Black Line: Strong Work in Public

A Positive Operating Discipline for Concise, Rigorous Research and Engineering

Daniel Ari Friedman

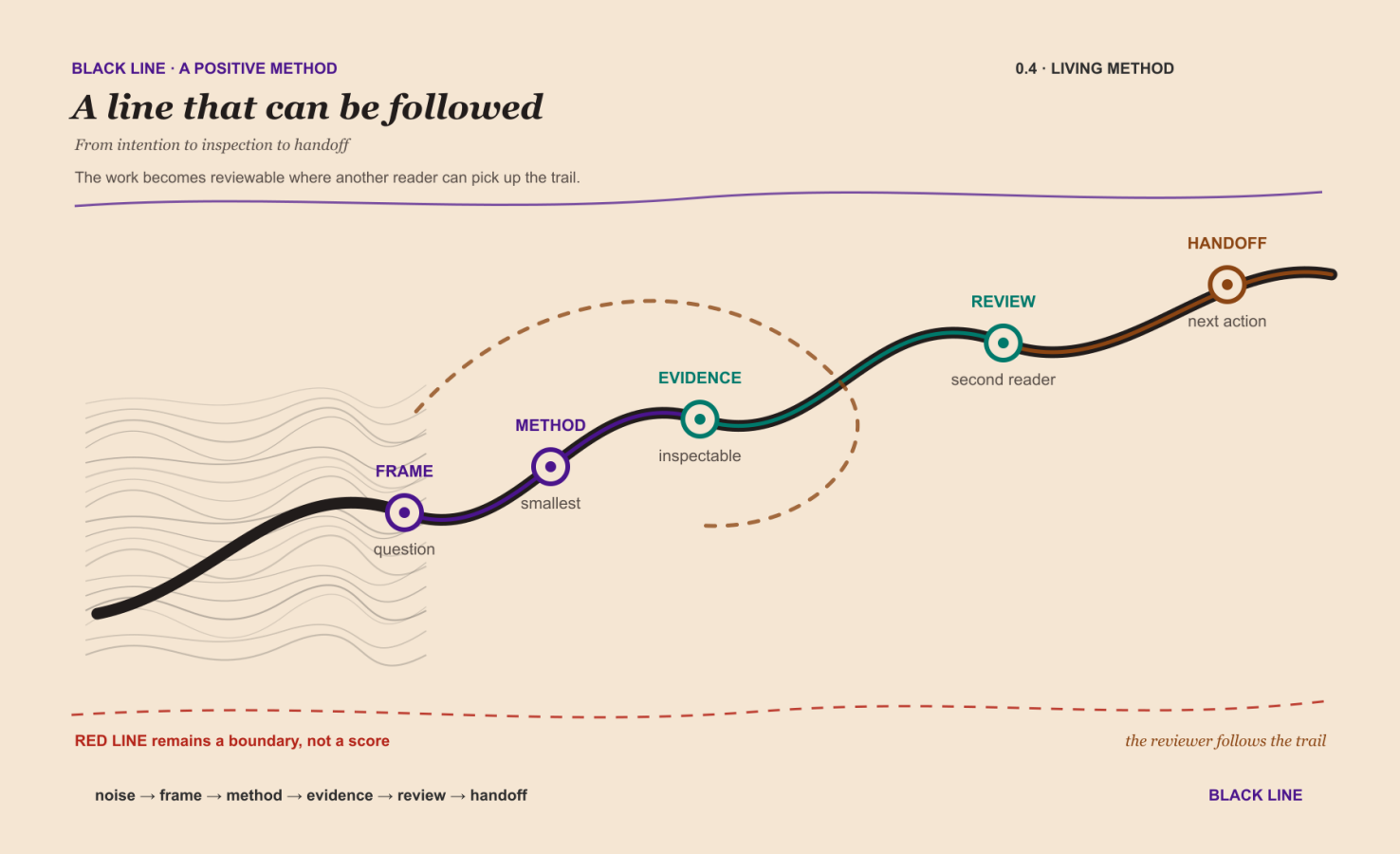
Active Inference Institute

daniel@activeinference.institute

ORCID: 0000-0001-6232-9096

DOI: 10.5281/zenodo.21754236

2026-07-18



Contents

1 Abstract

Black Line is a positive operating discipline for concise, inspectable, revisable work and research. It treats disciplined practice as a set of visible *wires* rather than a claim about character: state the question, trace substantive claims, use the smallest sufficient method, expose failure conditions, verify and steward the result, and leave a handoff another person can recover. The discipline is not new. It collects habits that recur across the philosophy of science, the sociology of knowledge, and the reproducible-research literature, and makes one small operational slice mechanically inspectable [Popper, 1959, Merton, 1973, Goodman et al., 2016].

The instrument is executable and modest. A versioned registry names eleven practices in six practice families, each with coarse review labels a collaborator could inspect. `evaluate_work` validates the review configuration and the registry's own shape, then runs four stages — intake normalization, freshness partition, tag matching, and scoring with aggregation — and returns `ALIGNED`, `NEEDS_EVIDENCE`, `NEEDS_REWORK`, or `OUTSIDE_SCOPE`. Malformed input becomes review notes instead of an exception, a blank description blocks scoring outright, an unscoreable registry fails closed, and an optional staleness window lets dated evidence age into a refresh request rather than a rework. Seven structural invariants check the registry's own shape, and each is shown firing on a planted-bad registry rather than merely passing on the real one. A deterministic digest makes any edit to a wire visible in a diff and travels on each serialized assessment, so the method version behind a review stays recoverable.

What comes back is a prompt for better work, not a safety verdict, an institutional accreditation, or permission to cross the Red Line security boundary. An `ALIGNED` status means only that every required label for every applicable practice was declared as fresh under the chosen review date — never that a source is real or a claim is true. The clean-rerun wire targets computational reproducibility and stops well short of independent replication or inferential agreement. Black Line is the second work in the four-line set: it cross-references Red Line as the refusal boundary, Golden Line as the aspirational thread, and White Line as the record of absence, restraint, and unknowability, and it copies none of their registries, evaluators, or conclusions.

2 Introduction

Red Line answers what a practitioner must refuse. Black Line asks a different question: when a project is allowed to proceed, what makes its reasoning clear enough for another person to inspect and continue?

The answer is not maximal process. A strong method makes its question, evidence, failure modes, and next action visible with as little machinery as the decision allows — a positive discipline that gives work a constructive shape without pretending a checklist can guarantee truth. Black Line resists the belief that adding process is the same as adding rigor: more apparatus can hide a weak question as easily as a strong one. The discipline is to expose the load-bearing parts of the work, not to bury them.

Black Line uses the word *wire* for a bounded practice that carries work from intention to inspectable evidence. A wire can be tested, repaired, or cut. It is not a moral score, and a work can satisfy Black Line while still violating Red Line. Each wire names a coarse review surface a collaborator could inspect — a declared source, a rerun from a clean environment, a recorded null result — so that review begins with visible declarations rather than assumed diligence.

