SOVEREIGN PROOF

A Cryptographically Signed Biography for Physical Hardware

Tamper-evident chain of custody for the AI and critical-infrastructure supply chain

Technical White Paper · Version 1.3 · October 2026
Michael Bardwell-Scott · Sovereign Proof, Inc. · sovereignproof.io

ABSTRACT

High-value computing hardware crosses continents, carriers, integrators, and maintenance loops before and after it is trusted with critical workloads — yet its provenance is documented in artifacts that cannot be verified: vendor certificates, packing slips, and asset-database rows that nothing cross-checks and anything can silently edit. This paper presents the Sovereign Sentinel, a working system that gives every serialized asset a cryptographically signed, hash-chained custody biography; scores each asset's trustworthiness with deterministic, reproducible rules; and anchors the fleet's integrity to the Bitcoin blockchain through a single 32-byte commitment — while, in operation, all custody evidence remains inside the operator's facility. We describe the architecture, the trust-scoring model, the threat model — including the input layer, which we treat as the primary attack surface — and the system's honest limitations; we map the evidence it produces to the documentation standards emerging under DFARS 252.246-7007/-7008 and Section 5949 of the FY2023 NDAA; and Appendix A specifies the integration that binds custody records to manufacturer-rooted silicon identity.

1 · The problem: provenance is paper 2
2 · Design principles 3
3 · The biography model 3
4 · The custody ledger 4
5 · Silicon binding 5
6 · Public anchoring without a token 5
7 · The deterministic trust engine 6
8 · The interrogator 7
9 · Ingestion and the Baseline Risk Audit 7
10 · Threat model and honest limitations 8
11 · Compliance mapping 10
12 · Alternatives considered 10
13 · Conclusion 10
Appendix A · Hardware attestation integration 12

1 · The problem: provenance is paper

A modern accelerator is a seven-figure asset whose life before the rack is almost entirely undocumented in any verifiable sense. It is born in a fab, packaged, tested, shipped through freight forwarders and customs brokers, staged in warehouses, integrated by a systems vendor, shipped again, received, and installed. At each boundary it is accompanied by paper: a certificate of conformance, a packing manifest, a test report. Those documents die at the receiving dock. They are filed, not verified; none of them is cryptographically bound to the physical object; and the asset-management rows they populate can be edited by anyone with database access, silently, forever after.

Three structural gaps follow:

- **The custody dark zone.** Between the manufacturer's records and the operator's first scan there is typically no custody evidence at all — months in which a part can be swapped, re-marked, or interdicted with nothing in the record to show for it.
- **The maintenance loop.** Replacement parts enter operating facilities through spares pools and RMA channels with no provenance check at the moment of installation — the most common counterfeit entry point in fielded systems.
- **The facility blind spot.** Silicon attestation — via a Trusted Platform Module (TPM), DICE-derived device identity, or SPDM device measurement, the established device-attestation standards — protects compute devices that carry attestable silicon. The power and cooling chain that keeps those devices alive — switchgear, UPS modules and battery strings, generators, PDUs, chillers — generally carries none. In a representative synthetic data-center digital twin we analyzed (21,474 serialized components; anomalies deliberately seeded as a controlled detection test), every IT asset class carried trusted-platform silicon and the entire facility plant carried none. For the facility side of a data center, physical custody evidence is the only trust layer available.

The same controlled test illustrates what unverified records look like when something finally cross-checks them: 7,270 of 21,474 components (34%) carried a manufacture date *after* their recorded installation; 126 lot codes were claimed by more than one manufacturer; and the median gap between factory and rack with zero intervening custody events was 209 days. All counts are deterministic: three independent recomputations produced identical figures, every seeded defect in these classes was surfaced, and components carrying no seeded defect produced zero findings in them — the engine's false-positive guards are part of its test suite. The point of the exercise is not that any particular operator's data looks like this — the anomalies were seeded — but that every one of these defect classes is mechanically detectable from fields that standard asset databases already hold, and that in conventional systems nothing checks them.

Meanwhile the documentation standard is acquiring dates. DFARS 252.246-7007/-7008 — counterfeit-part detection, avoidance, and reporting — is in force today for the defense supply chain, and its evidence expectations flow down contractually. Section 5949 of the FY2023 NDAA restricts covered Chinese-fab semiconductors in federal procurement effective December 23, 2027. The proposed FAR implementation (February 2026) sets the operative standard: certification after a **"reasonable inquiry."** None of this mandates any particular technology, and this paper claims no mandate — but an inquiry that cannot be

proven is an inquiry that did not happen, as far as an auditor is concerned. The question the supply chain will face is not *"are you compliant?"* but *"show me the inquiry."*

