

## DOCUMENTATION PAGE

# Smoke-Runner Manifest Contract

---

## Smoke-Runner Manifest Contract

### Overview

---

This page is the operator-facing contract for the one Internal Knowledge workstream serving Lee and current company operators. Its intended outcome is engineering velocity through a short, reproducible documentation smoke path; that is an intended product outcome, not a measured platform result. The sealed oracle is `tests/smoke_manifest.json`, whose current SHA-256 is `b53bea17c6a278817074fa8d87365d0ad4b329453dc8e1b003c180fa211938c1`. That manifest names the product oracle `tests/test_product_contract.py`, pinned at `76c6b6e1fcaa1e8e2851134c32db89e54d1dc8857f5240dea2d56dea4f75290c`. Operators use those pins, the exact command order below, and the journey manifest to work from sealed inputs instead of rediscovering sources. The page records product evidence only; it does not authorize deployment, publication, promotion, rollback, scheduling, pointer changes, or live-directory writes.

### Validation Contract

---

Before semantic output is trusted, an operator must parse `tests/smoke_manifest.json`, verify its SHA-256 pin, and then follow the manifest's declared smoke sequence in order. The separate `tests/smoke_manifest.agent-orch.json` is an orchestration gate and cannot replace the product oracle or product-side evidence. The product oracle is read-only for this run: if its digest differs, stop and preserve the mismatch rather than changing the expected pin or rerunning an unsealed test.

Run the deterministic preflight in this order:

1. Parse and seal `tests/smoke_manifest.json` at `b53bea17c6a278817074fa8d87365d0ad4b329453dc8e1b003c180fa211938c1`.

2. Verify the pinned oracle digest for `tests/test_product_contract.py` at `76c6b6e1fcaa1e8e2851134c32db89e54d1dc8857f5240dea2d56dea4f75290c`.
3. Run the declared oracle exactly as `python3 -m pytest tests/test_product_contract.py -q` and retain its exit code and failure output as product evidence.
4. Seal `sources/authority.json` at `973d42700fcf5046c8c1b8b408337987537e04d8eec4c742574fb30f7a10c3b6` before interpreting or producing semantic output.

After that preflight, run the bounded projection command `python3 tools/build_projection.py -output /tmp/gee-smoke-index.json` and compare the temporary JSON structurally with `build/index.json`. A mismatch is a stale or unexplained generated projection and must be reported for an authorized producer; this page does not make generated output current by assertion and the operator run does not write a live directory.

The six acceptance obligations are:

- AC-1 seals the parseable smoke manifest before consumption and identifies the sealed product oracle.
- AC-2 generates a bounded projection and compares it with `build/index.json`, reporting any stale output rather than silently repairing it.
- AC-3 seals `sources/authority.json` before semantic interpretation; its required digest is the value stated above.
- AC-4 checks this page under the exact `Overview`, `Validation Contract`, and `Operator Runbook Journey` headings, with reader-useful content under each heading.
- AC-5 requires `journeys/user_journeys_manifest.json` to declare complete, safe, reproducible operator journeys and to cover AC-1 through AC-6.
- AC-6 requires an independent evaluator and an independent reviewer to keep product conclusions separate from platform-owned attestations and to cite an allowlisted deterministic check.

Each writable semantic output therefore needs its own content-bearing check: the page's headings and explanatory meaning are inspected, the journey manifest is parsed and its commands are replayable, and registry references resolve. A successful file-existence check or unrelated command is not enough. These are product conclusions only; validator identity, routing, activation, publication, and other platform attestations remain owned by the governing repository and its platform evidence. A clean product smoke result therefore does not by itself establish approval, deployment, publication, or readiness.

These checks establish a reproducible product conclusion. They do not replace the governing repository's authority contract, routing mandate, validator authority, activation rules, or publication policy.

## Operator Runbook Journey

---

Start at the repository root with the sealed paths named in the manifest. Use

`journeys/user_journeys_manifest.json` as the runbook index: every non-exploratory journey has one exact command, one expected observable result, and a `traces_to` entry for at least one page acceptance criterion. Exercise the journeys in their listed order after the preflight: seal the manifest, confirm the bounded projection path, seal authority, confirm the page contract, confirm the journey bindings, and obtain the independent evidence boundary. The manifest's `command_allowlist` contains the complete command prefixes; copy a journey's `command` verbatim so a second evaluator can reproduce it without guessing arguments or discovering another source.

For diagnosis, stop at the first failed gate and preserve the exact command, exit code, stdout, stderr, and observed digest. A manifest parse or digest failure means the smoke oracle is not sealed: compare the file bytes with the documented digest and do not edit the manifest to satisfy the pin. An oracle digest failure means `tests/test_product_contract.py` is not the pinned test: retain the mismatch and do not treat a later test result as valid. A non-zero pytest result is a product-contract failure: retain the named failing test and its output, then report the failure without converting it into a platform finding. An authority mismatch means semantic interpretation must stop until the sealed authority input is reconciled. A projection mismatch means the bounded output and `build/index.json` differ; report the differing structure or hash and leave generated-output repair to an authorized producer. A page, journey, or review-boundary failure means the product evidence is incomplete, not that approval or publication has occurred. This record-and-stop pattern keeps the failure actionable while preserving the publication boundary.

---

**Provenance:** registry-listed (no candidate packet)

**Registry Source:** `content/operations/smoke-runner-manifest-contract.md`