Chapter 13 — Audit Infrastructure

13.1 Purpose and Distinction from Operational Logging

This chapter specifies the **evidentiary tier**: the tamper-evident, hash-chained audit record backing the Platform's compliance claims — every AI Agent decision tagged by Risk Tier, admissible for FCC and SOX review, retained seven years minimum (per the engagement's audit-log commitments). It is deliberately separate from Loki (§12.3), and the separation is the design:

Property	Loki (operational)	Audit tier (this chapter)
Purpose	Search, debugging, alerting	Evidence, attestation, regulatory review
Retention	90 days	**7 years minimum**
Integrity	Best-effort (filesystem)	**Cryptographic hash chain, externally anchored**
Mutability	Deletable by admins	Append-only; no platform-held credential can delete (§7.3)
Query	LogQL, fast	Sequential verification + indexed retrieval, slow is acceptable

As of 2026-07-18 nothing is built; nothing blocks it either — the storage substrate (`sbs-dash-audit` bucket, §7.3) and the event sources (§13.3) are already specified. This is the **audit layer** of the eight-layer Compartment isolation model (§1.2) and the infrastructure realization of MON-01 (unified audit trail), consumed by Guardian Agents as writers and by SBS/regulators as readers.

13.2 Hash-Chain Architecture

Design: **per-Compartment append-only event chains, batched into signed, anchored segments.**

- **Event record.** Every auditable event is a canonical JSON document: `{seq, ts (RFC 3339 UTC), compartment, actor (agent id | human identity per §9), event_type, risk_tier, payload_hash, prev_hash, entry_hash}` where `entry_hash = SHA-256(canonical_serialization(record minus entry_hash))` and `prev_hash` is the previous event's `entry_hash`. Canonicalization is fixed (RFC 8785 JCS) — hash stability across implementations is what makes verification portable to an auditor's independent tooling.

- **Payload separation.** The chain entry carries the **hash** of the full decision payload, not the payload itself; full payloads (prompts, tool calls, outputs — potentially large, potentially sensitive) are stored adjacent in the same segment object, addressed by `payload_hash`. The chain stays compact and verifiable in isolation; payload disclosure decisions (e.g., what an auditor sees) are separable from integrity proof.
- **Ten chains, not one.** One chain per Compartment (`c01`…`c10`) plus a `platform` chain (identity events, break-glass use, Vault audit summaries, infra changes from §10.4). Per-Compartment chains preserve the isolation model in the evidence itself — Compartment 03's audit history is disclosable to an auditor without touching Compartment 08's — and keep verification parallelizable. Cross-chain ordering, where ever needed, is provided by the anchor layer, not by merging chains.
- **Segments.** The writer service (§13.3) closes a segment every hour or 10 000 events, whichever first: `segment = {chain_id, seq_range, first_hash, last_hash, event_file, payload_file}`, uploaded to `sbs-dash-audit` under `{chain}/{yyyy}/{mm}/{dd}/{segment_id}.tar.zst` with a manifest. Segment manifests are **cosign-signed** (keyless, the CI/service identity — same machinery as §11.5, no new key lifecycle).
- **Anchoring (the external trust root).** Each segment's `last_hash` is anchored outside the platform's own control so that even a full-platform compromise cannot silently rewrite history: (a) appended to a daily **anchor digest** — one line per chain per day — which is (b) cosign-signed and (c) submitted to the **Sigstore Rekor public transparency log**, whose inclusion proof is stored alongside. Rekor gives an independent, publicly verifiable timestamp for free, using tooling already in the stack. The daily anchor digest is additionally emailed to the SBS security alias (§4.4) — a low-tech second anchor in SBS-controlled custody that no platform credential can retract.

13.3 Write Path and Event Sources

- **Audit writer service** — a small dedicated service (one per environment, prod's on `sbsdash-server-prod`, its own compose network `net-audit`): receives events on a VPC/localhost-only endpoint, assigns `seq`/`prev_hash`/`entry_hash`, appends to the active segment on a dedicated DO Volume (write buffer), closes/uploads/anchors segments. It holds the **only** credential with PUT access to `sbs-dash-audit` (§7.3 write-once posture: PUT-only, no DELETE, no overwrite — bucket versioning on as belt-and-braces).
- **Sources** (deduplicated against §12.3 — same events, second delivery, different guarantee): Guardian Agent decision records (the primary stream — every agent action with Risk Tier, per SOW-side commitments); Supervisor routing/resolution events (Medium/High/Critical human decisions, §9.5 supervisor groups); identity events (§9.4 SCIM lifecycle, break-glass use §9.6); Vault audit device (§8.4); infrastructure changes (merged PRs, Prod applies, emergency backports §10.4); egress-denial summaries (§3.4).
- **Delivery semantics, stated honestly:** at-least-once from sources to writer with idempotent dedup on `(source, source_seq)`; the writer's local buffer means a writer crash loses at most the unflushed tail (fsync per event for Guardian decisions — the stream where loss is least acceptable; batched fsync elsewhere). The *chain* structure means loss is always **detectable** (sequence gap) even where not preventable — a gap is an incident (MON-05), never silently renumbered.

