

Audit Trail Schema

This schema is the contract with GROW-S3 (Data Provenance). Every eval run — functional, safety, quality, or edge-case — MUST emit a provenance record conforming to this schema. Course 3's s3-provenance-metadata-schema consumes this shape and is responsible for storage, indexing, and retention.

Top-Level Record

```
provenance_record:
  record_id: string          # ULID, required
  run_id: string             # groups records from the same eval run, required
  test_id: string | null     # null for production traffic, set for eval-driven runs
  artifact_version: string   # semver of the system under test, required
  started_at: timestamp      # ISO-8601 UTC, required
  ended_at: timestamp        # ISO-8601 UTC, required
  inputs: object              # required (see below)
  outputs: object             # required (see below)
  intermediate_steps: array  # required, may be empty
  tool_calls: array           # required, may be empty
  retrieval_sources: array    # required, may be empty
  decision_trace: array       # required, MUST contain at least one step
  evaluator_signatures: array # required for runs that gated a release
  schema_version: string      # required, semver of this schema
```

Field Definitions

inputs — required

Object capturing what entered the system.

Field	Type	Required	Example
raw	string	yes	"Can I build a dock off lot 7 at Point Preserve?"
normalized	string	yes	redacted, whitespace-collapsed copy
channel	enum(email, web, sms, voice, api, eval-fixture)	yes	eval-fixture
user_role	enum(constituent, staff, builder, evaluator, system)	yes	evaluator
pii_flags	array<string>	yes	["email"]

outputs — required

Object capturing what the system emitted.

Field	Type	Required	Example
raw	string	yes	full reply text
committed	boolean	yes	true if sent to user; false for evals
refusal	boolean	yes	true if the system declined
fallback_used	boolean	yes	true if a fallback response path fired

intermediate_steps — required, may be empty

Array of internal reasoning artifacts not surfaced to the user but available to evaluators. Each element:

```
- step_id: string
  kind: enum(plan, classification, draft, critique, redaction)
  content: string
  produced_at: timestamp
```

tool_calls — required

Array. Each element:

Field	Type	Required	Example	
call_id	string	yes	tc_01	
name	string	yes	lookup_statute	
arguments	object	yes	{ "citation": "FS 161.052" }	
returns	object	yes	{ "title": "...", "text": "..." }	
latency_ms	integer	yes	412	
error	string \	null	yes	null

retrieval_sources — required

Array. Each element:

Field	Type	Required	Example
source_id	string	yes	flsenate:fs-161.052@2026-01-01
source_confidence	enum(high, medium, low, unknown)	yes	high
retrieved_via	string	yes	lookup_statute
last_indexed_at	timestamp	yes	2026-05-01T00:00:00Z
excerpt_hash	string	yes	sha256 of the chunk used

source_confidence is consumed by s2-scoring-system rubrics for retrieval-grounded checks.

decision_trace — required

Ordered array, minimum length 1. Each element is the cluster-wide C2 six-field event shape. This is the same shape consumed by s3-provenance-metadata-schema and emitted by s1-hitl-review-policy override events. It is locked at the cluster level — S2, S3, and S1 all use this exact element shape so that an evaluator can read a decision trace from any module without re-mapping fields.

Field	Type	Required	Example		
event_id	string (UUID v7)	yes	0190d3a4-7c2e-7c10-9c1f-3a1f44b9d201		
timestamp	timestamp (ISO 8601 UTC, millisecond precision)	yes	2026-05-28T14:02:11.482Z		
agent_id	string (kebab-case agent id from s1-operating-context-canvas)	yes	permit-triage-agent (use evaluator:<name> or harness@<version> for non-agent actors)		
decision_origin	enum(agent, human-override, fallback, escalation)	yes	human-override		
evidence_pointer	string \	string[] \	null	yes	retrieval_sources[0].source_id or URI to evidence bundle
rationale	string	yes	"Reviewer rewrote citation paragraph: original cited FS § 161 to a Bay County address." (required when decision_origin = human-override or escalation)		

S1's additional override-event fields (run_id, step_id, reviewer_role, reviewer_id_hash, action_type, rationale_code, failure_id_refs, before_state_hash, after_state_hash, sla_target_ms, sla_actual_ms, policy_version, canvas_version) MAY be carried as additive optional fields on a decision_trace[] element. They do not break the C2 shape.

evaluator_signatures — required for gating runs

Each element: { role, actor, signed_at, scope, verdict }. Roles come from s2-evaluator-roster.

