DRAFT — CONFIDENTIAL

TOWARD A FULLY ATTESTED OWNERSHIP SUBSTRATE

Provable Continuity Infrastructure for Institutional Digital Assets

Layers 1–5: Legal Ownership, Attested State, Governance, AI Governance, Post-Quantum Survivability

JOINT TECHNICAL PAPER — DRAFT V2

PayKing Corporation / KoreaGemLab × H33.ai, Inc.

Seong-Ho Lee × Eric Beans

May 2026

Note on Implementation Details: Certain implementation specifics presented in this paper are intentionally omitted due to pending patent protection (KR 10-2026-0068484). All Solidity code shown reflects interface-level architecture only.

ABSTRACT

Provable Continuity for Institutional Digital Assets

When a $12 million wire transfer is disputed eighteen months after execution, the institution must answer a simple question: what happened? In most systems today, the answer depends on whether logs survived, whether vendors cooperated, whether databases remained intact, and whether the AI model that made the compliance decision still exists in its original form. When any of these assumptions fail — and they fail precisely when the evidence matters most — the institution cannot prove what happened.

This paper presents a **provable continuity infrastructure** — a five-layer architecture that makes institutional memory survive failure. At its foundation is **Layer 1: the legal ownership substrate** — the IdentHash architecture that commits legal identity on-chain without exposing personal data, establishing the evidentiary foundation upon which all subsequent layers depend. Without provable ownership continuity, no attestation, governance proof, or AI decision has a verified subject. Layer 1 is the ownership substrate. Layers 2 through 5 — attested state, governance, AI governance, and post-quantum survivability — build upon it.

The architecture is designed to address the possession and transfer conditions articulated in the proposed Digital Asset Market Clarity Act of 2025 (CLARITY Act, H.R. 3633) at the protocol level, not through administrative procedure. Final determination of legal classification remains with relevant regulatory and judicial authorities.

The ownership object is the proof. They are inseparable.

Provable continuity — continuity of truth across time, systems, vendors, and cryptographic eras — is the category this paper defines. The five layers that follow are the architecture that delivers it.

THE CORE PROBLEM

Why Continuity Matters

Modern systems cannot prove what actually happened once trust breaks.

This is not a cryptographic observation. It is an operational reality that surfaces in every institutional dispute, regulatory investigation, and fraud inquiry. The moment a system's integrity is questioned, the

assumptions that sustained its operation collapse simultaneously:

Infrastructure Assumptions

Databases survive. Vendors survive. Logs remain intact. Chains remain stable. Keys remain secure.

Operational Assumptions

AI systems remain reproducible. Policy versions are tracked. Approval chains are complete. Records are unmodified.

Trust Assumptions

Counterparties are honest. Intermediaries are available. Auditors have access. Platform operators cooperate.

Institutional disputes occur precisely when these assumptions fail. When a wire is challenged. When an AI decision is questioned. When a regulator investigates. When a court demands replay. When fraud is alleged. When infrastructure is unavailable. When vendors are gone.

In each case, the question is the same: **can the original truth be independently reconstructed?**

Most systems cannot independently reconstruct: the original data, the governing policy, the AI decision, the approval chain, or the exact system state at the moment in question. This is a *continuity failure* — not a security failure, not a compliance failure, but a fundamental inability to prove what happened.

CANONICAL INCIDENT — THE WIRE THAT CANNOT BE PROVEN

An AI-assisted compliance system approves a $12M institutional wire transfer. Six months later, the transfer is disputed. The institution must prove what happened.

WITHOUT CONTINUITY INFRASTRUCTURE

- SIEM logs incomplete (90-day retention expired)
- AI model updated twice since the decision
- Compliance vendor changed platforms

WITH FULLY ATTESTED CONTINUITY

- Original ownership state reconstructed (Layer 1)
- State transition attestation verified (Layer 2)
- Governance approval proof replayed (Layer 3)

• Screenshots disputed as fabricated
• Database records modified during investigation
• 3-6 week forensic reconstruction attempt

Institution cannot prove what the AI saw, why the decision occurred, or whether the record changed.

• AI decision independently reproduced (Layer 4)
• Tampering detected cryptographically
• Offline proof succeeds — no vendor, no platform

Every action reconstructed. Every policy replayed. Every record verified. Independent of all infrastructure.

H33 turns system behavior into cryptographic evidence.

Not attestations. Not hashes. Not signatures. *Evidence* — of what happened, what policy executed, what model ran, what data entered, what state existed, and whether the record changed. That is the product. The five layers that follow are the architecture that produces it.

WHAT THIS SYSTEM DOES NOT PROVE

This system does not prove that the AI was *correct*. It does not prove that the data was *truthful*. It does not prove that no human collusion occurred. It does not prove that the policy was *well-designed*. It does not prove intent.

What it proves is: **what happened, what policy executed, what model ran, what data entered, what state existed, and whether the record changed.**

That precision is the source of its evidentiary value. It does not overstate its guarantees. It proves exactly what a court, regulator, or auditor needs: a reconstructable, tamper-evident, independently verifiable record of system behavior.

This canonical incident — the disputed wire transfer — recurs throughout this paper. Every layer maps back to the same event. Layer 1 proves who owned the asset. Layer 2 proves the state transition was

legitimate. Layer 3 proves the governance approval was valid. Layer 4 proves the AI decision is reproducible. Layer 5 proves the evidence survives decades and infrastructure changes.

One incident. Five layers. Complete evidentiary reconstruction.