Three commitments organize the rest of the paper. First, I do not present the practices as personal taste: the intellectual-lineage section situates each one against scholarship on falsification, scientific norms, literate programming, craft knowledge, situated action, and reproducible computation, and says where the borrowing stops. Second, the instrument states its own reach: the formal method section gives the exact decision rules the evaluator applies, each bound to a test that can fail, and the limits section executes the attacks that gaming it would use. Third, the boundary is firm — Black Line describes how to work well and never grants permission to cross Red Line. The four-line relationship is mapped in the companion `line_set` work, github.com/docxology/line_set, and restated for this paper in the next section.

The paper states the method; the package makes the same declarations and boundaries repeatable, so a reader can run what the prose describes.

The rest of this paper is organised as follows. Section sec. 5 defines the method and its evidence wires. Section sec. 8 states the evaluator formally as definitions and propositions. The worked examples in Section sec. 9 demonstrate the instrument over real declarations, and Section sec. 10 names the epistemic boundaries that the instrument cannot cross.

3 Relationship to the line set

Black Line is the positive-method work in the four-line set. Each of the four is its own repository: red_line is the personal security boundary and explicit No document; golden_line is the aspirational thread; white_line is the absence ledger: what is unknown, what is withheld under restraint, and what is left open rather than closed with a claim. Black Line does not copy their registries, evaluators, or manuscript claims, and its practice findings never grant permission to cross Red Line.

One word is deliberately shared with Red Line, and a reader holding only this paper should know the other sense exists. Both papers publish a status spelled `OUTSIDE_SCOPE`. Here it marks an attempt outside this discipline's evaluation scope — the registry did not look, because the attempt is not the kind of thing its practices assess. In Red Line it marks a complete, evidenced intake that implicates no red line — that instrument did look and found no prohibition. The spelling is shared by declaration; the two senses are not interchangeable.

A fifth work, line_set, is a thin reader that declares the set and checks that no two lines gave the same spelling to different things. It adds no substantive instrument, and Black Line does not import, depend on, or defer to it.

3.1 Note on the name

The four colors are not decorative. Black, White, Gold, and Red openly echo the stages of the alchemical *magnum opus*, and this paper is the black one: *nigredo*, the blackening — the classical opening stage of dissolution and of confronting what is base or unrefined. The register is strictly symbolic and psychological, in the sense of Carl Jung's reading of alchemy as a map of individuation: the blackening is the honest breakdown that must precede any clarification, the discipline of looking squarely at raw material before dressing it up. Nothing in the name is an empirical, mystical, or causal claim; the alchemy is a metaphor for a posture of work, not a theory of matter or mind. The set's own working order — refuse (Red), then method (Black), then aspire (Gold), then absence (White) — is functional, chosen for how the instruments are actually used, and deliberately does not reenact the opus's sequence (nigredo → albedo → citrinitas → rubedo). Readers who want the full framing, and the single Jung citation that grounds it, should consult the companion `line_set` work at github.com/docxology/line_set, where the alchemical layer is documented once for the whole set.

The same deterministic builder that draws the paper's figures also draws the title-page cover, so the visual argument is versioned with the method rather than commissioned around it. It is reproduced here because a cover printed without its caption states nothing a reader can check:

The narrowing is a metaphor for the posture, not a result. It is not evidence that the work is true, safe, or authorized.

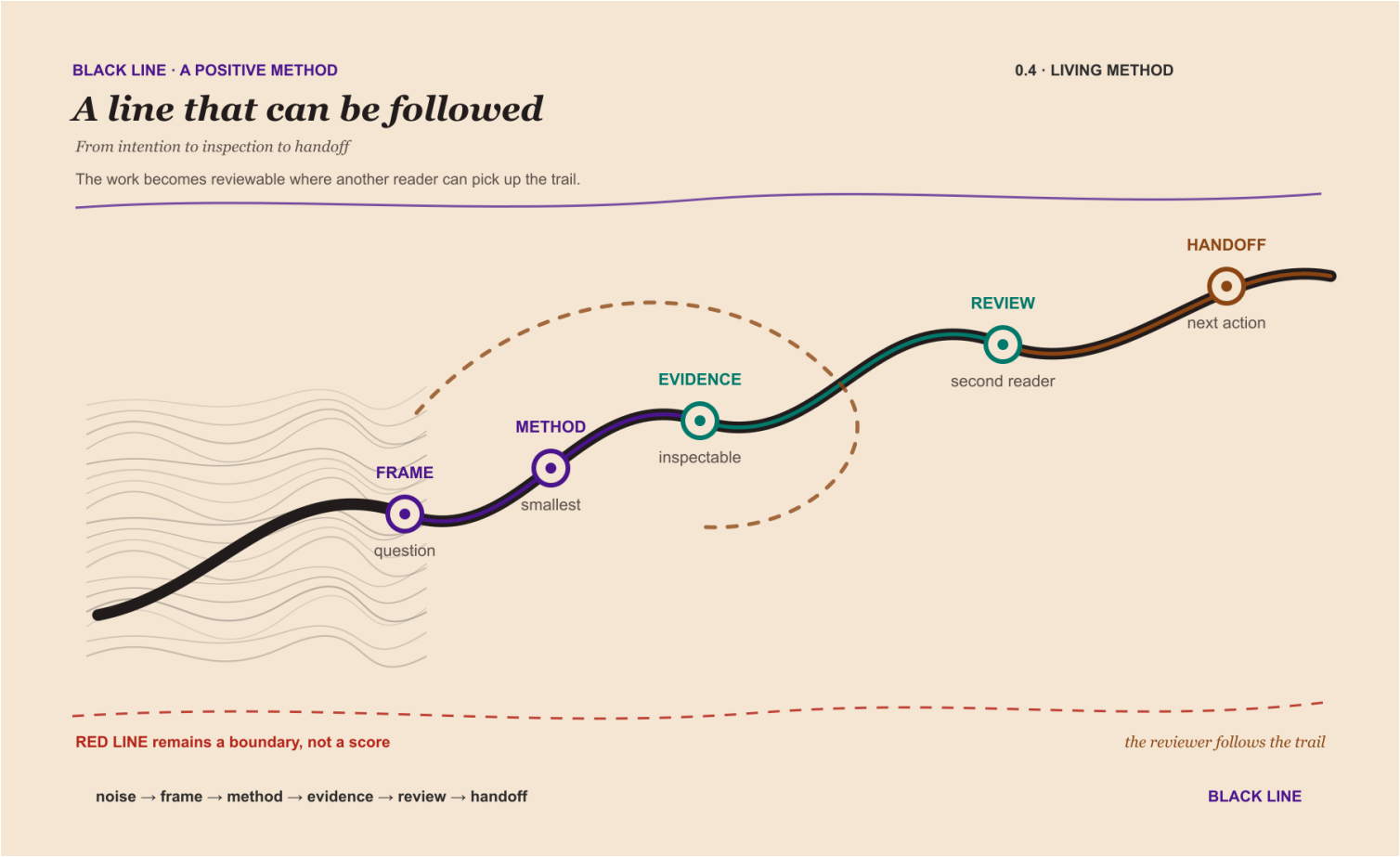


Figure 1: Cover art showing unresolved intention narrowing into a followable review line through method, evidence, review, and handoff.