2 · Design principles

1. **Determinism over judgment.** Trust must be computed by rules that are published, reproducible, and arguable — never by a model's opinion. In the Sentinel, a language model narrates evidence; it never computes or adjusts a score.
2. **Evidence stays home.** Custody data is security-sensitive: it describes what is installed where, and who touched it. In operation, the only data the Sentinel transmits off-site is a single 32-byte commitment to the fleet's integrity. There is no hosted registry, no telemetry, and no cloud dependency; the interrogation model runs locally.
3. **Tamper-evidence over tamper-prevention.** No record system can prevent a privileged insider from editing a database. A signed hash chain makes the edit *provable* — and an external anchor makes wholesale substitution provable too.
4. **Fail closed.** A key secret that cannot decrypt the existing keystore refuses to mint keys rather than silently re-keying; a missing hardware attestation library is an error, never a silent downgrade to software; a watchdog without its secret refuses to arm. Security backends do not degrade quietly.
5. **Asset-agnostic.** A diesel generator, a battery string, or a pallet of DIMMs gets the same signed biography as a GPU server. Nothing in the data model assumes the asset can compute.
6. **No token.** Public verifiability is achieved by timestamping a hash into Bitcoin via OpenTimestamps. There is no cryptocurrency, no gas, no chain dependency in the product, and nothing to buy to verify.

3 · The biography model

The Sentinel models an asset's life as a graph: a **Hardware** node (uid, serial, model, genesis date) experiences an ordered sequence of **Events**, each of which may occur at a **Location** (with a declared security level, 1–5) and be performed by an **Actor** (with a declared clearance level, 1–5). Five event types cover the custody lifecycle: `BIRTH` (manufacture/genesis), `SCAN` (a verification touchpoint — receiving, install, inspection), `TRANSPORT`, `MAINTENANCE`, and `TAMPER` (a recorded physical-integrity incident). Events carry structured metadata, including the identity fields the trust engine reasons over: lot code, date code, wafer lot, fabrication site, manufacturer, part number, and parent-assembly references.

The store is a swappable repository (an embedded database by default; a graph database in larger deployments) behind a single interface, and every piece of trust logic consumes the same ordered event history regardless of backend.

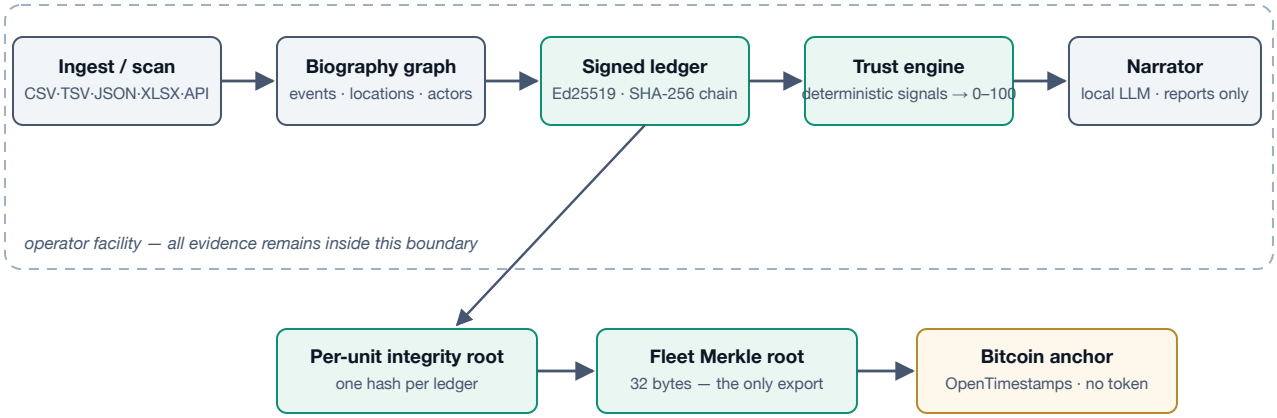


Figure 1 — The evidence pipeline. In operation, custody data never leaves the facility; only a 32-byte fleet commitment is published, and any party can later verify a unit's inclusion against it.

4 · The custody ledger

Every unit's event history is compiled into a per-asset ledger: one record per event, each carrying the record index, the previous record's hash, a SHA-256 hash over a canonical JSON commitment of the record's content, the signer's identity, and an Ed25519 signature over the committed hash. The first record chains from a fixed genesis value. Events performed by a known actor are signed with that actor's key; events with no human actor are recorded under a distinguished system identity that is hash-chained but signature-exempt by design, carrying a null signature that is visible as such in every verification output. System-recorded events still enter only through the authenticated write path and are committed by the public anchor like every other record — the exemption is in signature verification, never in write access. Actor identities carry declared clearance levels and are policy-bearing, and there are no silent writes: an event attested by a named actor is signed under that identity, an event signed by a cryptographically valid but unauthorized identity is itself a scored finding, and an event submitted with no actor at all is recorded under the system identity and surfaced by the engine as unattributed — its authorship rests on write-path access, which is why the reference deployment binds sessions to individuals (§10). In the shipped system, events are signed on receipt under the submitting actor's registered identity — over a write path the production baseline requires to be API-key authenticated — with keys held in the deployment keystore or HSM. Per-individual, physical-presence authentication is bound at the session layer in the reference deployment of §10, and signing at the scanning device itself — removing the server as the single signing point — is a hardening step of that same profile.