INSTITUTIONAL IMPACT

What This Replaces

Every institution currently relies on the same set of mechanisms to establish what happened after a dispute, investigation, or audit. These mechanisms share a common property: they depend on trust in the system that produced them. When that trust is questioned — which is precisely when the evidence matters most — the mechanisms fail.

TRADITIONAL STACK		FULLY ATTESTED CONTINUITY
SIEM logs	→	Cryptographic replay
Screenshots	→	Deterministic reconstruction
Vendor attestations	→	Independent verification
Policy PDFs	→	Policy-bound execution
Human testimony	→	Signed state transitions
Audit sampling	→	Full event continuity
Trust	→	**Proof**

The distinction is not incremental. SIEM logs record what the system chose to log. Cryptographic replay reconstructs what actually happened. Screenshots capture what someone chose to capture. Deterministic reconstruction reproduces the original state independently. Vendor attestations depend on the vendor existing. Independent verification depends on mathematics.

THE MISSING WIRE

A regulator asks six questions about a $12M wire transfer approved by an AI compliance system eighteen months ago:

Who approved the wire?	What sanctions list was active?
What model version executed?	Did the beneficiary change after approval?
Was the approval chain modified?	Were the vendor logs altered?

TRADITIONAL ANSWER

Weeks of forensic reconstruction. SIEM logs past retention window. Compliance vendor migrated platforms. AI model updated three times. Internal screenshots disputed as fabricated. Database audit trail shows unexplained modifications. Two conflicting internal reports produced.

The institution cannot answer any of the six questions with certainty.

ATTESTED CONTINUITY ANSWER

Replay the attestation chain. Verify governance signatures — three board members signed, quorum met, policy v2.4 active. Verify model ID — model hash matches the attested version at decision timestamp. Verify beneficiary — IdentHash unchanged since creation. Verify tampering — SHA3-256 hash chain intact. Verify sanctions list — attestation binds the exact list version used.

All six questions answered in minutes. Independently. Offline. No vendor required.

What Changes After Implementation

Regulatory Defensibility

Every compliance decision is independently reproducible. The regulator verifies the proof — they do not trust the institution's account of events.

Operational Survivability

Vendor disappears, platform migrates, database corrupts — the attestation chain survives independently. No single point of evidentiary failure.

Dispute Resolution

Courts receive cryptographic evidence, not testimony. The proof is deterministic — same inputs produce same verification result for any party.

Audit Survivability

Full event continuity replaces sampling. Every action in the lifecycle is attested, cached, and replayable — not a statistical subset.

Insurer Evidence

Cyber insurers verify the exact system state at the moment of breach — not what was reported after. Claims become provable, not argued.

Dependency Removal

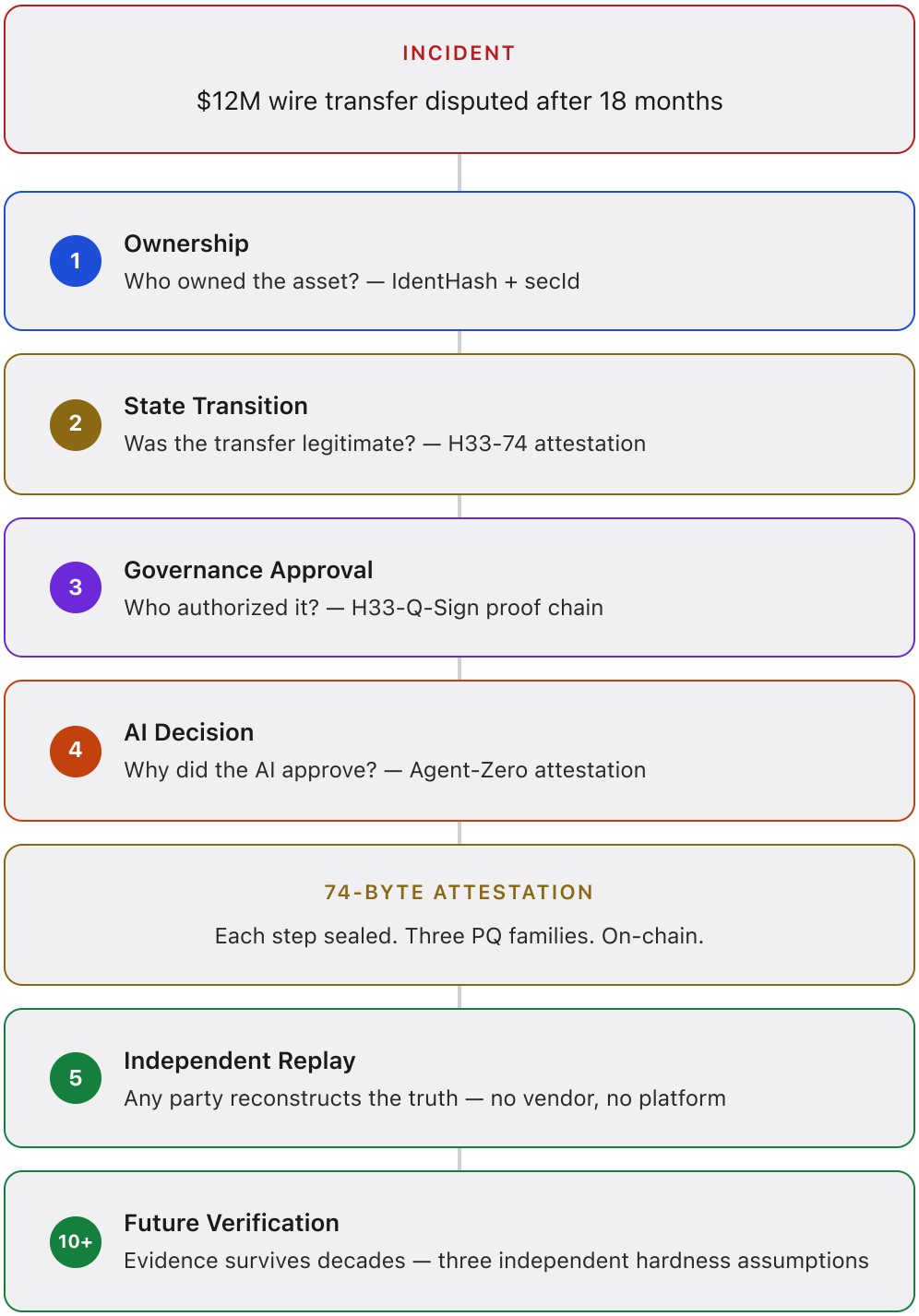
No intermediary is required for verification. No vendor must survive. No platform must cooperate. The proof is self-contained and independently executable.

