

DOCUMENTATION PAGE

Governed Evidence-Chain Failure Triage

Governed Evidence-Chain Failure Triage

Purpose

This guide gives agent-orch maintainers and workflow operators a bounded way to triage an evidence-chain failure without guessing at its cause or owner. It separates a product repair from a validator defect, missing evidence, an environment dependency, and a Lee-authority decision. The reader should leave with the evidence to inspect, the owning route, the smallest immediate remediation, and a clear escalation boundary. The sealed authority is a source boundary only: revision `2026-08-08` lists the three public-safe logical source IDs and says `employee_may_modify: false`; it does not authorize changing governance, validators, routing, activation, approval, publication, or release state. Product conclusions and platform attestations must remain separate.

Classification Procedure

Begin with the original evidence record, preserving its journey or work-item identifier, exact command, exit status, observed output, source references, acceptance traces, retry or recurrence information, and affected work item. Do not replace a failed observation with a later pass. Next inspect `sources/authority.json` and match its exact source IDs to `registries/sources.json`; verify locator, revision, digest, authority class, and `public-safe` disclosure without copying private governance or runtime material. Then ask, in order: does a deterministic product result name a repairable input; does independent evidence show the validator itself is wrong; is a required evidence field absent; does the observation require an external environment; or is the remaining action a human authority decision?

Choose the narrowest class supported by observed evidence, not the most severe possible explanation. A missing field is missing evidence, not a product defect. `Operation not permitted` from the known

localhost reader attempt is an environment-specific evaluation blocker, not proof of a recurring harness defect. A manifest command is a re-executable claim, not proof that a platform executed it. Record the class, inspected evidence, owner route, immediate remediation, unresolved fields, and escalation condition together so another reader can reproduce the classification.

The required synthetic journeys are intentionally independent of that localhost observation. They use allowlisted deterministic fixture checks over the guide and its journey manifest and expect exit code `0` ; that result only shows that the local fixture contract was read and checked. It does not overwrite the preserved smoke attempt, whose observed command remains `python3 tests/check_documentation_smoke.py` with exit code `1` . The known environment-blocker and Lee-authority journeys retain that smoke command so their environment evidence stays distinct from the passing synthetic checks.

Failure Classes

The five classes below are mutually useful routing labels, not permissions to cross an authority boundary. Apply one only when the stated evidence exists; otherwise retain the observation as unresolved and request the missing record.

Product repair

Inspect the deterministic product result, the affected Markdown metadata, the matching page registry entry, registered source references, journey traces, and any projection mismatch. Use this class only when the evidence identifies a concrete product-input defect such as malformed structure, an unresolved reference, missing registration, or stale generated projection. The owning route is the product documentation or workflow maintainer. Immediately repair only the earliest failed product prerequisite, retain the original failure, and regenerate output from source when needed. Escalate when the source or authority boundary is inconsistent, no bounded product fix can address the observed result, or the product owner remains unresolved; never edit sealed authority to make a validator pass.

Validator defect

Inspect the exact validator command and version or policy context, its complete result, the product contract it claims to enforce, and an independent reproduction or known-good fixture. Use this class only when those records support an inconsistency in validator behavior, rather than a malformed input or a

missing execution record. The owning route is the validator maintainer or platform boundary with authority over that validator. Immediately preserve the contradictory observations, stop treating the disputed result as a product conclusion, and request an independent validator review or corrected run. Escalate when validator authority, independent review, or a required gate capability is unavailable; do not rewrite content, claims, source records, or the authority seal to accommodate an unproven validator interpretation.

Missing evidence

Inspect the manifest, evidence packet, and source map for the journey ID, non-empty allowlisted command, integer exit status, observed output, source references, `traces_to`, retry or recurrence evidence, and affected work item. Use this class when the required record is absent, incomplete, stale, or conflicting and no supported failure cause can yet be established. The owning route is the evidence-producing journey, evaluator, or workflow that was required to capture the fact. Immediately mark the conclusion unresolved, preserve the partial record, and request a permitted capture with the exact allowlisted command; do not infer a pass, defect, retry count, or repair target. Escalate only when the missing owner cannot supply the evidence or a governed decision is blocked after the permitted request, carrying the missing fields and original observation into that escalation.

Environment dependency

Inspect the exact command, exit status, observed output, attempt count, reader service or network dependency, and platform-owned execution record. Use this class when the failure is caused by a required runtime capability outside the product source, such as the preserved `Operation not permitted` result from the evaluator environment denying localhost socket access. The owning route is the evaluator or platform environment owner. Immediately classify the observation as an environment blocker, retain it verbatim, and route a permitted follow-up evaluation where the required dependency is available. Escalate when the follow-up establishes a persistent capability gap or when the platform owner cannot provide the required attestation. One denied attempt does not establish a recurring product harness defect or authorize product mutation.

