DOCUMENTATION PAGE

Governed Documentation Cycle Lifecycle

Governed Documentation Cycle Lifecycle

Overview

This internal lifecycle map is for agent-orch maintainers, governed-cycle operators, and independent readers working in the **Internal Knowledge** workstream. Its objective is a reproducible path from bounded selection and authority tracing through semantic authoring, deterministic product checks, independent reading, independent review, and a candidate-only handoff. The outcome is a traceable product-evidence packet, not permission to change governance or to take a governed transition.

Bounded Objective

The bounded objective is to let a new operator trace one product-only cycle in order: record the ordered selection of the bounded **Internal Knowledge** slice; seal and read the authority and public-safe sources; author the semantic page and preserve deterministic validation and projection; replay the fixed smoke journey; capture independent reader evaluation; obtain independent review; and make a candidate-only handoff. The cycle ends with a reversible evidence packet and does not establish approval, activation, promotion, publication, scheduling, rollback, a release pointer, a live directory, or a platform attestation.

Read this map with the `lifecycle contract`, the reader-journey manifest, the `sealed authority`, and the `public-safe source registry`. The page identity is registered in `pages.json`; that registration and this page are synchronized semantic sources. The map records public-safe source IDs and reference metadata only. It does not reproduce private governance material, runtime evidence, or an engine-owned attestation.

Operator Problem

A fresh operator must answer six questions in order: what workstream and scope are selected; which authority and source records may be read; what product content is authored and what deterministic result was actually observed; what an independent reader concluded; whether independent review used a separate route; and what bounded packet may be handed off. A selected workstream is not transition authority. A command in the manifest is a declaration or allowlist entry, not evidence that it ran. A reader conclusion and a product validation result are product-side evidence, not platform-owned attestations. Missing, stale, conflicting, failed, or environment-blocked facts remain unresolved and must carry an owner or permitted follow-up.

Constraints

The sealed authority is read-only. It is revision `2026-08-08`, sets `employee_may_modify` to `false`, and lists exactly these logical source IDs: `source.naming-decision`, `source.documentation-boundary`, and `source.platform-delivery-contract`. Each must resolve to a registry record with its locator, revision, digest, authority class, and `public-safe` disclosure. Only those public-safe references belong in this page's metadata; private governance references and source corpora do not.

The manifest is the route contract. Its command allowlist includes exact commands for parsing the authority, parsing the source registry, running the deterministic product-contract check, and parsing the lifecycle manifest. A declared command is not an observed result. An observed record must retain the exact invocation, exit status, stdout or stderr, time when available, source references, acceptance traces, and output location. If a result is not available, the absence is the evidence gap; prose must not fill it.

The lifecycle ends at a reversible candidate-only boundary. This page does not authorize approval, publication, promotion, rollback, activation, scheduling, release-pointer creation, or writing a live directory. It also does not edit the authority inputs, source registry, journey manifest, generated output, release directories, publication boundary, or platform-owned evidence.

Acceptance Checks

- **AC-1 — Lifecycle orientation:** The operator can state the Internal Knowledge context, audience, bounded objective, six stages, and product-evidence outcome.
- **AC-2 — Authority and disclosure trace:** The operator verifies the sealed authority revision, immutable employee flag, exact three logical IDs, and each registry record's public-safe metadata without copying private material.

- **AC-3 — Deterministic evidence distinction:** The operator uses an exact, non-empty allowlisted command and keeps its observed result separate from a declaration, reader evidence, and platform-owned attestation.
- **AC-4 — Reader evidence completeness:** The reader record names the journey ID, exact command, exit status, observed output, source references, acceptance traces, and output location; failed or unavailable evidence stays unresolved.
- **AC-5 — Independent review separation:** The packet identifies an independent reviewer and separate route, with route details, verdict, findings, and evidence location distinct from author conclusions.
- **AC-6 — Candidate-only handoff:** Only bounded evidence, acceptance traces, authority references, and unresolved findings reach the candidate boundary; no governed transition is inferred.
- **AC-7 — Reconciliation and repair boundary:** Every trace is reconciled, every unresolved fact has an owner or permitted follow-up, and missing or environment-specific evidence does not become a defect, repair target, claim change, or transition.

Lifecycle Stages

The contract groups the journey into six ordered stages. The responsible route and its evidence owner are explicit so a new operator can locate the next record without maintainer assistance. Stage 3 groups semantic authoring with the deterministic validation and projection path; it does not turn generated output or a command declaration into authority. Stages 4 and 5 deliberately separate reader conclusions from review conclusions and from any platform-owned execution or transition attestation.