The category is not attestation. It is not post-quantum cryptography. It is not AI governance. The category is **provable continuity** — continuity of truth across time, systems, vendors, and cryptographic eras. Everything in this paper serves that single objective.

ARCHITECTURE OVERVIEW

The Continuity Chain

One incident. Five layers. Complete evidentiary reconstruction.



Institutional memory that survives failure.

LAYER 1 — THE OWNERSHIP SUBSTRATE

Legal Ownership Foundation

> **Layer 1 is the evidentiary foundation of the entire architecture. Every attestation (Layer 2), every governance proof (Layer 3), every AI decision (Layer 4), and every long-term verification (Layer 5) requires a verified ownership subject. Without Layer 1, there is nothing to attest, nothing to govern, and nothing to verify. The ownership substrate is not one layer among five. It is the layer upon which the other four exist.**

Contributed by Seong-Ho Lee, Founder & CEO, PayKing Corporation / KoreaGemLab.
Patent: KR 10-2026-0068484.

1.1 The Structural Gap

Every major platform — Franklin Templeton BENJI, Ondo Finance OUSG, Superstate USTB, Securitize, Tokeny — shares the same fundamental problem: the token lives on-chain, the legal owner lives off-chain. These two records have never been unified in a single, self-verifying, intermediary-independent ownership object.

The consequences are structural:

- Legal ownership requires external system query
- Proof is contingent on intermediary availability
- Key loss severs legal title from the blockchain record
- Inheritance requires intermediary cooperation
- CLARITY Act conditions for direct, person-to-person, intermediary-free possession remain unmet

1.2 Comparative Analysis

Three ownership models compared:

DIMENSION	WALLET ADDRESS	ERC-3643 / TRANSFER AGENT	IDENTHASH (SECID-BASED)
Identity Binding	Off-chain — no on-chain identity	Off-chain registry lookup	On-chain hash-committed via PII_commitment
Intermediary Dependency	None, but legally void	Required per transaction	Optional — not required per transaction
Key Loss Recovery	Ownership lost permanently	Intermediary assistance required	secId survives key loss
Past Ownership Proof	No mechanism	Requires external registry query	Merkle proof at any past block

1.3 The Root Cause

The root cause is a trilemma. Three requirements must be satisfied simultaneously, and existing approaches satisfy at most two:

1

On-Chain Identity

Legal identity bound on-chain without exposing personal data

2

Intermediary Independence

Ownership proof valid when intermediaries are removed

3

Historical Verifiability

Record verifiable at any past point without external queries

IdentHash resolves all three simultaneously. The mechanism: hash-commit identity on-chain (satisfies 1), embed secId inside the ownership object itself (satisfies 2), and anchor every state into a Merkle tree with block-level granularity (satisfies 3).

2.1 The Central Distinction: Authentication vs. Ownership

Current tokenized systems make a category error: they assume that whoever controls the private key *is* the owner. Authentication and ownership are conflated. IdentHash separates them.

DIMENSION	CURRENT MODEL	IDENTHASH MODEL
Authentication	Private key	Private key
Ownership	Private key	`secId` + `PII_commitment`
After Key Loss	Ownership lost	Ownership intact — secId persists
In Legal Dispute	Key controller prevails	Legal entity proven by PII_commitment
Inheritance	Requires key transfer	Legal succession — new secId issued

2.2 Why secId Is Not Another Central Registry

A Transfer Agent registry must be queried per transaction — every transfer depends on the registry being available and cooperative. `secId` is fundamentally different: it is a **one-time issuance**. The registry issues the `secId`, it is embedded inside the IdentHash, and then the registry exits the transaction path entirely. No per-transaction dependency. No availability requirement for settlement.

2.3 secId Issuance Process

```
// secId Issuance — one-time process

// Step 1: Investor submits KYC materials
investor → registry.submitKYC(legal_name, ID_number, address)

// Step 2: Registry verifies identity
registry.verifyIdentity(investor)

// Step 3: Generate cryptographic salt
salt = random_bytes(32)

// Step 4: Compute PII commitment — no plaintext stored on-chain
PII_commitment = H(legal_name || ID_number || address || salt)

// Step 5: Issue secId
secId = registry.issueSecuritiesId(investor, PII_commitment)

// Step 6: Binding stored — registry exits transaction path
binding = { secId, PII_commitment, issued_at }

// After issuance: registry is NOT required for transfers
```

3. IdentHash Architecture

The architecture below shows how ownership continuity becomes machine-verifiable at the protocol layer. Each interface represents an on-chain operation that any party can independently audit.