Tamper semantics. Verification recomputes every link and every signature against the stored chain. Mutating a stored record breaks that record's integrity check while the rest of the chain still verifies — a single edit is localized and provable. What remains attributable is the originally signed content; the altered content is precisely what no one ever signed. Detection then escalates on two independent axes: the trust engine re-scores the unit with a fatal `LEDGER_TAMPERED` finding (score floors to zero), and the unit's recomputed integrity root changes, which changes the fleet root — so the public anchor immediately reports **DIVERGED** against the last committed state. An attacker who can edit the database thus faces a

choice: leave the broken record (provable locally) or rebuild the whole chain (provable globally against the external anchor).

Key custody. Signing keys are generated randomly, never derived. With the keystore secret configured — required by the production hardening baseline — keys are encrypted at rest with AES-256-GCM, and a secret that cannot decrypt the existing store fails closed: the system refuses to mint keys over a store it cannot read. For production deployments the same interface is backed by a PKCS#11 HSM or token, with keys generated inside the device and marked non-extractable: private keys never enter process memory.

Write integrity. Every read-modify-write on a unit's ledger or on the anchor chain is serialized under a per-unit lock, and slow network operations are kept outside the critical section. The deployment model is deliberately one verification instance per site — matching, incidentally, the data-sovereignty posture most operators demand anyway.

5 · Silicon binding

Where the asset can hold a key, the record is bound to the silicon itself. Each unit receives an endorsement keypair at genesis, with the public key recorded in the BIRTH event's metadata — inside the signed chain. A verification event carries a per-scan challenge nonce and the device's quote over it — today those quotes enter the system as recorded custody evidence through ingestion (the product does not yet probe devices itself; Appendix A.1 states the boundary exactly) — and the engine verifies every recorded quote against the key the device was born with. Scanner-enforced nonce freshness — guaranteeing a quote was produced at the moment of challenge — lands with the hardware attestation integration. A quote signed by the wrong key is treated by the engine as a fatal `SILICON_MISMATCH`: the record no longer describes the device in hand.

The attestation layer is pluggable: a software implementation (the default, used in demonstrations and data-only audits), and hardware backends targeting TPM 2.0 quotes and SPDM device attestation. Hardware backends fail closed — if the library or device is absent, attestation errors rather than silently downgrading. Production integration of the hardware backends against vendor TPM/DICE/SPDM ecosystems is active roadmap, not shipped capability; the system consumes existing manufacturer attestation ecosystems rather than requiring foundry participation. Appendix A specifies that integration path — the manufacturer-rooted evidence that exists on commodity servers today, exactly what the shipped product does and does not consume, and the enrollment design that anchors custody chains to vendor roots where silicon permits. For the large class of facility assets with no attestable silicon at all, the custody chain itself is the identity layer — which is precisely why the architecture does not assume a TPM.

6 · Public anchoring without a token

Local cryptography proves a record was not *edited*; it cannot by itself prove the whole database was not *replaced*. For that the Sentinel publishes a commitment: each unit's ledger reduces to an integrity root, the fleet's roots reduce to a single 32-byte Merkle root, and that root is timestamped into the Bitcoin blockchain via OpenTimestamps. The anchor history itself is an append-only, hash-chained record.

Three properties matter. First, **privacy**: the published value is a hash commitment — no serials, no topology, no counts, nothing recoverable from it. Second, **independence**: OpenTimestamps proofs verify against public calendar servers and the Bitcoin chain with open-source tooling; no cooperation from Sovereign Proof is required to check one. Third, **selective disclosure**: the system produces per-unit Merkle inclusion proofs, so an operator can demonstrate to an auditor that one specific unit's intact history is covered by a public commitment without revealing anything about the rest of the fleet.

Air-gapped operation is a first-class state: a disconnected site stages its roots, and any connected machine can later submit them for timestamping. Anchor states progress `staged → pending → confirmed` (Bitcoin block height recorded), and the console surfaces one of two verdicts at all times: **ANCHORED** (current fleet state matches the latest public commitment) or **DIVERGED** (state has changed since — including by tampering).