Order	Lifecycle stage and responsible route	Required evidence artifact	Pass/fail gate	Next handoff
1	**Selection and bounded scope** — mission-selection and new-operator route	Workstream, audience, objective, entry point, and six-stage scope record, traced to the contract and manifest	**Pass:** the record names **Internal Knowledge**, the maintainer/operator/reader audience, bounded product scope, and candidate-only endpoint. **Fail:** keep selection unresolved if scope is broader or authority is implied.	Scoped packet to read-only authority tracing.
2	**Authority and public-safe source trace** — read-only authority route	`sources/authority.json`, `registries/sources.json`, and the source-reference comparison record	**Pass:** revision `2026-08-08`, `employee_may_modify: false`, the exact three logical IDs, and each public-safe registry mapping agree. **Fail:** a missing, stale, conflicting, or non-public-safe record blocks authoring.	Sealed reference metadata to product authoring.
3	**Semantic authoring, deterministic validation, and projection** — product author route followed by deterministic validator and projection inspector	This page, its registered page identity, the exact allowlisted command declaration, an observed command record when available, and projection inspection output	**Pass:** page metadata matches the registered identity, the command record preserves exact status and output, and projection identity is reproducible. **Fail/unresolved:** a declaration without an observed result, a mismatch, or missing projection record remains a product evidence gap.	Product evidence to independent reader evaluation.

Order	Lifecycle stage and responsible route	Required evidence artifact	Pass/fail gate	Next handoff
4	**Fixed smoke replay and independent reader evaluation** — independent reader route	Fixed smoke replay and reader record keyed to the manifest journey with journey ID, exact command, exit status, observed output, source references, acceptance traces, and output location	**Pass:** the reader can trace AC-1 through AC-7 and keeps observations separate from author conclusions and platform attestations. **Fail/unresolved:** missing, failed, or environment-blocked smoke or reader evidence is carried forward with an owner or follow-up.	Reader record and unresolved findings to independent review.
5	**Independent review** — reviewer route separate from production	Review record naming reviewer, route, selected adapter/provider/model/role/policy when applicable, verdict, findings or an explicit empty set, and evidence location	**Pass:** the route is independent and the verdict is inspectable. **Fail:** no review pass is inferred from author prose, reader evidence, or a platform-owned record.	Review record and reconciliation matrix to handoff.
6	**Candidate-only handoff** — governed-cycle operator route	Bounded evidence packet, AC-1–AC-7 matrix, authority references, unresolved findings, owners or permitted follow-ups, and candidate-only disposition	**Pass:** all traces and gaps are reconciled and only the candidate boundary receives the packet. **Fail:** park it when a trace, owner, evidence boundary, or transition distinction is missing.	Reversible candidate-only packet for a separately governed decision.

For an LLM-backed production or review route, the run receipt must freeze the selected adapter, provider, model, role, and policy; production and independent review routes remain separate. Deterministic commands do not invoke an LLM. The page author may describe the evidence required by a route, but may not self-certify the route, its identity, its execution, or its transition result.

Required Artifacts and Gates

Use this matrix when assembling the packet. A required artifact can be a declaration, an observed result, a conclusion, or a platform-owned record; those categories must be labeled rather than merged. The exact command from the manifest is retained beside its result. For example, the deterministic product-contract declaration is `python3 -m pytest -p no:cacheprovider tests/test_product_contract.py -q`. That string alone proves only that the command is allowlisted. A pass requires an observed record with its exit status and output; otherwise the check is unresolved.

Checkpoint	Responsible route	Required artifact and ownership	Pass/fail interpretation	Next handoff
Scope selection	Mission-selection route	Workstream, audience, bounded objective, and entry-point record	**Pass:** the selected slice is Internal Knowledge and the endpoint is candidate-only. **Fail:** do not interpret later evidence for an unresolved scope.	Authority trace.
Authority trace	Read-only authority route	Authority JSON plus source-registry records for the three IDs; public-safe reference metadata only	**Pass:** revision, employee flag, IDs, locator, revision, digest, authority class, and disclosure agree. **Fail:** retain the source gap and stop authoring.	Semantic authoring.
Page identity	Product author route	This file and the existing `registries/pages.json` registration	**Pass:** page ID, title, public visibility, audiences, reading modes, empty claim references, and three source references match. **Fail:** reconcile the page source before any downstream check; this step does not edit the registry.	Deterministic validation.
Deterministic product check	Deterministic validator route	Allowlisted command declaration plus observed command record: exact invocation, exit status, counts or output, time if available, and location	**Pass:** an observed result supports the product conclusion. **Fail/unresolved:** a declaration, absent result, or failed result	Projection inspection and reader route.