Hash Construction

```
// IdentHash — legal ownership anchor
IdentHash = H(asset_id || S_bundle || R_bundle || amount || nonce)

// Sender bundle — wallet + legal identity + privacy commitment
S_bundle  = (wallet_address || securities_id || PII_commitment)

// Receiver bundle — same structure
```

```
R_bundle  = (wallet_address || securities_id || PII_commitment)

// PII commitment — no plaintext on-chain
PII_commitment = H(legal_name || ID_number || address || salt)

// secId = legal entity anchor
// Survives key loss, chain migration, signature algorithm changes
```

Smart Contract Interface

```
// IIdentHashTransfer — escrow-based ownership transfer

interface IIdentHashTransfer {

    // Create escrow with both parties' IdentHashes
    function createEscrow(
        bytes32 senderIdentHash,
        bytes32 receiverIdentHash,
        uint256 amount,
        address tokenContract
    ) external returns (uint256 escrowId);

    // Manager confirms — governance proof attached here
    function confirmByManager(
        uint256 escrowId,
        bytes32 governanceProofHash
    ) external;

    // Settle escrow — atomic DvP execution
    function settleEscrow(
        uint256 escrowId
    ) external;

    // Verify ownership at any point
    function verifyOwnership(
        bytes32 identHash,
        bytes32[] calldata merkleProof
    ) external view returns (bool);
```

```
    // Freeze secId — court order or fraud detection
    function freezeSecId(
        bytes32 secId,
        bytes32 courtOrderHash
    ) external;
}
```

Events

```
// Contract events — on-chain audit trail

event OwnershipTransferred(
    bytes32 indexed identHash,
    bytes32 indexed fromSecId,
    bytes32 indexed toSecId,
    uint256 amount,
    uint256 blockNumber
);

event GovernanceProofAttached(
    uint256 indexed escrowId,
    bytes32 governanceProofHash,
    address approver
);

event EscrowSettled(
    uint256 indexed escrowId,
    bytes32 senderIdentHash,
    bytes32 receiverIdentHash,
    bytes32 h33Anchor
);

event SecIdFrozen(
    bytes32 indexed secId,
    bytes32 courtOrderHash,
    uint256 frozenAt
);
```

3.4 CLARITY Act Enforcement

The Digital Asset Market Clarity Act of 2025 (CLARITY Act, H.R. 3633) defines four conditions for digital asset possession and transfer. IdentHash is designed to address each condition articulated in the proposed CLARITY Act:

CLARITY CONDITION	REQUIREMENT	IDENTHASH RESPONSE
Exclusively possessed	Only one person can possess at a time	IdentHash uniquely binds `secId` to asset at each block — Merkle-provable
Person to person	Transfer directly between persons	`S_bundle` → `R_bundle` — legal entities on both sides via `secId`
Without intermediary	No intermediary required for transfer	`settleEscrow` executes on-chain — `secId` embedded, registry not queried
Recorded on DLT	Transfer recorded on distributed ledger	IdentHash + Merkle root committed on-chain per block

4. Merkle-Provable Ownership

Merkle Leaf Construction

```
// Each ownership event produces a Merkle leaf
leaf_node = H(identHash || timestamp || block_number)

// Leaves aggregated into per-batch Merkle tree
// Root committed on-chain — immutable after block confirmation
```

Past Ownership Proof

1. Retrieve the IdentHash for the asset at the target block number
2. Compute the Merkle leaf: `H(identHash || timestamp || block_number)`
3. Retrieve the Merkle proof (sibling hashes) from the on-chain root
4. Verify the proof against the committed root — ownership at that block is proven or disproven

Chain-agnostic by design. The Merkle structure does not depend on Polygon-specific features. Any chain that supports hash commitment and event logs can host IdentHash ownership proofs.

5. Recovery Architecture

Five failure scenarios compared between wallet-based ownership and IdentHash/secId:

SCENARIO	WALLET-BASED OUTCOME	IDENTHASH/SECID OUTCOME
Key loss	Ownership permanently lost — no recovery path	`secId` persists — new wallet bound via re-issuance
Death / inheritance	Requires key discovery or loss accepted	Legal succession — new `secId` issued to heir via probate
Court freeze order	Cannot enforce without custodian cooperation	`freezeSecId` enforced on-chain — `SecIdFrozen` event
Fraud / theft	Thief controls key = thief controls asset	`PII_commitment` proves original owner — legal recourse intact
Chain migration	Ownership proof tied to original chain state	IdentHash migrates — `secId` and Merkle history follow

6. CLARITY Act Alignment

CLARITY CONDITION	IDENTHASH DESIGN RESPONSE
Exclusively possessed	IdentHash uniquely binds one `secId` (one legal entity) to one asset position. Merkle tree enforces exclusivity — no double-ownership possible at any block height.
Person to person	`S_bundle` contains sender's legal identity (`secId` + `PII_commitment`). `R_bundle` contains receiver's. Both are legal persons, cryptographically bound.
Without intermediary	`settleEscrow` executes atomically on-chain. `secId` is already embedded in the IdentHash — no registry query at settlement time. The intermediary issued `secId` once, then exited.
Recorded on DLT	Every IdentHash, every Merkle root, every governance proof hash, every `EscrowSettled` event — all committed on distributed ledger. Independently verifiable.