7 · The deterministic trust engine

Trust is a number computed by rules, not a judgment. Every registered signal examines the unit's event history and emits findings; each finding carries a stable code, a severity, a point deduction, and an evidence message naming the specific records involved. A unit starts at 100; deductions subtract; any **fatal** finding floors the score to zero regardless of arithmetic. Thresholds: 100 **VERIFIED**, ≥50 **DEGRADED** (investigate), below 50 **CRITICAL** (quarantine). VERIFIED is deliberately strict — it means *no adverse finding at all*; any deduction demotes to DEGRADED, because a trust score that rounds away open questions is not a trust score. Conversely, structurally absent evidence is never penalized: an asset with no attestable silicon enrolled receives an informational custody-only-identity note at zero deduction, while an asset enrolled for attestation but never challenged carries a warning. Scoring the same history always produces the same score — a property that matters when the number appears in an audit trail.

Signal	Finding codes	Severity	What it catches
`Genesis`	NO_GENESIS · UNTRUSTED_GENESIS	Fatal	No birth record, or birth at a facility below the trusted-foundry threshold — provenance cannot be established.
`Chronology`	CHRONOLOGY_INVERSION	Fatal	A genesis record dated after later lifecycle events — the history is self-contradictory and cannot describe the physical part.
`Date-code consistency`	DATE_CODE_MISMATCH · MALFORMED_DATE_CODE	Critical / warning	A physically marked date code decoding to a fab week after the unit's first field event (re-marking fingerprint), or a code that cannot exist. Decode convention is pinned and published so counts reproduce exactly.
`Lot identity`	LOT_CONFLICT	Warning	The same lot code claimed under different manufacturers across the fleet — one batch identity cannot have two owners.

`Custody continuity`	TRANSIT_GAP	Warning	An unaccounted-for window between consecutive events exceeding the transit budget.
`Location security`	LOW_SECURITY_LOCATION	Critical	Custody passing through an uncontrolled handling environment.
`Route integrity`	UNSCHEDULED_STOP	Critical	A transport leg flagged as an unscheduled or unauthorized diversion.
`Tamper seal`	UNRESOLVED_TAMPER	Fatal	A recorded tamper event with no subsequent clearance-5 maintenance resolution.
`Ledger attestation`	LEDGER_TAMPERED · UNAUTHORIZED_SIGNER · NO_LEDGER	Fatal / critical / warning	A broken hash link or signature (the record was altered); a cryptographically valid signature from an identity not authorized to attest custody; or no signed chain at all.
`Device attestation`	SILICON_MISMATCH · UNATTESTED_SILICON · CUSTODY_ONLY_IDENTITY	Fatal / warning / info	A quote verifying against the wrong endorsement key (not the same device); an enrolled device never challenged; or — informational, zero deduction — no attestable silicon enrolled at all.

Two engineering disciplines keep the engine honest. **False-positive guards:** every signal ships with tests proving it fires on the defect and stays silent on clean histories and on absent data — a signal that cannot distinguish "no data" from "bad data" is a defect. **Calibration:** deductions are sized so one moderate issue lands a unit in DEGRADED and quarantine requires either accumulation or a genuinely fatal fact; fatality is reserved for findings that mean the record cannot describe the part, not for suspicion.

8 · The interrogator

Every verdict can be questioned in plain English — "where were you during the gap?", "who signed for you in transit?" — and the unit answers in the first person from its own evidence. The implementation discipline matters more than the feature: the language model runs locally (no cloud round-trip), at temperature zero, and is handed the computed verdict and findings as context it may only explain. The verdict headline is enforced server-side — the model cannot soften a quarantine regardless of how a question is phrased. Untrusted content (imported metadata, notes fields, operator questions) is fenced and bounded before it reaches any prompt, treating prompt injection via custody data as a first-class attack surface. When no local model is available, a deterministic templated narrator answers from the same findings — the evidence pipeline never depends on the model being present.

9 · Ingestion and the Baseline Risk Audit

The system meets operators where their data already lives. Custody histories import from the formats asset systems actually export — CSV (comma, tab, or semicolon delimited, auto-detected), JSON row arrays, and Excel workbooks (every sheet carrying the required columns is ingested) — with file-type detection by content rather than extension, hostile inputs bounded and rejected cleanly, and per-row

errors reported without aborting the good rows. Identity fields (lot, date code, wafer lot, fab site, manufacturer, part number) travel as structured metadata into the signed chain, where the provenance signals reason over them. A live ingestion API accepts signed custody events from scanners and operational systems as they happen.