4 Intellectual lineage: where the practices come from

Black Line does not invent its practices. It collects disciplines that recur, under different names, across the philosophy of science, the sociology of knowledge, the study of craft, and the literature on reproducible computation, and makes them mechanically inspectable. Situating the registry in that lineage does two things: it shows the practices are not personal taste, and it fixes which older idea each wire operationalizes — and, just as often, where the older idea reaches further than any wire here can.

4.1 Framing and falsification

The demand to *state the question before the method* and to *expose failure conditions* is informed by the falsificationist tradition. Popper proposed falsifiability as a criterion for empirical claims: a claim should rule out some observable state so that experience could in principle bear against it [Popper, 1959]. That is a philosophical criterion, not a sufficient condition for good work. Black Line therefore translates only the practical question — what would weaken this result? — into the `question-first` and `failure-visible` practices, keeping a useful demand at the surface of ordinary work without claiming to implement Popper's philosophy wholesale.

Feynman gave the same idea its ethical edge in his 1974 "Cargo Cult Science" address, where the missing ingredient in imitation science is "a kind of scientific integrity …a leaning over backwards" to report everything that might invalidate a result, not only what confirms it [Feynman, 1974]. That injunction is the reason Black Line treats `stated-uncertainty` and `negative-results-kept` as first-class practices rather than optional courtesies: a number without its boundary conditions, or a study that files away only its successes, is exactly the self-deception Feynman warned against.

4.2 Traceability, review, and the norms of science

The practices that concern *sourcing*, *review*, and *stewardship* draw on Merton's account of the ethos of science. His norms — communalism, universalism, disinterestedness, and organized skepticism — describe science as a community that holds claims in common and subjects them to structured, impersonal scrutiny as a social ideal [Merton, 1973]. `source-traceable`, `review-before-reliance`, and `negative-results-kept` are small mechanical echoes of organized skepticism: they make a claim's provenance and its exposure to a second reader into declared, checkable evidence rather than assumed virtue. Stewardship of a shared record has a parallel, not identical, lineage in Ostrom's study of how communities sustain common-pool resources through monitoring, graduated accountability, and locally legible rules [Ostrom, 1990]. A research record can function as a commons for a collaborating group; `versioned-increments` and `review-before-reliance` are small monitoring practices for that setting, not a claim that every record has the same governance structure.

4.3 Traceable prose and the smallest sufficient method

Knuth's literate programming reframed a program as a work of exposition — a document woven so that a human reader can follow the reasoning and the machine can still run it [Knuth, 1984]. Black Line's `concise-handoff` and `source-traceable` practices extend that design problem beyond code: the deliverable should let a collaborator recover purpose, evidence, and next action without private context. The complementary discipline of not over-building is the pragmatic-engineering counsel to prefer the simplest thing that works and to resist speculative machinery [Hunt and Thomas, 1999]; `smallest-sufficient-method` is that counsel stated as an inspectable wire — do not add apparatus whose output cannot change the decision.

4.4 Reproducibility as the load-bearing modern practice

The strongest recent influence is the reproducible-research literature. Peng framed reproducibility — the ability to recompute results from data and code — as a minimum standard that sits between a single study and full independent replication [Peng, 2011]. Sandve and colleagues distilled the working habits that make computation reproducible: track provenance, record exactly how every result was produced, and version the analysis end to end [Sandve et al., 2013]. Wilson and colleagues added the pragmatic layer of "good enough" practices — data management, modest automation, and version control that a working scientist can actually sustain [Wilson et al., 2017]. Black Line's `reproducible-from-clean`, `data-provenance`, and `versioned-increments` practices are small operationalizations of that literature, and its evidence labels (`environment`, `rerun`, `data_origin`, `transform_log`) are named to match its vocabulary. Finally, the discipline of honest presentation — refusing charts and summaries that imply more certainty than the data support — informs the figures in this paper and the `data-provenance` practice alike [Cairo, 2016].

4.5 Reproducibility is not replication

The vocabulary needs one further distinction, and the distinction is the whole scope of the `reproducible-from-clean` wire. Goodman, Fanelli, and Ioannidis separate methods reproducibility, results reproducibility, and inferential reproducibility; the word *reproducible* should not quietly become a synonym for *true* [Goodman et al., 2016]. The National Academies report likewise

distinguishes computational reproducibility — consistent computation from the same inputs, code, methods, and conditions — from replicability in a new study addressing the same question [National Academies of Sciences, Engineering, and Medicine, 2019]. Black Line's clean rerun is deliberately the first, narrower claim: whether another reader can re-execute the recorded procedure, not whether the result generalizes or the inference holds.

The confusion is historical, not careless. Claerbout and Karrenbach coined *reproducible research* for a specific engineering practice — one-command figure regeneration [Claerbout and Karrenbach, 1992] — and the term travelled into fields that already spoke of replication. Plesser traces the cross- disciplinary swap: the same word names re-running the author's artifacts in one literature and an independent redo in another [Plesser, 2018]. Black Line therefore leans on the labels rather than the word — `environment` and `rerun` name Claerbout's narrow property whichever term a reader's field attaches to it.

The replication literature is what makes the distinction consequential rather than pedantic. In the largest coordinated attempt of its kind, 100 psychology studies were re-run with high-powered designs and original materials where available; 36 percent of replications reached statistical significance against 97 percent of the original reports [Open Science Collaboration, 2015]. A survey of 1,576 researchers found a majority had failed to reproduce another scientist's result, and many their own [Baker, 2016]. Nothing in this instrument addresses that. A clean rerun of a recorded procedure cannot detect a design that would not survive a new sample, so an `ALIGNED` finding on the rerun wire is a statement about a declaration and never a prediction about a future study. The instrument sits on the near side of that gap on purpose, and says so rather than letting the shared vocabulary imply otherwise.

4.6 Openness is infrastructure; review is a social act

Open research culture is not produced by a single checkbox. Nosek and colleagues describe a system in which norms, incentives, reporting practices, and access to materials have to work together if openness is to improve credibility [Nosek et al., 2015]. Munafò and colleagues similarly frame reproducibility as a portfolio spanning methods, reporting, dissemination, evaluation, and incentives, with reforms requiring ongoing assessment rather than ceremonial adoption [Munafò et al., 2017]. Black Line translates only a small operational slice of that program: name a source, preserve a negative result, invite review, and leave a rerunnable handoff. The translation is useful because it is small; it is not equivalent to an open-science regime.

4.7 Craft, technē, and legibility as a designed partial view

Polanyi's account of tacit and personal knowledge is the standing counterweight to any instrument that rewards what can be written down [Polanyi, 1958]. The work that matters may include situated judgment, craft skill, embodied attention, or a relationship that no finite evidence vocabulary compresses. Black Line therefore treats legibility as a designed partial view: enough structure for a collaborator to inspect and continue the work, never a claim that the visible record exhausts the knowledge in the practice.