7. H33 Integration Points

Layer 1 defines three explicit interfaces where H33 infrastructure binds to the IdentHash escrow flow:

```
// Three H33 integration points in the IdentHash escrow lifecycle

// Point 1: Governance proof at escrow approval
confirmByManager(escrowId, governanceProofHash)
    → H33-QSign: PQ multi-party governance proof
    → governanceProofHash = Q-Sign quorum output
    → GovernanceProofAttached event emitted

// Point 2: PQ attestation at settlement
settleEscrow(escrowId)
    → H33-74: 74-byte PQ attestation
    → h33Anchor (32 bytes) committed on-chain
    → EscrowSettled event includes h33Anchor
```

```
// Point 3: Signature upgrade path
Signature layer
    → H33 3-Family PQ: ML-DSA + FALCON + SLH-DSA
    → Replaces ECDSA at attestation layer
    → secId and IdentHash unaffected by algorithm change
```

8. Foundation for Layer 2

Layer 1 is the foundation. Every subsequent layer depends on it:

Layer 2	→	Cannot attest state transitions without Layer 1 ownership records to attest
Layer 3	→	Requires Merkle-provable ownership to validate governance authority
Layer 4	→	Requires self-verifying ownership record for AI compliance evaluation
Layer 5	→	Requires Layer 1 as the verified starting state for long-term survivability

What Layer 1 Cannot Prove Alone

Layer 1 answers *who owns this, and when did they own it*. It does not answer:

- **Governance legitimacy** — was the approval process valid? (Layer 3)
- **Organizational mandate** — did the approver have authority? (Layer 3)
- **AI authorization** — was the agent acting within its parameters? (Layer 4)
- **Decision replayability** — can the compliance decision be independently reproduced? (Layer 4)
- **Cryptographic survivability** — will the proofs hold in 10–30 years? (Layer 5)

Layer 1 answers: *who owns this, and when did they own it.*
Layer 2 answers: *was the process that created this ownership state legitimate.*
Both are required for institution-grade tokenized systems.

LAYER 2

Attested State Continuity

H33-74 + H33-Upstream. Layer 1 proves ownership. Layer 2 proves the legitimacy of every state transition that produced that ownership.

What Layer 2 Must Prove

- Every state transition is cryptographically bound to the prior state
- The transition was authorized under the correct policy
- The attestation survives infrastructure boundaries
- Verification does not depend on trusting intermediaries
- The attestation chain is independently replayable

H33-Upstream — Provenance Binding at Creation

When IdentHash is first committed (asset creation), H33-Upstream binds a 202-byte provenance commitment to the same event. The provenance commitment captures: asset classification, regulatory jurisdiction, issuer identity, compliance framework, and creation timestamp. This commitment is referenced by the IdentHash `asset_id` field. Every subsequent state transition references this origin.

Append-Only Claim Types

CLAIM TYPE	TRIGGER	IDENTHASH EVENT
`References`	New ownership transfer	`OwnershipTransferred`
`Supersedes`	secId re-issuance (key loss recovery)	Key rotation
`Corrects`	Regulatory amendment to classification	Amendment event
`Retracts`	Asset delisting	Preserved audit history

H33-74 — Attestation Per State Transition

At `settleEscrow`: H33-74 produces a 74-byte attestation (32 bytes on-chain as `h33Anchor`, 42 bytes off-chain). The attestation commits: IdentHash value, fromSecId, toSecId, governance proof hash, timestamp, and policy version. Domain separator: `0x50` (WalletTransaction) for token transfers.

Substrate Layout (58 bytes — signing payload)

version	comp_type	fhe_commitment	timestamp	nonce
[0]	[1]	[2..34]	[34..42]	[42..58]
1 byte	1 byte	32 bytes — SHA3-256	8 bytes BE	16 bytes

```
// Signing message derivation
signing_message = SHA3-256(substrate_bytes)  // 58 bytes → 32 bytes

// Distillation: ~21 KB raw signatures → 74 bytes persistent
//   32 bytes on-chain (h33Anchor) + 42 bytes off-chain
//   Not compression. Distillation preserves independent verifiability.
```

Three Independent PQ Signature Families

ML-DSA-65

Standard: FIPS 204
Level: 3
Public Key: 1,952 B
Signature: 3,309 B

MLWE Lattice

FALCON-512

Standard: Draft FIPS 206
Level: 1
Public Key: 897 B
Signature: ~666 B

NTRU Lattice

SLH-DSA-SHA2-128f

Standard: FIPS 205
Level: 1
Public Key: 32 B
Signature: 17,088 B

Hash-based (Stateless)

Relationship to Layer 1 Merkle Tree

The Layer 1 Merkle root aggregates all leaves in a settlement batch. The H33-74 attestation attaches to the same Merkle root. The Merkle proof proves ownership at time T; the H33-74 attestation proves the state transition at time T was legitimate, authorized, and policy-compliant. Together: ownership + legitimacy at any past point in time.

Verification Protocol

Any party — regulator, auditor, counterparty, court — can execute these six steps independently to verify a state transition. No API key. No platform access. No vendor cooperation required.

1. Deserialize 58-byte substrate from attestation payload
2. Validate domain separator — `0x50` for token transfers
3. Check timestamp within acceptable window
4. Check nonce not replayed (anti-replay enforcement)
5. Compute `signing_message = SHA3-256(substrate_bytes)`
6. Verify ML-DSA-65 detached signature (standard path).
   Optional: verify FALCON-512 + SLH-DSA for full three-family audit.

