



Governance & Lifecycle Pack

What this is. The system-of-record rules, as ratifiable policy: content standards, the lifecycle model with volatility-tiered review, the ownership model, and the decision log. Four fill-in templates in `policies/` plus this guide on how to adopt them without creating a bureaucracy.

When you need it. Before your corpus is big enough to rot — which is smaller than you think. If you have 50+ articles or any AI touching content, you're already late in the best possible way: adopt this week.

Which maturity level it serves. The floor under Levels 3–5; the difference between Level 1 ("a KB exists") and everything above it.

1. Governance, promoted from chore to architecture

In classical KCS, content health was the vegetables — the Evolve Loop work everyone agreed mattered and nobody staffed. In the AI era it's the load-bearing wall, because machines now consume your knowledge at face value: an ungoverned article isn't just unhelpful, it's an ingredient in a confident wrong answer delivered with citations.

The good news: governance at rung 1 is small. Four short policies, one log, and the discipline to keep them. Everything in this module fits on a handful of pages *by design* — a policy nobody reads governs nothing.

Adoption path (one afternoon): fill the bracketed blanks in the four `policies/` templates → strike what doesn't apply → have the support leader and the curators ratify them (a meeting, not a committee) → put the review date on the calendar. Revisit quarterly, adjusting thresholds against what the decision log and Module 6's dashboards actually show.

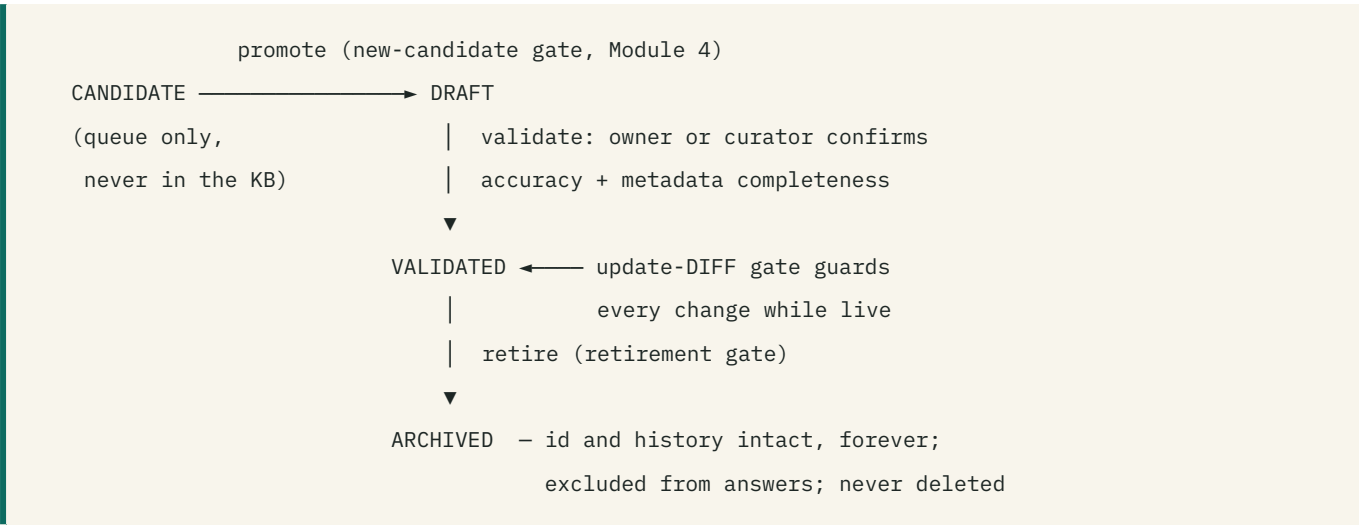
2. Rules and records: the two layers

Keep separate:

- **The rules layer** — what *should* be true: the four policies in this module (standards, lifecycle, ownership) plus your thresholds (validation quorum, review cadences, archive triggers).
- **The records layer** — what *did* happen: the append-only decision log (`policies/decision-log-template.md`) of every gate disposition, lifecycle transition, and policy override.

The records layer is how the rules layer evolves: when the quarterly review finds a rule being constantly overridden, the rule is wrong — change it deliberately instead of eroding it silently. Overrides are permitted; *unlogged* overrides are the only governance sin.

3. The lifecycle in one view



Three states, four transitions, every transition gated and logged. Only `validated` content grounds answers or displays as trusted — this single rule is what makes the whole system safe to put an AI on top of.

Freshness is signal-based, not calendar theatre. Articles come up for review when signals say so — usage decline, negative feedback, contradiction reports, product changes, a passed review date. The `volatility` field sets the *backstop* cadence so nothing can silently outlive its truth; the signals do the real work. Tiers and triggers are in `policies/lifecycle-policy.md`.

4. What’s in `policies/`

File	What it sets	Ratified by
<code>content-standards-policy.md</code>	What may enter the record: types, language, metadata, and the probabilistic-write rule	Support leader + curators
<code>lifecycle-policy.md</code>	States, transitions, who may move what, volatility cadences, signal triggers, archive rules	Support leader + curators
<code>ownership-policy.md</code>	Every article owned; owner duties; orphan sweeps; departure handoffs	Support leader
<code>decision-log-template.md</code>	The append-only records layer: format, retention, and the override rule	— (it’s a habit, not a policy)

What this module assumes from its siblings: articles typed per Module 2; gates per Module 4 (the lifecycle’s transitions are those gates). The decision log is the raw feed for Module 6’s governance metrics — % of corpus within review date, drift backlog, override rate.

Contextkeeping is a KCS-aligned operating model. KCS® is a service mark of the Consortium for Service Innovation; this material is not affiliated with or endorsed by the Consortium.