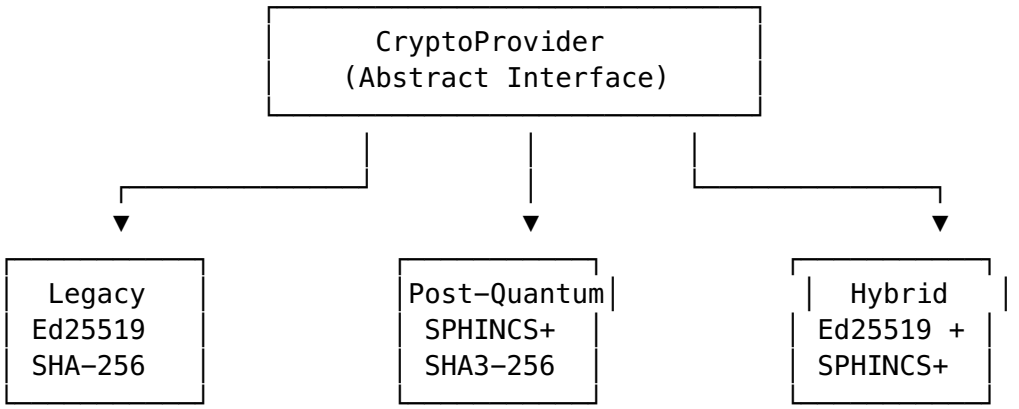
```python
MAX_CLUSTER_INFLUENCE = 0.33

def apply_cluster_cap(probabilities, clusters):
    for cluster in clusters:
        cluster_share = sum(prob[n] for n in cluster.members)
        if cluster_share > MAX_CLUSTER_INFLUENCE:
            reduction = MAX_CLUSTER_INFLUENCE / cluster_share
            for node in cluster.members:
                probabilities[node] *= reduction
    return probabilities
```

---

9. Post-Quantum Cryptography

Complete quantum-resistant cryptographic stack following NIST post-quantum standards.

9.1 Crypto-Agility Architecture



9.2 Algorithms

Function	Algorithm	Security
Signatures	SPHINCS+-SHAKE-128f	128-bit PQ
Hashing	SHA3-256	128-bit PQ
VDF	SHAKE256 + STARK	128-bit PQ
Key Exchange	ML-KEM-768	128-bit PQ

---

10. Attack Resistance Analysis

10.1 Attack Vector Matrix (v4.2)

Attack	Difficulty	Mitigation	Notes
Flash Takeover	IMPOSSIBLE	180-day saturation	
Slow Takeover	VERY HARD	Correlation + 33% cap	
Sybil via Keypairs	VERY HARD	Hal Humanity System	**NEW v4.1**
Fake Apostle Network	HARD	Humanity tier limits	**NEW v4.1**
Hardware Spoofing	HARD	Multiple attestation sources	**NEW v4.1**
Quantum Attack	IMPOSSIBLE	SPHINCS+, SHA3, SHAKE256	

10.2 Sybil Attack Cost Comparison

Version	Sybil Cost (100 fake nodes)
v3.x	$0 (just wait 180 days)
v4.0	$0 (just wait 4 years)
v4.1	**400 years** (100 × 4-year halvings)

10.3 51% Attack Requirements (v4.2)

To control majority weighted influence:

Requirement	v3.x	v4.2
Time	180 days	180 days
EPOCHS	N/A	4+ years
Humanity	N/A	Tier 3 proof (4+ years per identity)
Apostles	10	12 (requires Tier 3)

---

11. Known Limitations

11.1 Correlation Detection Evasion

Sophisticated attackers can add random delays (150ms+ jitter) to avoid timing correlation detection.

Mitigation: Hal Humanity System limits even undetected Sybils to Tier 1 (3 Apostles each).

11.2 Cold Start Problem

Time-Locked Identity requires 4 years for first proof.

Mitigation: Graduated Trust allows Hardware (Tier 1) and Social (Tier 2) bootstrap.