Reference implementation: `github.com/H33ai-postquantum/h33-74-verifier` — open source, standalone Rust binary, no API key required.

LAYER 3

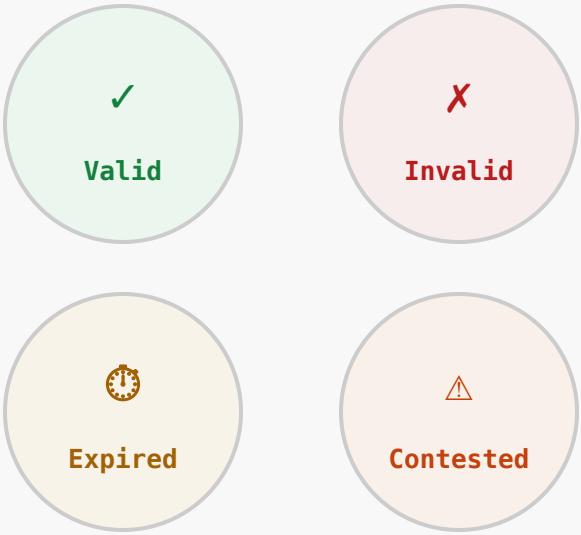
Governance Attestation

H33-Q-Sign. Layer 1 identifies the legal entity. Layer 2 attests the state transition. Layer 3 proves the governance action that authorized the transition was legitimate.

Integration Point

At `confirmByManager` in the IdentHash escrow flow, `governanceProofHash` is the H33-Q-Sign output. Stored on-chain via `GovernanceProofAttached` event. The governance proof is cryptographically bound to the specific escrowId, IdentHash, policy version, quorum of signers, and timestamp of authorization.

Four-State Validation Machine



Valid	→	Expired	*time elapsed*
Valid	→	Contested	*dispute filed by authorized party*
Contested	→	Valid	*dispute resolved, original upheld*

```
Contested    →  Invalid        dispute resolved, original overturned
```

Amendment Protocol

When a governance action is contested, the following protocol preserves auditability:

1. Original proof is NOT modified
2. Contested marker appended to original commitment
3. New proof chain created for dispute resolution
4. Final determination recorded as new governance action
5. Full history preserved — nothing overwritten
6. Auditor reconstructs complete governance chain

Multi-Party Signing

Each approver signs with their own PQ key pair. Q-Sign aggregates individual signatures into a quorum proof. Quorum threshold is policy-configurable (2-of-3, 3-of-5, etc.). Each individual signature is independently verifiable. The quorum proof is what gets committed as `governanceProofHash`.

What Q-Sign Prevents

FAILURE MODE	DESCRIPTION
Unverifiable logs	Platform claims approval occurred but cannot prove it cryptographically

FAILURE MODE	DESCRIPTION
Silent overrides	Administrator modifies governance decision after the fact without verifiable trail
Missing approvals	Required sign-off absent but transaction proceeded anyway

Relationship to Layer 1

Q-Sign governance proofs reference IdentHash ownership records. A governance action is only valid if the approver's `secId` has the required organizational authority. Authority itself can be Merkle-proved at the governance event timestamp using Layer 1 ownership continuity.

THE CONCEPTUAL BRIDGE

Once ownership, authority, and state transitions become attestable, autonomous systems inherit the same continuity model. AI decisions are no longer external to governance — they become governance events themselves, subject to the same replay, attestation, and independent verification as human actions. The wire transfer approved by an AI compliance engine is not a different category of event. It is the same ownership state transition, authorized by the same governance framework, attested by the same 74-byte proof. The only difference is that the decision-maker is a machine — and Layer 4 ensures that the machine's decision is as permanently reconstructable as the human's.

LAYER 4

AI / Agent Governance

The AI decision becomes evidence.

H33-Agent-Zero + Cachee. Layers 1–3 handle human-initiated actions. Layer 4 makes AI and autonomous agent actions attestable and replayable under the same ownership and governance framework.

The institutional question is not whether AI can make compliance decisions. It already does. The question is: **can the institution prove why the AI made that decision eighteen months from now, when the model has changed, the vendor has migrated, and the regulator is asking?**

Without Layer 4, the answer is no. AI decisions are ephemeral — they exist in memory during execution and vanish. The model updates. The training data shifts. The policy version changes. Six months later, the institution cannot reproduce the original decision, cannot prove which model ran, cannot verify which policy was active, and cannot demonstrate that the input data was unchanged.

Layer 4 makes the AI decision permanent, reproducible, and independently verifiable. **The AI decision becomes evidence** — cryptographic evidence that binds the input, the model, the policy, and the output into a single attestation that can be replayed years later by any party without trusting the institution, the vendor, or the AI system itself.

RETURNING TO THE WIRE

The $12M wire from our canonical incident was approved by an AI compliance system. Eighteen months later, the regulator asks: what model version executed? Layer 4 answers: model hash `0x7a3f...`, attested at the moment of execution, bound to policy version 2.4, input commitment verified against the original encrypted data. The AI decision is not a log entry. It is cryptographic evidence.