The commercial entry point is a fixed-scope diagnostic — the **Baseline Risk Audit**: a read-only export of one facility's asset records (pseudonymized serials acceptable) is run through the full engine, producing a findings report and a per-asset trust register. The deliverable maps each finding class to the specific compliance element it bears on, and carries — prominently and by design — the disclaimer that it is a technical diagnostic, not a legal opinion or certification, and that absence of a finding is not a representation that no defect exists. The report generator is built into the product: the operator's own console produces the compliance evidence report on demand, from the current state of the signed fleet — full verification and scoring of the demonstration fleet completes in well under a second on commodity hardware, and fleet-scale ingestion is bounded by configurable caps and chunked imports.

10 · Threat model and honest limitations

A trust product earns credibility by stating what it does not do. The Sentinel's boundaries:

- **Tamper-evidence, not tamper-prevention.** The system cannot stop a privileged actor from editing a database. It makes the edit provable (broken record integrity, fatal re-score) and wholesale substitution provable (anchor divergence). Detection of a rolled-back ledger relies on the external anchor and covers events committed before the most recent anchor — anchoring cadence is therefore a security parameter, not a cosmetic one.
- **Records versus reality.** A signed chain proves custody *records* are intact and attributable. The correspondence between records and physical objects is established by the custody process itself — scans at handoff, attestation where silicon permits — and strengthens as instrumentation of those touchpoints deepens. A perfect forgery entered at the point of origin is a lie signed honestly; what the system guarantees is that the lie is permanent, attributable to its signer, and exposed the moment any cross-checkable fact (a date code, a lot identity, a chronology, a silicon challenge) contradicts it.
- **Attestation coverage is heterogeneous.** Software attestation is the shipped default; TPM 2.0 and SPDM hardware backends exist behind a fail-closed interface but their production integration is roadmap. Facility assets without attestable silicon rely on the custody layer alone — by design, since no alternative exists for them.
- **Scale posture.** The deployment unit is one instance per site, with write serialization in-process. Multi-site estates run parallel instances; cross-site aggregation is roadmap. This matches the air-gap posture but is a stated architectural boundary, not an accident.
- **Assurance status (input layer).** See the input-layer model below — the operational threat surface is treated there as a first-class subject, not a bullet.
- **Assurance status.** The software is working and heavily tested — 200+ automated tests covering the cryptographic layer, signal engine, ingestion, and APIs, plus published conformance vectors that let third parties verify the custody-ledger format independently (anchor-side vectors are planned alongside the hardware integration) — and it has undergone internal security review with all findings

remediated. It has not yet completed an independent third-party security audit; one is planned following hardware integration, ahead of production deployments. Materials from Sovereign Proof state this plainly, and claims of "audited" should be read accordingly.

The input layer, treated as the primary attack surface

The cheapest realistic attack on this system involves no cryptanalysis. Nobody breaks AES-256-GCM or unspools a hash chain when the alternative is a compromised scanner at the receiving dock, a login shared across a shift, or an operator paid to scan a counterfeit in as genuine. We state the consequence as bluntly as an auditor would: **if the input layer lies, the chain hardens the lie.** This is the governing physical threat to any custody system, ours included, and the design treats it on three levels.

What the architecture enforces today. A forged human attestation must be signed under a specific named credential and is permanently attributable to it; an event submitted with no actor at all is not silent either — it is recorded under the system identity and surfaced as an unattributed-event finding, so authorship questions are visible rather than buried. Authorization is scored, not assumed: a valid signature from an identity not authorized to attest custody is itself a finding. And the record-consistency signals of §7 are an input-layer defense in their own right: a fabricated event must agree with the unit's chronology, its lot identity across the whole fleet, its date codes, its custody budgets, and every record entered before and after it — at entry and at every future re-score. A lie does not have to be caught at the dock to be caught; it has to stay consistent with everything ever recorded, forever, while the public anchor bounds any attempt to quietly rewrite the past. Credential compromise is handled as a policy event, not a schema event: trust policy lives in the signals and is re-evaluated over full history at every scan, so deregistering or demoting a compromised identity retroactively converts every event it ever signed into scored findings — a compromise window becomes a visible, bounded set of flags rather than a permanent blind spot. Lying to this system is possible; lying cheaply and permanently is not.

The reference deployment profile. The product does not dictate a facility's device fleet or identity stack — an evidence system that requires re-provisioning a warehouse before it runs will simply not be deployed. The reference profile the pilot ships with is deliberately ops-tolerable, on the principle that security which slows the floor gets bypassed: single-purpose scanning devices locked to the Sentinel client (MDM/kiosk-class management), per-individual session authentication bound to physical presence (badge-tap-class, with an operator-tuned idle timeout — no shared logins, and no added steps per scan), scanners enrolled as named actors in the custody graph, and event signing on receipt under the authenticated identity with keys in the deployment keystore or HSM. Device-resident signing — the step that removes the server as a single signing point — lands with the hardware attestation integration (Appendix A).