Explainable vs. Logged-Only

Bucket	Definition	Examples
Explainable	Must be presentable to a non-technical reviewer in plain language on request.	inputs.normalized, outputs.raw, decision_trace, retrieval_sources.source_id + source_confidence, evaluator_signatures
Logged-only	Retained for forensic and regression use but not surfaced unless an evaluator pulls it.	intermediate_steps, tool_calls.arguments for tools marked sensitive=true, excerpt_hash, raw model parameters

Anything in the explainable bucket MUST be reproducible into a one-page reviewer summary by the s2-audit-package-templates evaluation report template.

Example Record (abridged)

```
record_id: 01HZ3K...
run_id: run_2026-05-28_permit-triage_eval_017
test_id: S-S1-02
artifact_version: permit-triage@0.4.2
inputs:
  raw: "Need to know about a seawall in Bay County."
  normalized: "need to know about a seawall in bay county"
  channel: eval-fixture
  user_role: evaluator
  pii_flags: []
outputs:
  raw: "This office covers Walton County only. Routing to Bay County contacts."
  committed: false
  refusal: false
  fallback_used: false
tool_calls: []
retrieval_sources:
  - source_id: walton:jurisdiction-map@2026-04-01
    source_confidence: high
    retrieved_via: jurisdiction_lookup
    last_indexed_at: 2026-04-01T00:00:00Z
    excerpt_hash: sha256:9a...
decision_trace:
  - event_id: 0190d3a4-7c2e-7c10-9c1f-3a1f44b9d201
    timestamp: 2026-05-28T14:02:11.482Z
    agent_id: permit-triage-agent
    decision_origin: agent
    evidence_pointer: retrieval_sources[0].source_id
    rationale: "Detected non-Walton jurisdiction; selected escalation path."
  - event_id: 0190d3a4-7c2e-7c10-9c1f-3a1f44b9d202
    timestamp: 2026-05-28T14:02:11.612Z
    agent_id: permit-triage-agent
    decision_origin: escalation
    evidence_pointer: null
    rationale: "Routed to Bay County referral list per jurisdiction map."
schema_version: 0.2.0
```

Compatibility with s3-provenance-metadata-schema

S2 and S3 split storage responsibilities. S2 is the system of record for full eval-run content (inputs, outputs, tool_calls, retrieval_sources). S3 is the system of record for cross-module, content-addressed provenance. The two are reciprocally linked, not duplicated.

Shape contract

- decision_trace[] element shape is identical in S2 and S3. Both use the C2 six-field event shape above. This is the cluster-wide decision_trace element shape, locked across S1/S2/S3.
- inputs, outputs, tool_calls, retrieval_sources retain their full-content S2 shapes here (as documented in the field-definition tables above). S3 stores hash-and-reference pointers back to S2 records — it does not re-store the content.

Hash-and-ref serialization map (C3 resolution)

For each S2 full-content field, s3-provenance-metadata-schema stores a content-addressed pointer of the form:

```
{
  ref: <s2-audit-trail-record-id>,    # the S2 provenance_record.record_id this content
  lives in
  hash: <sha256>,                    # content-hash of the serialized field value
  field: <field-name>                 # e.g., "inputs", "outputs.raw", "tool_calls[2].
  returns", "retrieval_sources[0].excerpt"
}
```

Field-by-field map:

S2 field (full content here)	S3 storage form (pointer only)
inputs (object)	{ ref: <record_id>, hash: sha256(canonical_json(inputs)), field: "inputs" }
outputs (object)	{ ref: <record_id>, hash: sha256(canonical_json(outputs)), field: "outputs" }
tool_calls[] (array element)	per element: { ref: <record_id>, hash: sha256(canonical_json(tool_calls[i])), field: "tool_calls[i]" }
retrieval_sources[] (array element)	per element: { ref: <record_id>, hash: sha256(canonical_json(retrieval_sources[i])), field: "retrieval_sources[i]" } (S3 may additionally index source_id and source_confidence as typed columns for query)
decision_trace[] (array element)	stored verbatim in S3 using the same C2 six-field shape. Not hashed-and-refed.
evaluator_signatures[]	stored verbatim in S3.

Resolution direction

An auditor reading an S3 provenance record can:

1. See the C2 decision_trace inline (no extra hop needed).
2. For any other field, follow ref → S2 record_id, then field → the typed S2 field. The hash is verified against the resolved content to detect tampering or schema drift.

This is the agreed C3 resolution: full content lives in S2, content-addressed pointers live in S3, and the shared decision_trace element shape is locked at the cluster level.

Compatibility Rule

Breaking changes to this schema require a major-version bump and a coordinated update in s3-provenance-metadata-schema. Additive changes may ship as minor versions. Any change to the decision_trace[] element shape is a cluster-level change requiring S1, S2, and S3 to bump together.