Polanyi is one point in a longer argument about craft, and the rest of it sharpens what a label can hold. Ryle separates knowing how from knowing that: a competence is not a set of propositions, and someone who can recite every rule of a craft has not thereby acquired it [Ryle, 1949]. Aristotle's *technē* already names craft as a distinct kind of knowledge, held in the making rather than in demonstration [Aristotle, 1999]. Dreyfus and Dreyfus press the point developmentally: as skill matures the expert stops consulting the rules a novice depends on, so a written procedure describes the beginner's practice more faithfully than the expert's [Dreyfus and Dreyfus, 1986]. Schön locates professional competence in knowing-in-action and reflection-in-action, which happen inside the doing rather than in a prior specification [Schön, 1983]. Sennett supplies the motivational half — the desire to do a job well for its own sake, and the slow entanglement of hand and judgment — which no checklist installs [Sennett, 2008].

Collins partitions what Polanyi left whole, splitting tacit knowledge into *relational* (tellable but untold), *somatic* (embodied), and *collective* (societally held), and argues only the first can in principle be explicit [Collins, 2010]. An evidence label reaches only the relational kind — and only the portion someone wrote down. A `handoff` label points at a document, not the somatic skill of analysis or a field's collective judgment. The registry is therefore a pointer set for the one externalizable layer of craft, not a compression of it. That is why White Line records what this instrument leaves out, and why a declared trace must never be read as the skill that produced it.

4.8 Verification is not validation

The language of checking needs its own boundary. Oreskes, Shrader-Frechette, and Belitz distinguish the internal assessment of a model or computation from the much harder question of whether it adequately represents a non-closed world; they argue that confirmation is necessarily partial and comparative [Oreskes et al., 1994]. Black Line uses **verification** in the narrow engineering sense of checking whether a declared procedure, invariant, or serialization rule behaves as specified. Its proof-of-detection tests show that a check can fire on a planted counter-example. They do not validate a research claim, a model of the world, or a decision made from the result. The distinction is not a disclaimer added after the method; it is why the evaluator calls its output a review status rather than a truth verdict.

4.9 Handoffs are situated coordination objects

The concise handoff also has a social-scientific lineage. Suchman's analysis of plans and situated action warns that an abstract plan does not determine what people will do in a changing setting [Suchman, 1987]. Star and Griesemer's account of boundary objects shows how a shared artifact can support cooperation across social worlds while remaining locally interpretable [Star and Griesemer, 1989]. Black Line's handoff is deliberately modest in this sense: it preserves the decision, evidence trail, current limit, and next action so another reader can orient themselves, but it does not pretend to transfer the author's tacit skill or eliminate the need for situated judgment. A handoff is a coordination surface, not a complete substitute for collaboration.

4.10 Humility is an operational requirement

Jasanoff's call for "technologies of humility" shifts attention from prediction alone to the unknowns, framing choices, distributional consequences, and questions that a technical system leaves out [Jasanoff, 2003]. That orientation sharpens Black Line's limits: a declaration-status instrument should make its omissions legible, ask who must review what it cannot see, and keep authority separate from procedural completeness. The `stated-uncertainty`, `review-before-reliance`, and `concise-handoff` practices are therefore not a claim to have solved governance; they are small prompts for returning governance to the people and institutions that hold it.

The literature-to-wire map is therefore deliberately asymmetric:

Scholarly concern	Black Line operational slice	Boundary preserved
Falsification and failure visibility	question-first; failure-visible	a declared falsifier is not a successful test
Reproducible computation	clean rerun; explicit data origin	same-input rerun is not new-study replication
Independent replication	none: the rerun wire stops at the same inputs	a clean rerun predicts nothing about a new sample
Open research culture	source traceability; review; negative results	a label is not independent verification
Tacit and situated knowledge	concise handoff; explicit limits	the record is not the whole practice
Craft and technē	declared traces of externalizable work only	a trace is not the skill that produced it
Honest visual communication	evidence matrix; uncertainty notes	clarity does not increase evidential strength
Verification versus validation	proof-of-detection tests; clean reruns	a passing check is not world validation
Situated coordination	handoff; review note; next action	a plan does not replace local judgment
Epistemic humility	uncertainty; limits; review boundary	procedural completeness is not governance

The table is a design map, not a claim that eleven practices capture these traditions. Its purpose is to make the borrowing inspectable and the non-borrowing equally explicit.

4.11 What the lineage does and does not license

Citing these works situates Black Line; it does not borrow their authority. None of these authors claims that following a practice guarantees a true result, and neither does this registry. The lineage explains *why* each wire is worth making visible; the formal method explains *what the evaluator can actually check*, which is only whether the declared evidence is present — never whether the underlying claim is sound.

5 Method: positive wires and observable evidence

The registry contains eleven practices grouped into six practice families. Each practice has a short wire, a tag set drawn from a reviewed five-tag vocabulary, a family, and required evidence labels. The labels are deliberately coarse: the evaluator checks whether labels were *declared and matched*, not whether the underlying artifacts or claims are adequate. This is the load-bearing design choice of the whole instrument — it can expose a review surface, never certify its truth — and everything downstream inherits that limit.

5.1 Six practice families: decision, procedure, record, world, authority

The instrument is easiest to use when six practice families are kept separate. The first is the **decision**: what choice could the work change? The second is the **procedure**: what is the smallest method that could inform that choice? The third is the **record**: what source, observation, transformation, failure, or review note can another reader inspect? The fourth is the **world**: whether the source is authentic, the observation is adequate, the inference is sound, and the claimed effect would survive an independent study. The fifth is **authority**: whether the work is safe, lawful, permitted, or governed by a refusal boundary. Black Line structures the first three and deliberately refuses to certify the fourth or decide the fifth.

This distinction keeps the instrument aligned with the reproducibility literature. A clean rerun with the same data, code, and conditions is a useful computational property, but it is not the same as reproducing a result under a new study or agreeing on the inference drawn from it [Goodman et al., 2016, National Academies of Sciences, Engineering, and Medicine, 2019]. The registry therefore names a review surface, not a truth or authority surface.