Checkpoint	Responsible route	Required artifact and ownership	Pass/fail interpretation	Next handoff
			cannot be represented as a passing validator or platform gate.	
Projection inspection	Deterministic projection route	Generated projection observation tied back to the semantic page and registry	**Pass:** identity and source references project reproducibly. **Fail/unresolved:** generated output is inspected as evidence and never edited as source.	Independent reader evaluation.
Reader journey	Independent reader route	Journey ID, exact allowlisted command, exit status, observed output, source references, AC traces, output location, and unresolved owner/follow-up when needed	**Pass:** all required fields are present and the reader conclusion is bounded to product reading. **Fail/unresolved:** missing, failed, or environment-blocked evidence remains explicit.	Independent review.
Independent review	Separate reviewer route	Reviewer identity, route and route details, verdict, findings or empty finding set, and evidence location	**Pass:** review is independent and inspectable. **Fail:** author conclusions cannot substitute for review, and neither can a platform-owned attestation.	Candidate reconciliation.

Checkpoint	Responsible route	Required artifact and ownership	Pass/fail interpretation	Next handoff
Candidate handoff	Governed-cycle operator route	Reconciled packet, AC-1–AC-7 matrix, authority references, unresolved findings with owners or permitted follow-ups, and candidate-only disposition	**Pass:** only bounded product evidence is parked. **Fail:** keep it parked when a boundary or trace is missing; do not infer approval or readiness.	Separate governed decision, if any.

The evidence labels are strict. A declared command is an allowlist fact; an observed command record is execution evidence. A reader conclusion says what the reader could establish from the page; a product validator conclusion says what the product-side check established. A platform-owned attestation is a separate fact supplied by its owning boundary. None may be substituted for another, and no absent result may be reconstructed from expected output.

Candidate-Only Handoff

The handoff packet contains only the bounded product material needed for a separate decision: the synchronized page identity, the three public-safe source references, authority and registry trace, deterministic command declaration and observed result or explicit gap, projection observation, reader record, independent review record, AC-1–AC-7 reconciliation, and every unresolved finding with its owner or permitted follow-up. The reader record must preserve `journey_id`, `exact_command`, `exit_status`, `observed_output`, `source_references`, `acceptance_traces`, and `output_location`. The review record must preserve reviewer, route, verdict, findings, and evidence location.

Candidate-only means parked, reversible, and incomplete as a governed transition. It is not approval, publication, promotion, rollback, activation, scheduling, release-pointer creation, or a write to a live directory. It is also not a readiness or platform-attestation claim. The operator transfers an unresolved gap exactly as observed, including its owner or permitted follow-up; the operator does not turn a missing reader result, failed review, or environment-specific observation into a defect, repair target, claim change, or pass.

The safe endpoint is singular: reconcile the bounded packet, place it at the candidate-only boundary, and stop. No command in this map authorizes approval, publication, promotion, rollback, activation, scheduling, release-

pointer creation, or live-directory writes. Those actions, if ever considered, belong to separately governed authority and evidence boundaries outside this page.

Operator Trace

Follow this trace in order and keep each conclusion beside the evidence that supports it:

1. **Select and scope.** Record the Internal Knowledge workstream, intended audience, bounded objective, contract entry point, and candidate-only endpoint. Selection is context, not permission.
2. **Read authority.** Parse `sources/authority.json` and `registries/sources.json`; verify revision `2026-08-08`, `employee_may_modify: false`, the exact three logical IDs, and the public-safe locator, revision, digest, authority class, and disclosure for each.
3. **Author and check.** Compare this page's metadata with its registered page identity. Keep the exact deterministic command declaration separate from its observed record, inspect projection output without editing it, and leave any missing result unresolved.
4. **Replay and read independently.** Replay the fixed smoke journey using the journey manifest's exact command, then record the journey ID, command, status, output, source references, AC traces, and output location. State the reader conclusion only from that record; do not call it a platform attestation.
5. **Review independently.** Route the packet to an independent reviewer, preserve route details when applicable, and record verdict, findings, and evidence location. A reader conclusion and an author conclusion do not become review evidence merely by repetition.
6. **Reconcile and hand off.** Check AC-1 through AC-7, attach an owner or permitted follow-up to every unresolved fact, and transfer only the candidate-only packet. Stop before approval, publication, promotion, rollback, activation, scheduling, release-pointer creation, or live- directory writes.

The trace is complete when another operator can identify what was declared, what was actually observed, what the reader concluded, what the reviewer found, which facts remain platform-owned, and why the endpoint is candidate only. It is not complete when a command declaration is mistaken for a run, reader prose is mistaken for an attestation, or a missing fact is filled with an assumption.

Provenance: registry-listed (no candidate packet)
Registry Source: `content/governed-documentation-cycle-lifecycle.md`