11.3 Hardware Attestation Spoofing

Determined attackers may spoof hardware attestation.

Mitigation: - Multiple attestation sources (TPM + Enclave + FIDO2) - Hardware tier limited to 3 Apostles - Transition to Time-Locked over 4 years

---

12. Network Protocol

12.1 Message Types

```
Type 0-15:  Standard (VERSION, BLOCK, TX, etc.)
Type 100-102: Noise Protocol handshake
Type 200: HUMANITY_PROOF      # NEW v4.1
Type 201: APOSTLE_HANDSHAKE   # NEW v4.0
Type 202: SLASHING_EVIDENCE   # NEW v4.0
```

12.2 Encryption: Noise Protocol

All peer connections use Noise Protocol Framework, XX pattern.

12.3 Bitcoin Oracle (v4.0)

Real-time BTC block verification:

```
BTC_APIS = [
    "https://blockstream.info/api",
    "https://mempool.space/api",
    "https://blockchain.info",
```

```
]

def get_btc_block_hash(height):
    """Get BTC block hash from multiple sources."""
    results = []
    for api in BTC_APIS:
        try:
            hash = fetch_block_hash(api, height)
            results.append(hash)
        except:
            continue

    # Require 2/3 consensus
    if len(results) >= 2:
        return most_common(results)
    return None
```

---

13. Privacy

All transactions use ring signatures and stealth addresses.

13.1 Privacy Tiers

Tier	Hidden	Fee Multiplier
T0	Nothing	1×
T1	Receiver	2×
T2	+ Amount	5×
T3	+ Sender	10×

13.2 Ring Signatures (LSAG)

Linkable Spontaneous Anonymous Group signatures hide sender among decoys.

13.3 Stealth Addresses

Each transaction generates a unique one-time address.

---

14. Emission Schedule

14.1 Supply Parameters

```
Name: J Montana
Symbol: J
Ticker: $MONT
```

```
Base unit: 1 Ɉ = 1 second
Total supply: 1,260,000,000 Ɉ (21 million minutes)
Halving interval: 210,000 blocks (~4 years)
Full emission: ~132 years
```

---

15. Implementation

15.1 Repository Structure (v4.2)

```
montana/
├── pantheon/
│   ├── prometheus/              # Cryptography
│   ├── athena/                  # Consensus
│   ├── hermes/                  # Networking + RHEUMA
│   ├── hades/                   # Storage
│   ├── plutus/                  # Wallet
│   ├── nyx/                     # Privacy
│   ├── adonis/                  # Reputation (Five Fingers)
│   ├── apostles/                # 12 Apostles + Slashing  (v4.0)
│   │   ├── handshake.py         # Handshake protocol
│   │   └── slashing.py          # Collective slashing
│   └── hal/                     # Humanity System  (v4.1)
│       ├── humanity.py          # Core verification
│       ├── hardware.py          # TPM/Enclave/FIDO2
│       ├── social.py            # Social graph
│       └── timelock.py          # Time-locked proofs
├── tests/
│   ├── test_integration.py      # 48 integration tests
│   ├── test_dag.py              # 48 DAG tests
│   ├── test_fuzz.py             # 27 fuzz tests
│   └── test_pq_crypto.py        # 56 PQ crypto tests
└── Montana_v4.2.md              # This document
```

15.2 Running a Node

```
pip install pynacl gmpy2 noiseprotocol
python node.py --run
```

---

16. Conclusion

16.1 What Ɉ Montana v4.2 Guarantees

1. **No instant takeover:** Minimum 180 days to reach maximum influence
2. **Cluster cap:** No coordinated group exceeds 33% influence
3. **Quantum resistance:** Signatures and VDF secure against quantum computers
4. **Sybil resistance:** Humanity proofs via graduated trust
5. **Time-locked identity:** 4-year Bitcoin halving anchors cannot be faked

6. **Collective accountability:** 12 Apostles + slashing creates real consequences

16.2 Sybil Resistance Evolution

Version	Sybil Cost	Mechanism
v1.0-v3.x	$0 (wait 180 days)	TIME only
v4.0	$0 (wait 4 years)	TIME + EPOCHS
v4.1+	**N × 4 years**	TIME + EPOCHS + HAL

Creating 100 fake identities = 400 years. This is the Hal Finney vision realized.