The boundary, stated without abdication. The Sentinel binds the custody reality the operator's process produces; it cannot make a dishonest process honest. What it changes is the economics and the accountability: every assertion has a named, signed author; every contradiction within the engine's published signal classes is mechanically detectable for the lifetime of the record — and the signal set extends without touching stored history; and the operator's own auditors can see precisely which identities attested what, when. An operator whose input layer is weak does not get a false clean bill — they

get a permanent, attributable record of exactly what their process claimed, and a trust engine built to notice when those claims disagree with the physics of their fleet.

11 · Compliance mapping

The Sentinel is an evidence substrate, not a compliance opinion. Its output is mapped to what each finding class *bears on*:

Evidence the system produces	Bears on
Signed, tamper-evident custody chain from genesis to present; per-asset trust register	DFARS 252.246-7007 supplier traceability and documentation review; the "reasonable inquiry" standard of the proposed FAR rule implementing NDAA §5949
Origin and fab-site attribution bound into the signed record; genesis trust policy	§5949 covered-source inquiry; counterfeit-avoidance traceability to original manufacturer
Record-consistency findings (chronology inversions, date-code contradictions, lot-identity conflicts)	Counterfeit-detection criteria of SAE AS5553-class screening programs adopted under DFARS -7007
Recorded tamper events and their authorized resolution status	DFARS 252.246-7008 reporting obligations for suspect items
Publicly anchored fleet commitment with per-unit inclusion proofs	Evidence-integrity: demonstrating to any auditor, without privileged access, that records predate the inquiry and have not been rewritten since

12 · Alternatives considered

Token-based provenance chains. Writing custody data to a public blockchain (typically with a protocol token) puts sensitive operational data — or at minimum its traffic patterns — outside the facility, introduces a currency dependency enterprises and defense programs will not accept, and adds nothing evidentiary beyond what a 32-byte timestamped commitment already provides. The Sentinel keeps evidence local and publishes only the commitment; public verifiability without a token is the entire point of using OpenTimestamps.

Silicon attestation alone. TPM/DICE/SPDM attestation answers "is this the same device?" for devices that can hold a key. It does not cover the facility plant, passive components, or the custody question — where the device was, who handled it, and whether its paper trail is self-consistent. The Sentinel consumes silicon attestation where it exists and roots the rest of the record below it.

Vendor certificates and audits of record. Per-vendor birth certificates are unverifiable paper that dies at receiving and crosses no vendor boundary. Periodic manual audits sample records at a point in time. A signed chain is continuous, cross-vendor by construction, and verifiable record-by-record years later.

13 · Conclusion

The supply chain for critical hardware has an evidence problem before it has a security problem: nothing that accompanies a physical asset today can prove anything, to anyone, later. The Sovereign Sentinel's contribution is to make custody evidence *cryptographic without becoming exotic* — standard signatures and hash chains over the operator's own records, deterministic rules anyone can re-run, silicon binding where physics permits, and a public commitment that costs 32 bytes of disclosure. The result is a biography that hardware can be asked to defend: provable where it has been, attributable in every assertion, and loud precisely when someone tries to rewrite it.

A live demonstration is public at **demo.sovereignproof.io** — including the tamper exercise: alter a record and watch the chain, the score, and the public anchor catch it.

Appendix A · Hardware attestation integration

Added in v1.1. This appendix answers a question we are asked in technical diligence: what can a scan obtain from a commodity Dell, HPE, or NVIDIA-equipped server without prior Sovereign Proof enrollment, and which of those attributes trace cryptographically to a manufacturer-issued root of trust? We verified every vendor-ecosystem claim below against vendor primary documentation in October 2026.

A.1 · What the shipped product does today — stated exactly

A Sentinel scan today is a records operation: it re-verifies and re-scores the signed custody history already in the operator's store. It does not probe the server. Scanning a never-enrolled serial fails with "Unknown serial number" (HTTP 404) — nothing is obtained. Every cryptographic property of the custody chain itself (signed events, the hash-chained ledger, the fleet Merkle root, the public anchor) proves **record integrity and attributability since enrollment** and chains to keys held in the deployment's own keystore. One manufacturer-rooted channel is now consumed by a working prototype of this appendix's phase 1: TPM EK certificate chains are validated against pinned vendor roots at enrollment, stored as evidence, fingerprint-committed into the signed ledger, and re-validated from the record at every scan. The TPM 2.0 and SPDM attestation backends exist in the codebase as deliberately fail-closed integration seams — they refuse to run rather than simulate — and the product ships no Redfish, IPMI, SMBIOS, or NVML collection code today. The remainder of this appendix describes the evidence that exists on the hardware side and the integration that consumes it. Status per phase: **EK certificate-chain enrollment is a working prototype** (validation, pinned roots, ledger-committed evidence, per-scan re-validation — revocation checking and TPM credential activation, which proves key *possession* rather than certificate existence, are its named next steps); everything else remains design, and we flag the distinction wherever it matters.