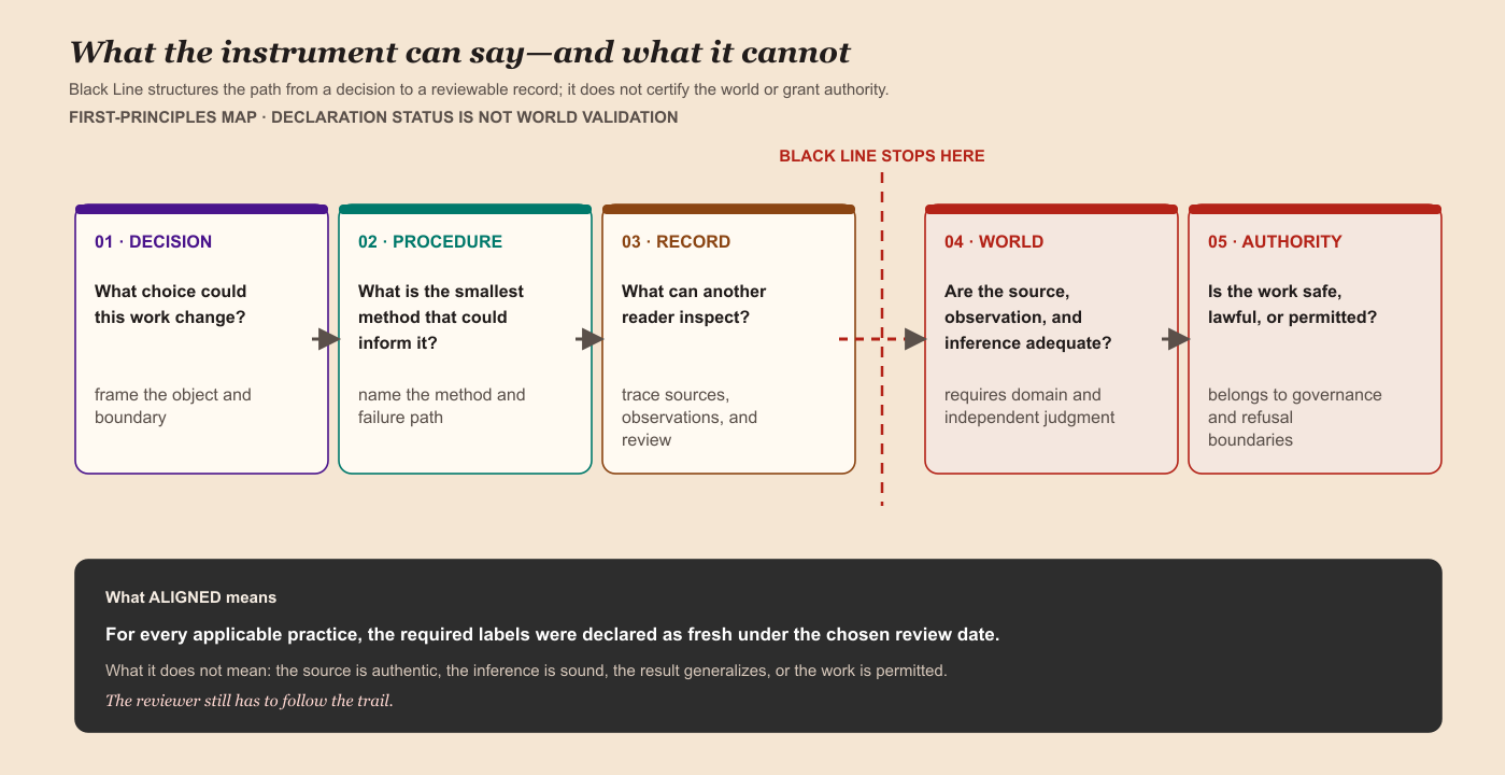


Figure 2: The six-practice-family boundary map separates the decision, procedure, and record that Black Line can structure from world adequacy and authority that require domain, independent, or governance review. `ALIGNED` means fresh declaration coverage for applicable practices, not truth or permission.

5.2 Three claim classes, three evidentiary burdens

The boundary becomes operational when claims are classified before they are written. Black Line distinguishes three burdens:

Claim class	Example	What supports it	What it cannot become
Implementation	"The evaluator returns `NEEDS_REWORK` for a blank description."	Source inspection and a test that exercises the branch.	A claim that the work itself needs substantive rework beyond the declared contract.
Methodological	"A concise handoff is worth requiring."	Design rationale, scholarly lineage, and a stated limitation.	A causal finding that the practice improves outcomes.
World or authority	"The source is authentic" or "the work is permitted."	Domain verification, independent evidence, or governance review.	A conclusion licensed by `ALIGNED`.

This classification prevents a polished figure, a citation, or a deterministic test from silently changing the type of claim being made. The companion claim ledger keeps the same distinction available during maintenance.

`evaluate_work` runs four stages behind two pre-stages, and the formal method section states each of them exactly. In outline: configuration validation rejects an invalid review date or freshness window, and a registry that cannot be scored fails closed; intake normalization records malformed tags, labels, and dates as notes rather than crashing, while a blank or non-text description blocks scoring entirely; dated evidence is partitioned into fresh and stale under the optional window; practices are selected by tag intersection; and each selected practice is scored, with the overall status taking the most demanding per-practice finding, or `OUTSIDE_SCOPE` when no practice applies.

One branch of that scoring is worth stating in prose because it is the one readers misread. The first branch is about the attempt, not the practice: if the attempt declared no usable evidence at all, every selected practice returns `NEEDS_EVIDENCE`. Where none of a practice's own labels are present but other evidence exists, control reaches the last branch instead — `NEEDS_REWORK`, which is missing work rather than an empty declaration. Stale gaps are the third case: a practice whose only remaining gaps are aged observations asks for a refresh, not a rework, and future or unreadable dates are never counted and surface as notes for the declarer to fix.

Inside the scoring stage, the status word is the last step, not the whole state. Each selected practice is first split into typed *evidence surfaces* — its present, missing, and merely-stale required labels, co-present in declared order — and the finding's status and reasons trail are projected from those surfaces (Definition 15 and Proposition 1). The projection selects the most demanding reading for action; the surfaces preserve what it compresses, so strong support and strong resistance on one practice stay readable together instead of collapsing into the projected word. `evaluate_with_surfaces` returns the surfaces beside the identical assessment — one shared staged implementation, never a second evaluator.

This is a deliberately positive counterpart to Red Line's refusal evaluator. It does not inspect prohibited uses, infer hidden semantics, or turn alignment into permission. The evidence vocabulary is a handoff surface for review, not a claim that a source, test, or limitation note suffices by itself. The registry digest travels with each assessment and beside each generated figure, so a changed method is visible in a diff — a drift instrument for method content, never an authentication of evidence.

The same distinction travels across domains. In research, the question may be whether a result should be rerun; in engineering, whether a change is ready for another reviewer; in writing, whether a synthesis can be handed off without private context. The labels change, but the decision–procedure–record boundary does not. Black Line becomes more relevant by staying at that shared layer and leaving domain-specific truth and authorization to the people and systems that actually possess them.

6 Operating protocol: the smallest honest loop

The protocol operationalizes the six practice families of the method.

The instrument is designed around a short operating loop rather than a final score. Begin with the object of work, the decision it is meant to inform, and the constraints or exclusions that make the question meaningful. Choose the smallest method whose output could change that decision. Name the evidence a collaborator could inspect, including what would falsify or materially weaken the result. Run the method, keep the negative path, and leave a handoff that another reader can continue without private context.

The executable protocol has five review actions:

1. **Frame.** Construct a `WorkAttempt` with a non-blank description and reviewed tags. An unknown tag is not an implicit approval; it simply may produce `OUTSIDE_SCOPE` when no practice is reached.
2. **Declare.** Add coarse evidence labels and, when observations can age, dated `EvidenceItem` records. Labels are pointers to artifacts for review, not the artifacts or their proof.
3. **Evaluate.** Pin `as_of`, optionally set a non-negative freshness window, and read every finding's reasons and the intake notes.
4. **Repair or refresh.** Treat missing evidence as a request to do work and stale evidence as a request to repeat an observation. Do not convert either into a success by changing a label alone.
5. **Archive.** Serialize the assessment and retain the source or observation record behind each label. The assessment carries the registry digest so a later reader can distinguish a changed method from a changed result, while the retained artifacts make the declaration inspectable.