Why This Matters for IdentHash

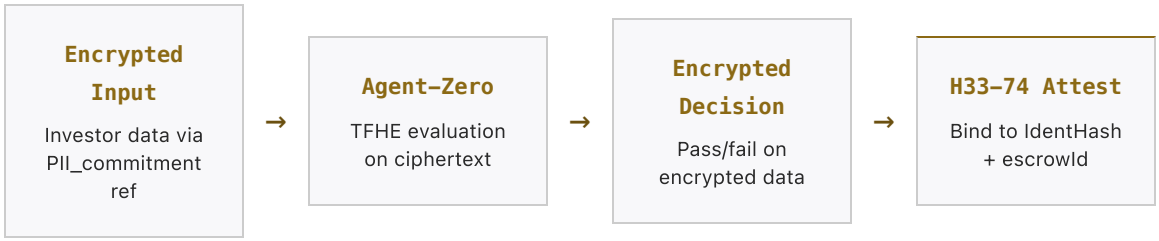
- Autonomous agents will initiate transfers, approve compliance, and execute governance actions

- Without attestation, agent actions are unverifiable — the institution cannot prove what the AI did or why
- Without replayability, agent decisions cannot be audited — the original computation is gone
- The CLARITY Act's "person to person" condition requires proof that a human or authorized agent acted within its mandate

H33-Agent-Zero — Encrypted Compliance Decisions

Agent-Zero uses TFHE programmable bootstrapping to evaluate compliance policies on encrypted data. Every AND gate refreshes noise, enabling unlimited circuit depth. The system never decrypts the data it evaluates. Decision finality: the encrypted output IS the binding decision. The institution can prove why the AI made the decision years later — because the decision, the model, the policy, and the input are all cryptographically bound at the moment of execution.

Decision Flow



Three Confidence Modes

HardClassification

Argmax output. Class label only. Zero information leakage beyond the binary decision. Designed for regulatory pass/fail determinations.

ThresholdProof

Single bit: above or below threshold. No score exposure. Designed for limit checks and accreditation verification.

CustomerDecrypt

Full score vector released to authorized keyholder only. Designed for internal risk assessment where the data owner controls disclosure.

Performance

METRIC	VALUE	CONDITION
Encrypted decisions	`768 TPS`	8-bit comparisons
Per decision	`4.1 ms`	Single evaluation
Hardware	ARM Graviton4	No GPU required

Cachee — Replayable Audit Infrastructure

Every H33-74 attestation is cached with a PQ-attested receipt. A SHA3-256 hash chain links sequential operations. Tamper-evident: any modification breaks the chain. Deterministic replay: same inputs produce the same proof chain. No trust in the replaying party is required.

Replay Model for AI Actions

1. Agent-Zero receives encrypted compliance request
2. Evaluates policy on ciphertext (TFHE programmable bootstrapping)
3. Produces encrypted decision
4. Decision attested via H33-74 (binds: input commitment, model ID, policy version, decision output, timestamp)
5. Attestation cached in Cachee with hash chain link
6. Auditor replays: retrieves attestation chain, verifies each independently, confirms policy version, confirms model ID, confirms decision was deterministic

What Layer 4 Answers That Layers 1–3 Cannot

- Was the AI acting within its authorized parameters?
- Can the AI's decision be independently reproduced?

- Did the AI have access to data it should not have? (FHE ensures it never sees plaintext.)
- Was the AI's policy version the correct one at that timestamp?

LAYER 5

Post-Quantum Survivability

Everything in Layers 1–4 must survive cryptographic transitions, chain migrations, and infrastructure changes over institutional time horizons (10–30+ years).

Signature Upgrade Path

IdentHash is hash-function and signature-family agnostic. H33 3-family PQ signatures replace ECDSA at the attestation layer. Existing ownership records require no reissuance. `secId` persists across signature algorithm changes. The upgrade is at the attestation layer, not the ownership layer.

Three Independent Hardness Assumptions

HARDNESS ASSUMPTION	FAMILY	IMPLICATION
MLWE Lattice	`ML-DSA-65`	Breaking lattice problems does not affect hash-based or NTRU families
NTRU Lattice	`FALCON-512`	Independent mathematical structure from MLWE
SHA2-256 Preimage	`SLH-DSA`	Hash-based; no algebraic structure to exploit

Breaking one family does not compromise the attestation. All three must be broken simultaneously for forgery.

Time Horizons

2026
All three families hold. NIST standards finalized.

10-Year
All three families expected to hold.

30-Year
If one weakens, two remain independently secure.

Crypto-agility: the substrate version field (byte 0) enables algorithm upgrade without breaking the existing attestation chain.

Cross-System Verification Matrix

TRANSITION	ATTESTATION SURVIVES	MECHANISM
Chain migration	**VERIFIED**	Attestations are chain-agnostic; IdentHash migrates, attestations follow
Custody changes	**VERIFIED**	Proof follows the asset, not the custodian
Key rotation	**VERIFIED**	`secId` persists; new keys re-bound to existing identity
Platform interop	**VERIFIED**	Open verification algorithm; no vendor lock-in
Intermediary removal	**VERIFIED**	No platform trust required; all proofs independently checkable

Bitcoin Mainnet Anchoring

H33-74 commitments are anchored to Bitcoin via `OP_RETURN`. 32 bytes on Bitcoin mainnet provide a permanent, immutable timestamp independent of Polygon. This establishes an independent time anchor outside the primary settlement chain.

END-TO-END

Canonical Workflow

$100M tokenized treasury asset transfer between two institutional counterparties under the CLARITY Act. Aligned with Seong-Ho's IdentHash escrow flow.

1

Asset Creation `LAYER 1 + 2` `H33-Upstream` `H33-74`

Issuer creates tokenized treasury on Polygon. IdentHash committed with issuer's `secId` and `PII_commitment`. H33-Upstream binds provenance: CUSIP, maturity, coupon rate, regulatory classification, issuer identity.