Lee-authority decision

Inspect the completed product conclusion, retained evidence chain, sealed authority metadata, `questions-for-the-human.md`, and any candidate-only boundary. Use this class when product repair and evidence routing are complete but the next action requires Lee's explicit advice or decision, such as

the human-gate question about an unresolved environment-specific evaluation blocker. The owning route is Lee through the separately governed human gate. Immediately keep the candidate or work item parked, state the smallest decision requested, and preserve all evidence without inferring consent. Escalate to that human route only when the packet names the evidence and decision boundary; do not turn readiness, a validator pass, or silence into approval, pilot authorization, promotion, publication, scheduling, rollback, pointer mutation, or a live-directory write.

Remediation Routes

Route by the evidence boundary rather than by the filename that happens to contain the failure. For a product repair, the content or workflow maintainer owns the smallest source-side correction and the original observation remains beside the result. For a validator defect, the validator owner receives the contradictory command and independent reproduction; product authors make no compensating edit. For missing evidence, the journey or evaluator producer must capture the absent command result, source map, trace, recurrence, and affected work item. For an environment dependency, the platform evaluator must supply a permitted follow-up or an attestation of the capability gap. For a Lee-authority decision, the packet goes to the human gate with a narrow question and a reversible parked default. In every route, record `owner unresolved` when the supplied evidence names no owner, keep generated projections derived rather than hand-edited, and leave publication state unchanged.

Failure class	Evidence to inspect	Owning route	Immediate remediation	Escalate when
Product repair	Deterministic product result, page, registry, source refs, and projection	Product documentation or workflow maintainer	Fix the earliest failed source prerequisite and retain the failed result	The defect is outside the product surface or owner is unresolved
Validator defect	Exact validator result plus independent reproduction or known-good fixture	Validator maintainer or platform owner	Freeze the disputed conclusion and request independent review	Validator authority or independent review is unavailable
Missing evidence	Journey record, command, status, output, refs, traces, recurrence, and affected work	Evidence-producing journey or evaluator	Mark unresolved and request a permitted exact capture	Required evidence cannot be supplied or a gate remains blocked
Environment dependency	Runtime dependency, exact attempt, output, and platform execution record	Evaluator or platform environment owner	Preserve the blocker and route a permitted follow-up	Capability remains unavailable after governed follow-up
Lee-authority decision	Product conclusion, sealed authority, human-question record, and candidate state	Lee through the human gate	Park the work and ask the smallest explicit question	The named human decision is required to resolve the boundary

Escalation Boundaries

Escalation is required only after the record is complete enough to show what was observed, what was checked, and why the current owner cannot resolve the gap within its boundary. Escalate a product issue to

its product owner when a concrete deterministic defect is outside the current bounded repair. Escalate to validator or platform ownership when independent evidence identifies a validator inconsistency or an environment capability gap. Escalate missing evidence to the producer when a required field, command result, recurrence, source reference, or affected work item prevents a supported conclusion. Escalate to Lee only when the packet presents a completed evidence chain and a smallest human decision, not merely because a check was not run.

The known reader-gate observation remains narrow: the one `journey.resolve-reader-journey-harness-decision` attempt used `python3 tests/check_documentation_smoke.py`, exited `1`, and reported `Operation not permitted` because localhost socket access was denied. That is an environment-specific evaluation blocker. It does not establish recurrence, a product harness defect, an affected claim, or a repair target. Preserve the candidate-only and unresolved default until a permitted follow-up supplies those facts. No escalation may modify sealed authority, private governance, validator authority, routing, activation, approval, publication, promotion, scheduling, rollback, a release pointer, or a live directory.

Acceptance Checks

- **AC-1 — Correct classification:** Given a known or synthetic evidence-chain record, the reader identifies the narrowest supported class among product repair, validator defect, missing evidence, environment dependency, and Lee-authority decision; cites the exact evidence inspected; and keeps an unsupported cause, recurrence, defect, or repair target unresolved.
- **AC-2 — Correct owner or remediation route:** The reader names the owning product, validator, evidence-producing, environment, or Lee route and states the smallest immediate remediation, including preservation of the original observation and the required exact command, status, output, references, and traces. An unprovided owner is recorded as `owner unresolved` rather than invented from a filename or generated artifact.
- **AC-3 — Correct escalation boundary:** The reader states when escalation is required, carries the complete evidence chain into that route, and preserves the reversible default when proof is incomplete. The reader does not infer a platform attestation, human approval, promotion, publication, scheduling, rollback, pointer change, or live-directory write from a product check, missing evidence, or one environment-denied attempt.

Provenance: registry-listed (no candidate packet)

Registry Source: `content/governed-evidence-chain-failure-triage.md`