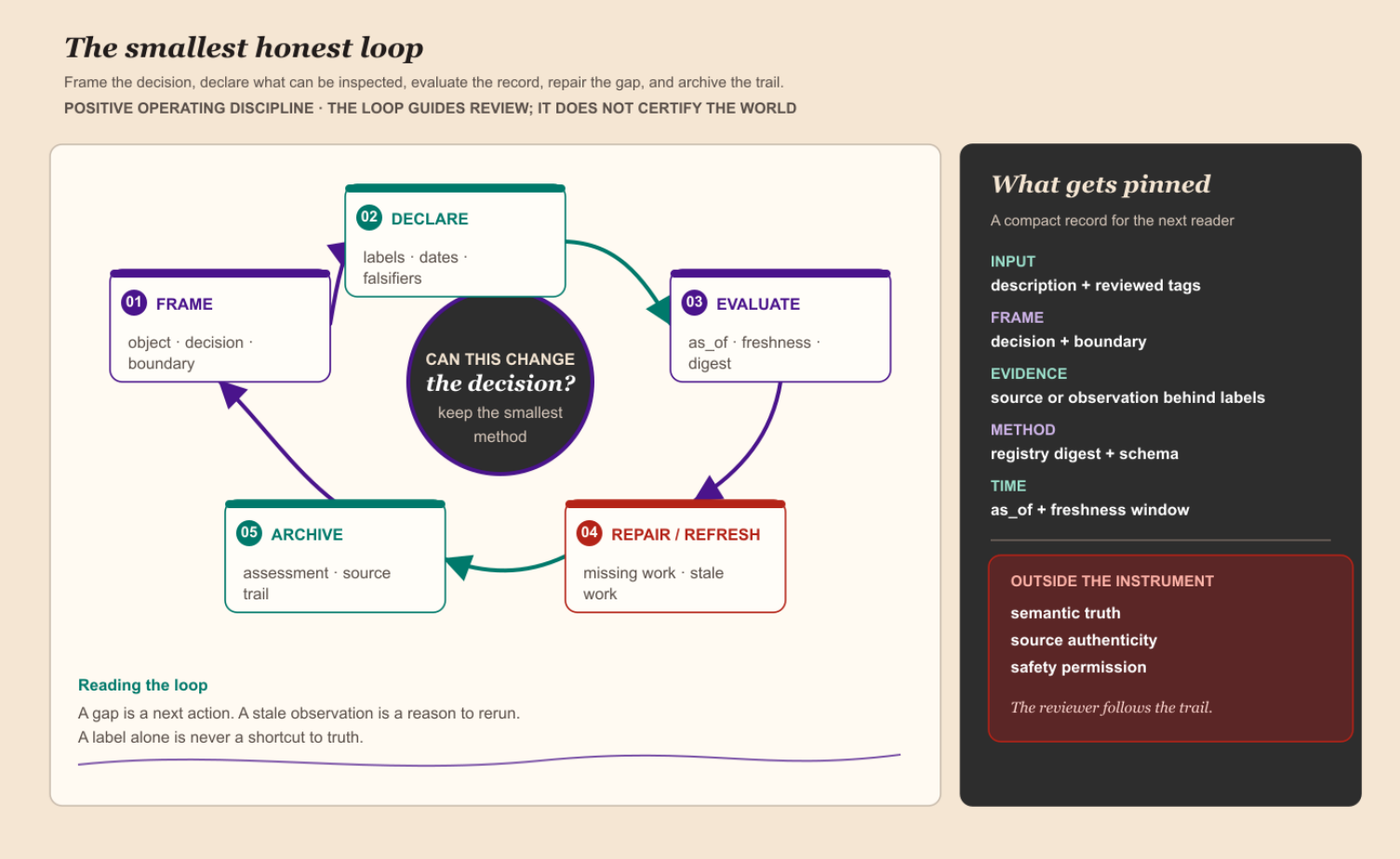


Figure 3: The smallest honest operating loop: frame the decision, declare pointers to inspectable artifacts, evaluate with a pinned date and registry digest, repair or refresh gaps, and archive the trail. The dark panel makes explicit what remains outside the instrument's authority.

This loop is deliberately asymmetric. The evaluator can block an empty work description and can refuse to call an unknown or stale declaration fresh, but it cannot establish semantic truth from a label. The decisive review therefore remains with the collaborator who follows the evidence trail. The evidence matrix figure makes that boundary visible: it is a map of what the evaluator asks for, not a certificate of what the world contains.

7 The Black Line practices

The registry holds eleven practices, each assigned to a craft family (framing, traceability, method, verification, communication, stewardship).

1. **State the question before the method** (framing). Name the decision, object, and boundary before choosing tools.
2. **Make substantive claims traceable** (traceability). Point each substantive claim to a source, local observation, or explicit hypothesis label.
3. **Use the smallest method that can answer the question** (method). Do not add machinery whose output cannot change the decision.
4. **Make failure conditions explicit** (verification). Record what would falsify, break, or materially weaken the result.
5. **Leave a handoff another person can continue** (communication). A collaborator should recover the purpose, next action, and evidence without private context.
6. **Rerun the result from a clean environment** (verification). Treat a result that appears only in one warm environment as provisional until rerun.
7. **Keep changes small and reviewable** (stewardship). Record each increment so its history can be read, reverted, and reviewed.
8. **State uncertainty and limits with results** (traceability). A number without its uncertainty and boundary conditions can overstate what is known.
9. **Record null and negative results** (verification). A documented non-result can narrow the hypothesis space and guide the next attempt.
10. **Invite review before relying on a result** (communication). A second reader can expose omissions the author no longer sees.
11. **Keep data origin and transformations explicit** (traceability). State where each dataset came from and which transformations produced the analyzed form.

These practices are intentionally ordinary. Their value lies in making ordinary discipline inspectable and repeatable across research, software, and writing. The figure below draws the registry in its declaration order, with each card color-keyed to its craft family.

The six practice families exist so the registry can be reviewed for balance: a registry that only rewarded framing but never verification would have drifted from the purpose of making strong work inspectable end to end. Each practice also carries the reviewed tags — drawn from `analysis`, `data`, `engineering`, `research`, and `writing` — that decide which work attempts it applies to. The taxonomy figure groups the practices by family and shows the tags that reach each one; the structural invariants require every family to retain at least one practice and every tag to stay inside the reviewed vocabulary.

The registry's evidence contract is shown separately so that applicability and evidence are not mistaken for proof. The labels below are declarations the evaluator can match; they still require a human or external system to inspect the underlying source, test, observation, or transformation.