- **Backpressure rule:** if the writer is unavailable, Guardian Agents **queue locally and continue enforcing** (availability of the platform does not hinge on the audit path), but a queue older than 15 minutes is a `critical` alert (§12.4) — extended audit outage is treated with the severity of a control failure, because it is one.

13.4 Retention and Lifecycle

- **Seven years minimum**, rolling, per the engagement's FCC/SOX-grade retention commitment; segments carry their close-date and become deletion-eligible only after `close + 7y`, and deletion is a **manual, dual-controlled annual ceremony** (SBS sign-off via Alexandra Del Rey routing + Ledger Hub execution), logged — there is no automated deletion path at all, which is the safest failure mode for an evidentiary store.
- **Storage layout:** hot segments in `sbs-dash-audit` (NYC3), replicated to the SFO3 backup bucket on the §14 schedule — two regions, plus the Rekor/email anchors which survive even dual-bucket loss (anchors prove *what the data was*; replication ensures *the data survives* — both are required, neither substitutes).
- **Capacity honesty:** at plausible agent volumes (10 Compartments × thousands of decisions/day), zstd-compressed JSON runs low single-digit GB/year for chains and tens of GB/year with payloads — Spaces-trivial; the 7-year cost concern is nil. The Volume write buffer is sized for 72 h of disconnected accumulation.
- **Legal hold:** a hold flag on any chain/date-range suspends deletion eligibility indefinitely; holds are themselves platform-chain events.

13.5 Integrity Verification

Verification is continuous and layered — an integrity system that is only checked when someone subpoenas it is not an integrity system:

- **Continuous (automated):** a verifier job on `monitor-servers` (deliberately a *different host* than the writer — the checker must not share the writer's compromise) re-downloads a rolling sample daily: yesterday's segments fully, plus a random historical segment per chain per week. It recomputes every `entry_hash`, walks `prev_hash` continuity across segment boundaries, checks cosign signatures, and verifies Rekor inclusion proofs. Results export as `integrity` job metrics (§12.2); any mismatch is `critical` (§12.4) and a MON-05 incident.
- **Full-chain verification:** quarterly, complete re-verification of all chains end-to-end (compute-cheap at these volumes), timed with the quarterly executive review cadence — the review receives a signed verification report, generated evidence rather than asserted health.
- **Independent verification path (the point of the whole design):** a standalone, dependency-light verifier CLI (single static binary, in the repo, itself cosign-signed) that an SBS auditor or regulator can run against exported segments **without any Ledger Hub system access**: input segments + the public Rekor log + the emailed anchor digests → PASS/FAIL per chain. Admissibility rests on this property — the evidence verifies against public roots, not against the operator's word.

- **Tamper response:** verification failure freezes the affected chain's deletion eligibility, triggers incident response (MON-05), and the last-good anchor bounds the affected window precisely — the chain structure turns "was anything altered?" into "these specific sequence ranges after this timestamp are suspect."

13.6 Verification (Build Gates)

Gates (LH-SBS-INST-001): writer reachable only on `net-audit`/VPC (public + cross-Compartment probe fails); Guardian test event → chained entry with correct `prev_hash` linkage (recompute check); segment close → bucket object + cosign signature + Rekor inclusion proof all present; PUT-only posture proven (writer credential DELETE attempt fails, §7.7 carried gate); induced-tamper drill — modify one byte in a staged segment copy, continuous verifier alerts within its cycle; sequence-gap drill — drop an event in staging, gap detected and alerted; independent verifier CLI produces PASS on clean export and FAIL on the tampered copy, run from a machine with zero platform access; queue-age alert fires in writer-outage drill; anchor email arriving at SBS alias confirmed; 7-year lifecycle rules + hold flag present in IaC with no automated deletion path (code review gate).

Revision #1
Created 2026-07-18 11:56:06 UTC by SBS Admin
Updated 2026-07-18 11:57:02 UTC by SBS Admin