A.2 · Manufacturer-rooted evidence on commodity servers, as of October 2026

Table A-1 — Evidence present on the hardware side as of October 2026, independent of Sovereign Proof. **Of these channels, the TPM EK certificate path is consumed by a working prototype (A.1); the rest are not yet consumed** — A.3 specifies the integration designed to consume them.

Evidence	Read via	Chains to	Presence on a stock machine
Service tags, serials, UUIDs, FRU inventory	Redfish / IPMI / SMBIOS	**Nothing** — firmware-reported strings, spoofable by anything that controls firmware or the management plane	Universal
GPU UUID, board serial, PCIe IDs, VBIOS version	NVML / nvidia-smi	**Nothing**	Universal
TPM 2.0 Endorsement Key certificate	TCG-specified TPM NV indices (OS-level access); some firmware TPMs	**TPM silicon vendor's CA** (Infineon, Nuvoton, STMicroelectronics, …)	Near-universal on current fleets: embedded TPM 2.0 is standard on ProLiant Gen11; on PowerEdge

	require retrieval from the TPM vendor's service		the TPM 2.0 module is a factory option fitted to virtually all non-restricted-market configurations
BMC device identity (IEEE 802.1AR IDevID)	iDRAC9 "Default IDevID"; iLO factory IDevID	**Dell / HPE manufacturing CA**	Dell: stock — burned at the factory on every iDRAC9 (using it for 802.1X is license-gated). HPE: factory-provisioned only when the IDevID SKU is ordered — an order-time option like the Platform Certificate
TCG Platform Certificate (binds platform model/serial to the TPM EK)	TPM NV / vendor tooling (HPE PCVT and equivalents)	**Dell / HPE factory CA**	**Order-time option only** — Dell via the SCV SKU, HPE as a factory option; not common in deployed fleets
Dell Secured Component Verification: factory-signed component inventory (validation binds the inventory to the TPM EK certificate serial — SCV is Dell's platform-to-EK binding)	iDRAC + SCV tooling (licensed)	**Dell SCV Root CA** (generation-scoped roots: a common 15G/16G root, a separate 17G root)	Paid SKU + license, iDRAC9-era platforms onward
SPDM component attestation (NICs, storage controllers, add-in cards)	iDRAC9 7.x (license-gated); iLO6 Gen11 "Global Component Integrity"; Redfish ComponentIntegrity (DMTF DSP0268 data model; SPDM per DSP0274)	**Per-component vendor DeviceID chains**	Shipping on current-generation platforms (iDRAC9 7.x; iLO6 on Gen11) but gated per component and per license — not yet a fleet-wide assumption
NVIDIA GPU device identity + signed firmware measurements	SPDM; NVML attestation-report calls	**NVIDIA Device Identity CA** — per-device ECC P-384 key fused at manufacture; measurements validated against NVIDIA-signed reference values (RIMs)	**Hopper (H100/H200) and Blackwell only**; report collection requires a CC-capable SKU with Confidential Computing mode enabled before GPU boot. **A100, V100, and pre-Blackwell RTX-class parts expose no externally verifiable manufacturer identity** — only unauthenticated NVML strings (the CEC-class ERoT on DGX/HGX A100 boards enforces firmware authenticity internally but,

			lacking SPDM, emits nothing a third party can verify)
ConnectX-7 NICs / BlueField-3 DPUs	SPDM certificate slots	**NVIDIA-provisioned DICE certificate chains** (production-provisioned, write-protected)	Shipped on those parts
HGX/DGX baseboard ERoTs	Out-of-band via platform BMC (Redfish/SPDM)	Component chains surfaced by the BMC	Platform-dependent

Three boundaries matter. First, **an EK certificate proves "a genuine TPM from a stated vendor" — it does not bind to the server's model or serial**; that binding is precisely what Platform Certificates and Dell SCV add, and both are opt-in purchases rather than fleet norms. Second, **HPE's iLO silicon root of trust is internal enforcement** — it verifies firmware before boot but emits no externally verifiable evidence, so we never cite it as evidence a third party can check. Third, NVIDIA's hosted attestation services (NRAS, RIM service, OCSP) **authenticate with NGC API keys since August 2025** (unauthenticated requests are heavily rate-limited rather than refused); verification against NVIDIA's published PKI and signed reference values can be performed locally, which is the path an air-gapped deployment uses anyway.