7.1 Registry coverage and burden

Applicability is deliberately asymmetric across the tag vocabulary, and the asymmetry is itself a reviewable property of the method. Computed directly from the registry via `coverage_matrix`, 27 of the 55 tag-practice cells are applicable, and the per-tag reach is:

Tag	Practices reached	Required labels	Practices
`research`	8 of 11	16	`question-first`, `source-traceable`, `failure-visible`, `concise-handoff`, `reproducible-from-clean`, `stated-uncertainty`, `negative-results-kept`, `review-before-reliance`

Tag	Practices reached	Required labels	Practices
`analysis`	7 of 11	14	`question-first`, `source-traceable`, `smallest-sufficient-method`, `failure-visible`, `stated-uncertainty`, `negative-results-kept`, `data-provenance`
`engineering`	6 of 11	12	`smallest-sufficient-method`, `failure-visible`, `concise-handoff`, `reproducible-from-clean`, `versioned-increments`, `review-before-reliance`
`writing`	5 of 11	10	`question-first`, `source-traceable`, `concise-handoff`, `versioned-increments`, `review-before-reliance`
`data`	1 of 11	2	`data-provenance`

The declaration burden therefore varies eightfold with tag choice — a `data`-only attempt is scored against a single two-label practice, while a `research` attempt must declare sixteen labels to reach `ALIGNED`. Every practice carries exactly two required labels, so burden scales linearly with reach; the asymmetry lives entirely in how many practices each tag selects.

This creates a coverage-side failure mode the instrument cannot police from inside: *tag minimization*. Because tags are self-declared, a declarer can narrow the tag set to buy a cheaper `ALIGNED` — the status is then true, but over a smaller review surface (this attack is executed, with others, in the adversarial-declarations subsection). The coverage matrix exists so a reviewer can read the status together with the burden it was earned against; whether the declared tags honestly describe the work remains a human judgment, exactly like the evidence behind each label.

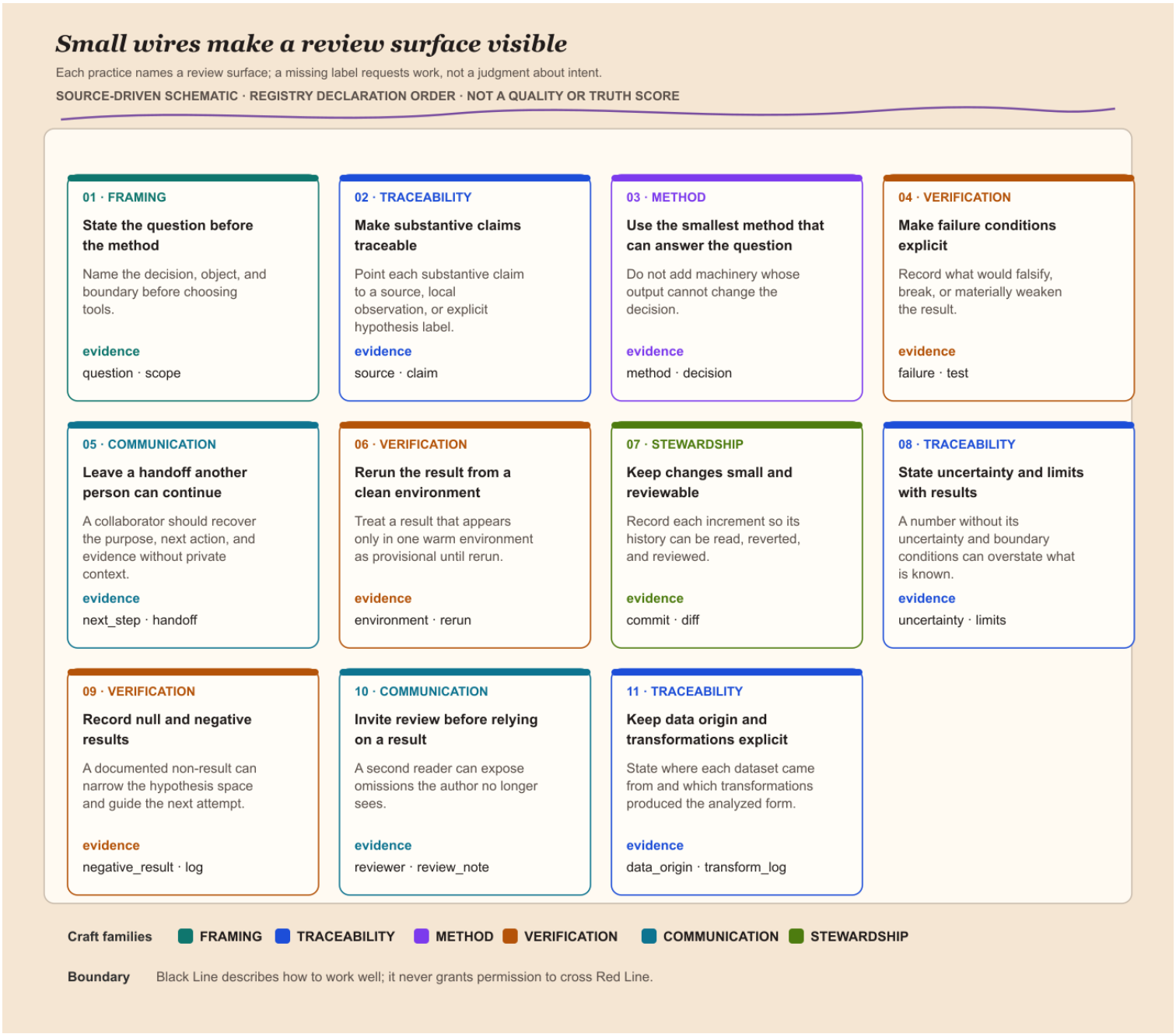


Figure 4: The 11 practices drawn as cards in registry declaration order, color-keyed to the 6 practice families. The order interleaves families and ends at `data-provenance`: it is a declaration order, not a workflow. Each card names a review surface and its required labels; the figure is not a quality or truth score and does not grant permission to cross Red Line.