16.3 Final Statement

Ɉ Montana removes capital as the basis of influence. The system uses: - **Time** — cannot be purchased, accelerated, or concentrated - **Humanity** — cannot be multiplied across Bitcoin halvings

With quantum-resistant cryptography and the Hal Humanity System, these guarantees extend indefinitely into the future.

"Running bitcoin" — Hal Finney, January 2009

"Time is priceless. Humanity is sacred. Now both have cryptographic proof."

Ɉ

References

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008.

[2] D. Boneh et al., "Verifiable Delay Functions," CRYPTO 2018.

[3] NIST FIPS 202, 203, 205 — Post-Quantum Standards, 2024.

[4] H. Finney, "RPOW - Reusable Proofs of Work," 2004.

[5] R. Dunbar, "How Many Friends Does One Person Need?" 2010.

Appendix A: Constants Reference (v4.2)

```
# ============================================================
# ADONIS WEIGHTS (Five Fingers)
# ============================================================
WEIGHT_TIME = 0.50        # THUMB
WEIGHT_INTEGRITY = 0.20   # INDEX
WEIGHT_STORAGE = 0.15     # MIDDLE
```

```
WEIGHT_EPOCHS = 0.10      # RING (replaces GEOGRAPHY)
WEIGHT_HANDSHAKE = 0.05   # PINKY

# ============================================================
# EPOCHS (v4.0)
# ============================================================
HALVING_INTERVAL = 210_000  # Bitcoin blocks per epoch
MAX_EPOCHS_FOR_SATURATION = 4  # 16 years

# ============================================================
# 12 APOSTLES (v4.0)
# ============================================================
MAX_APOSTLES = 12
MIN_INTEGRITY_FOR_HANDSHAKE = 0.50
HANDSHAKE_COOLDOWN = 86400  # 24 hours

# ============================================================
# HAL HUMANITY SYSTEM (v4.1)
# ============================================================
# Tier limits
MAX_APOSTLES_HARDWARE = 3      # Tier 1
MAX_APOSTLES_SOCIAL = 6        # Tier 2
MAX_APOSTLES_TIMELOCKED = 12   # Tier 3

# Weights
HUMANITY_WEIGHT_HARDWARE = 0.3
HUMANITY_WEIGHT_SOCIAL = 0.6
HUMANITY_WEIGHT_TIMELOCKED = 1.0

# Minimum for handshake eligibility
HANDSHAKE_MIN_HUMANITY = 0.3

# ============================================================
# SLASHING (v4.0)
# ============================================================
ATTACKER_QUARANTINE_BLOCKS = 180_000  # ~3 years
VOUCHER_INTEGRITY_PENALTY = 0.25      # -25%
ASSOCIATE_INTEGRITY_PENALTY = 0.10    # -10%

# ============================================================
# ANTI-CLUSTER
# ============================================================
MAX_CORRELATION_THRESHOLD = 0.7
MAX_CLUSTER_INFLUENCE = 0.33
MIN_NETWORK_ENTROPY = 0.5
```

Appendix B: Version History

Version	Date	Changes
1.0	Dec 2025	Initial specification

Version	Date	Changes
2.0	Dec 2025	Five Fingers of Adonis, anti-cluster
3.0	Dec 2025	Post-quantum cryptography
3.1	Dec 2025	Network security hardening
4.0	Dec 2025	**12 Apostles, EPOCHS dimension, collective slashing, Bitcoin Oracle**
4.1	Dec 2025	**Hal Humanity System: graduated trust, time-locked identity, Sybil resistance**
4.2	Dec 2025	**Documentation and integration completeness**