A.3 · The integration design: enrollment anchored to manufacturer roots

The integration's principle is that **enrollment itself consumes the manufacturer root**, replacing the deployment-minted key as the identity anchor wherever silicon permits. At first scan of a TPM-bearing server: read the EK certificate; validate its chain to the pinned TPM-vendor root (with revocation checking); bind an Attestation Key to that EK by TPM credential activation (TPM2_MakeCredential / TPM2_ActivateCredential — the TPM can recover the challenge secret only if it holds that EK's private key *and* the attestation key named in the credential is resident on the same TPM); and record the **full certificate chain and the activation evidence inside the signed genesis record**. That last step carries the burden of proof: the ledger stores the vendor's certificates, not our conclusion about them, so an auditor re-validates the chain independently of Sovereign Proof, years later, from the record alone. The NVIDIA path is parallel: SPDM certificate retrieval and a signed measurement report at enrollment, chain-validated to NVIDIA's published root, measurements compared against NVIDIA-signed reference values, revocation checked at validation time — all of it recorded, with timestamps, in the chain. Where a Platform Certificate or SCV inventory exists, it is validated and recorded, upgrading the binding from "genuine TPM" to "genuine TPM in this specific platform"; where it does not, the record says so, and the custody process carries the platform binding. Identity tiers are explicit in the record — manufacturer-rooted, deployment-keyed, or custody-only — and scoring policy can distinguish them.

Scoring semantics are unchanged by the integration: a quote that fails against the recorded identity remains fatal (`SILICON_MISMATCH`); absence of attestable silicon remains an informational note, never a deduction; and hardware backends keep the fail-closed contract — a missing library or device is an error, not a silent downgrade. Per-scan nonces are recorded today; scanner-enforced nonce freshness ships with this integration, as stated in §5.

A.4 · Attack surface and honest boundaries

- **Identity is not custody.** A manufacturer root proves what the device is and what firmware it runs — never where it has been. A stolen genuine H100 attests as a genuine H100. The custody chain remains the layer that catches diversion, substitution windows, and paperwork contradictions; the two layers compose and neither substitutes for the other.
- **Coverage is bounded, and the paper's own dataset shows the bound.** Manufacturer-rooted identity today reaches TPM-bearing servers and Hopper-onward GPUs. It does not reach A100-class accelerators, the facility power chain, spares, or passives. In the reference digital twin of §1, every IT asset class carried attestable silicon and the entire power and cooling plant — roughly half the equipment inventory — carried none. An architecture that depended on vendor roots would address a fraction of a facility's serialized assets.
- **Measurements need references.** A firmware measurement is only verification when compared against a signed reference value. NVIDIA publishes signed RIMs; server-side reference values vary by vendor and generation. Where no reference exists, the record stores the measured value and flags the absence — measurement without a reference is inventory, not verification, and the record never conflates them.
- **Chain validation is policy-bearing.** Vendor root stores are pinned and versioned in the deployment (an air-gapped site cannot fetch roots ad hoc), revocation artifacts are cached with their retrieval times recorded, and a validation performed against stale revocation data is recorded as such rather than presented as current.
- **Privacy.** EK certificates are privacy-sensitive platform identifiers. They are recorded inside the operator's own deployment and never leave it — consistent with the architecture's standing boundary that only the 32-byte fleet commitment is ever published.
- **Status.** Phase 1 of A.3 — EK chain validation against pinned vendor roots, evidence stored and fingerprint-committed into the signed ledger, re-validated at every scan, with destroyed or swapped evidence scored fatally — is a working prototype in the codebase (enrollment endpoint, validation module, scoring signal, and a software-TPM demo kit). Revocation checking and credential activation are its named next steps. The remainder of A.3–A.4 is design, specified against the same fail-closed seams. Device attestation beyond the EK path remains deployment-keyed.

A.5 · Phasing

Integration lands in the order the evidence is ubiquitous: TPM EK first (near-universal on target server fleets) — certificate-chain enrollment is prototyped; credential activation completes the phase — NVIDIA Hopper/Blackwell device attestation second (where CC-capable SKUs exist), SPDM component attestation third (as the device ecosystem matures), with Platform Certificate / SCV validation applied opportunistically wherever the customer's procurement included them. Each phase is integration engineering against the attestor seams already in the codebase — new evidence readers and validation policy, not new cryptographic machinery — and each lands behind the existing attestor interface, inheriting its fail-closed contract and its tests.

About. Sovereign Proof, Inc. is a Delaware corporation building hardware supply-chain provenance systems. Michael Bardwell-Scott (CEO) is the architect of the Sentinel platform; Robert Gray III (COO) leads delivery and go-to-market, with prior Fortune-100 program leadership at Ford and Daimler. Contact: michael@sovereignproof.io · sovereignproof.io.

© 2026 Sovereign Proof, Inc. This paper describes the system as implemented at the date of publication; roadmap items are identified as such. It is a technical description, not a compliance opinion, certification, or warranty of fitness for any regulatory purpose.