- H33-74 attestation #1: creation event (domain `0x50`)
- Bitcoin mainnet anchor: 32 bytes via `OP_RETURN`

2

Ownership Establishment `LAYER 1`

Institutional buyer identified via IdentHash. Buyer's `secId` bound to ownership record. Merkle leaf: `H(identHash || timestamp || block_number)`. No PII on-chain — only `PII_commitment` inside `S_bundle`.

- IdentHash function: `createOwnership`

3

Escrow Creation `LAYER 1 + 2` `H33-74`

`createEscrow` called with both parties' IdentHashes. Token locked, stablecoin locked (atomic DvP preparation).

- H33-74 attestation #2: escrow creation
- IdentHash function: `createEscrow`

4

Governance Approval

`LAYER 3` `H33-Q-Sign` `H33-74`

Compliance committee evaluates transfer. H33-Q-Sign: 3-of-5 multi-party governance proof. `governanceProofHash` committed via `confirmByManager`. `GovernanceProofAttached` event emitted.

- H33-74 attestation #3: governance approval
- IdentHash function: `confirmByManager`

5

AI-Assisted Compliance

`LAYER 4` `Agent-Zero` `H33-74` `Cachee`

H33-Agent-Zero evaluates on encrypted data: counterparty accreditation, sanctions screening, jurisdiction restrictions, transfer limit compliance. Decision in 4.1ms on encrypted data.

- H33-74 attestation #4: compliance decision
- Attestation cached in Cachee with hash chain link

6

Settlement

`LAYER 1 + 2` `H33-74`

`settleEscrow` executes atomic DvP. Token to receiver wallet, stablecoin to sender wallet. Full revert if either transfer fails. New Merkle root computed with updated ownership.

- H33-74 attestation #5: settlement event (`h33Anchor` on-chain)
- `OwnershipTransferred` + `EscrowSettled` events emitted
- IdentHash function: `settleEscrow`

7

Audit Replay

`LAYER 4` `Cachee`

Regulator queries attestation chain (#1 through #5). For each: verifies H33-74 substrate, checks domain, validates three PQ signatures. Reconstructs the full

provenance chain: who created, who owned, who approved, what AI decided, when transfer settled.

- Merkle proof confirms ownership at each timestamp
- Q-Sign proof confirms governance was valid at each timestamp
- Agent-Zero attestation confirms compliance evaluated without plaintext exposure
- Auditor never accesses: investor PII, financials, sanctions data, jurisdiction mappings
- **Compliance proven, not trusted**

8

Future Verification

LAYER 5 3-Family PQ

10 years later: ECDSA potentially weakened. `secId` persists — ownership identity unchanged. H33-74 attestations remain verifiable (three PQ families). Merkle proofs remain verifiable (hash-function agnostic). Q-Sign governance proofs remain verifiable. The entire ownership history is independently reconstructable using only on-chain data + open-source verification tools.

STATUS

Deliverables

FROM H33 (LAYERS 2–5)

- [x] Layer 2 section draft (attestation + provenance)
- [x] Layer 3 section draft (Q-Sign governance)
- [x] Layer 4 section draft (Agent-Zero + Cachee replay)
- [x] Layer 5 section draft (PQ survivability)

- [x] Full canonical workflow (8-step, aligned with IdentHash)
- [x] Attestation lifecycle diagrams
- [x] Q-Sign four-state machine diagram
- [x] Agent-Zero decision flow diagram
- [x] Regulator replay workflow diagram
- [x] H33-74 substrate layout visual

FROM SEONG-HO LEE (LAYER 1)

- [x] Layer 1 draft v4 (legal ownership foundation)
- [x] IdentHash architecture
- [x] Smart contract interfaces
- [x] CLARITY Act alignment
- [x] Merkle ownership continuity
- [x] Recovery architecture
- [x] H33 integration points defined
- [x] Layer 1 content fully integrated into joint paper

JOINT

- [x] Abstract (rewritten to reflect full 5-layer scope)
- [x] Section ordering and paper flow
- [ ] Peer review target list
- [ ] Venue selection (journal / conference / institutional whitepaper)

THE SHIFT

What Changes

TRADITIONAL SYSTEM		PROVABLE CONTINUITY
Logs expire	→	Replay survives
AI model changes	→	Decision preserved
Vendor dependency	→	Independent verification
Screenshots disputed	→	Cryptographic reconstruction
Reconstruction: weeks	→	Verification: minutes
Ownership: key-dependent	→	Ownership: identity-permanent

The industry has spent a decade building systems that record what happened.

This paper describes infrastructure that proves it.

Ownership continuity. Governance continuity. AI decision continuity. Cryptographic survivability. One incident. Five layers. Complete evidentiary reconstruction — independent of time, infrastructure, vendors, and cryptographic eras.

Provable continuity is not a feature. It is the minimum requirement for institutional-grade digital infrastructure.

SEONG-HO LEE

Founder & CEO, PayKing Corporation / KoreaGemLab

Layer 1 — Legal Ownership Foundation (Ownership Substrate)

Patent: KR 10-2026-0068484

paykingos.com · seongho@paykingos.com

ERIC BEANS

CEO, H33.ai, Inc.

Layers 2–5 — Attestation, Governance, AI, PQ Survivability

7 Patents Pending · 300+ Claims

eb@h33.ai · h33.ai