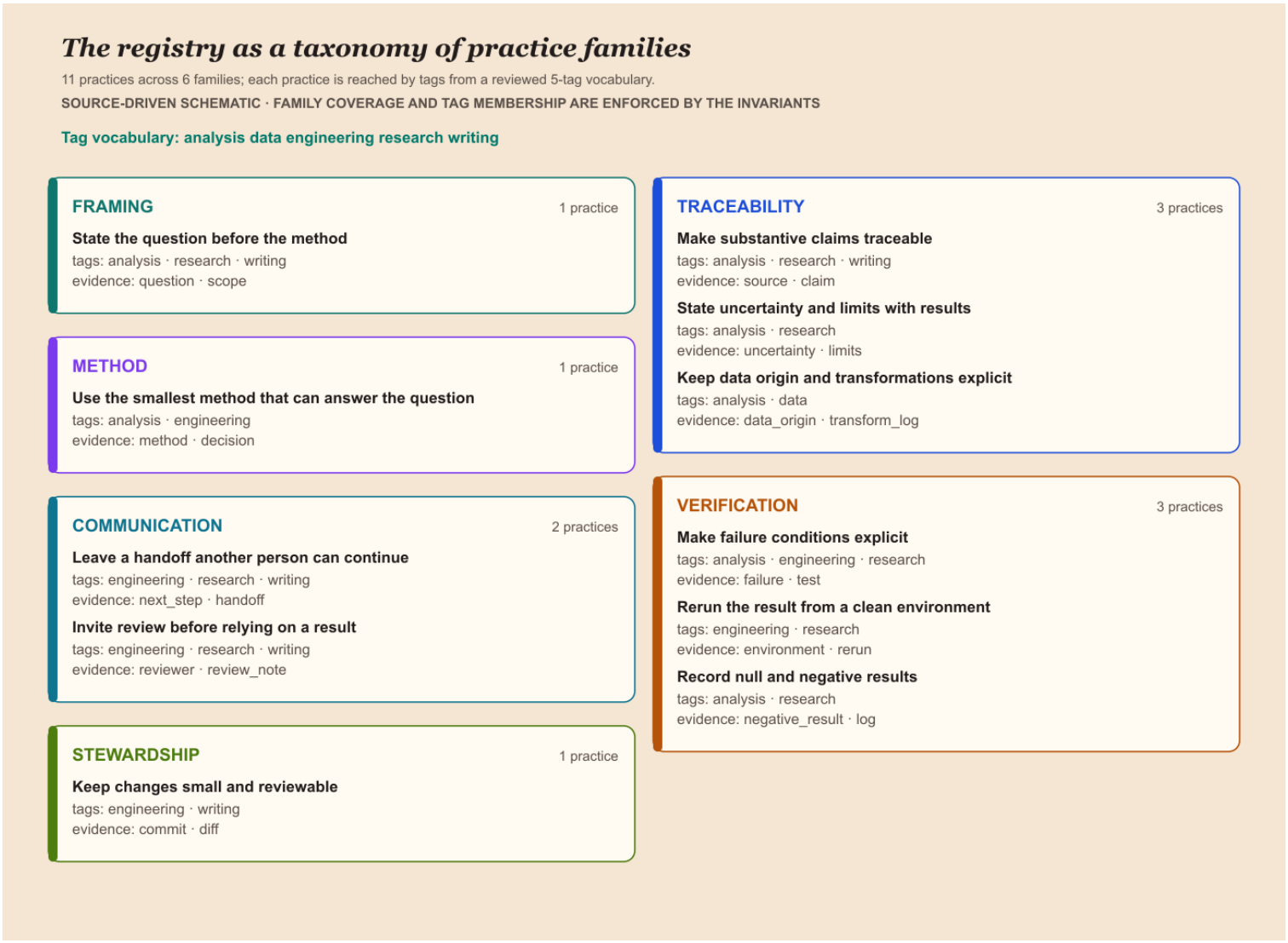


Figure 5: The 11 practices grouped by 6 practice families, with the reviewed 5-tag vocabulary that makes each practice applicable to a work attempt. Family coverage and tag membership are enforced by structural invariants; the taxonomy maps reachability, not evidence quality.

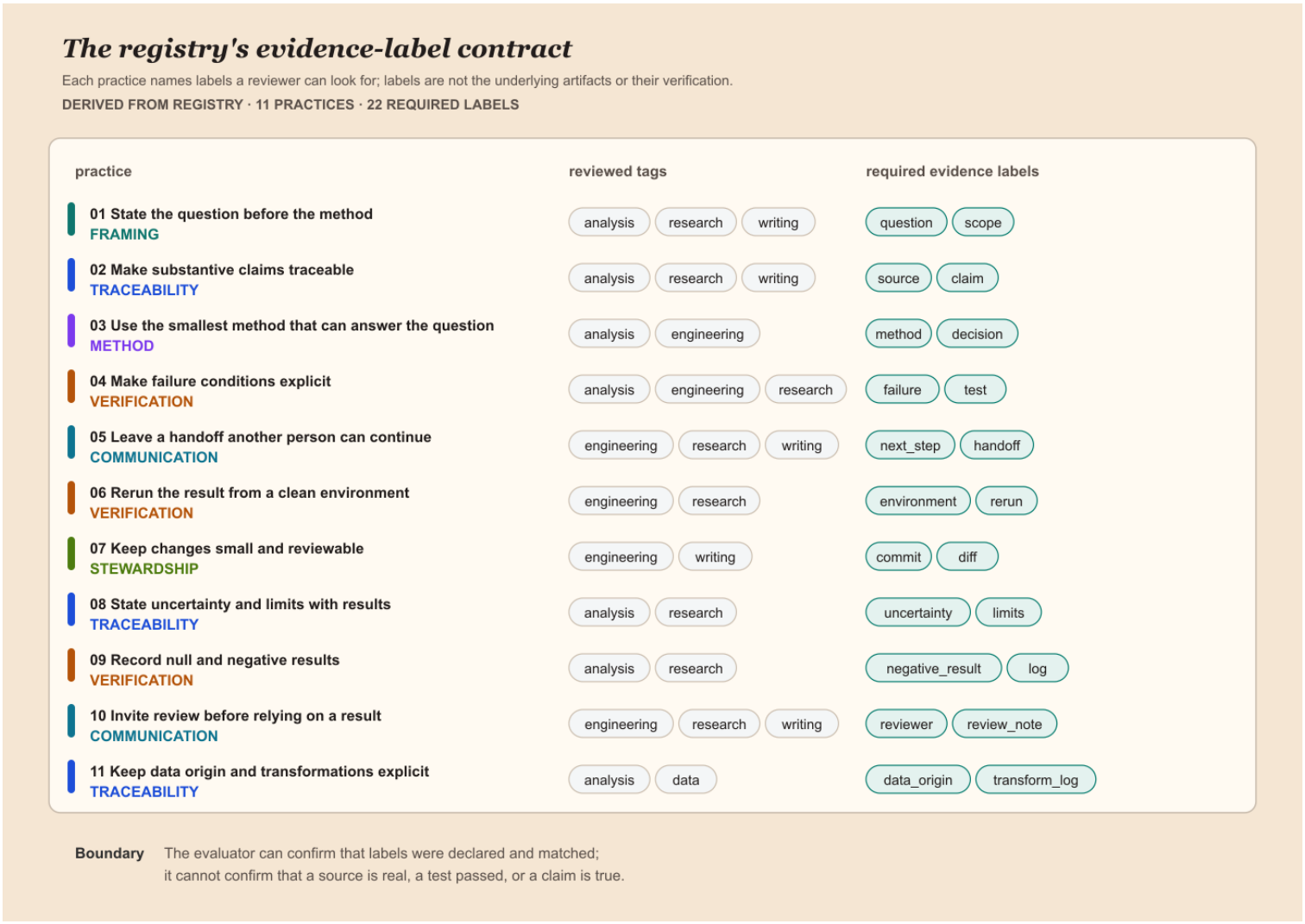


Figure 6: The registry's evidence-label contract: each practice names labels a reviewer can look for, while the instrument explicitly does not treat a declaration as independent verification of a source, test, or claim.

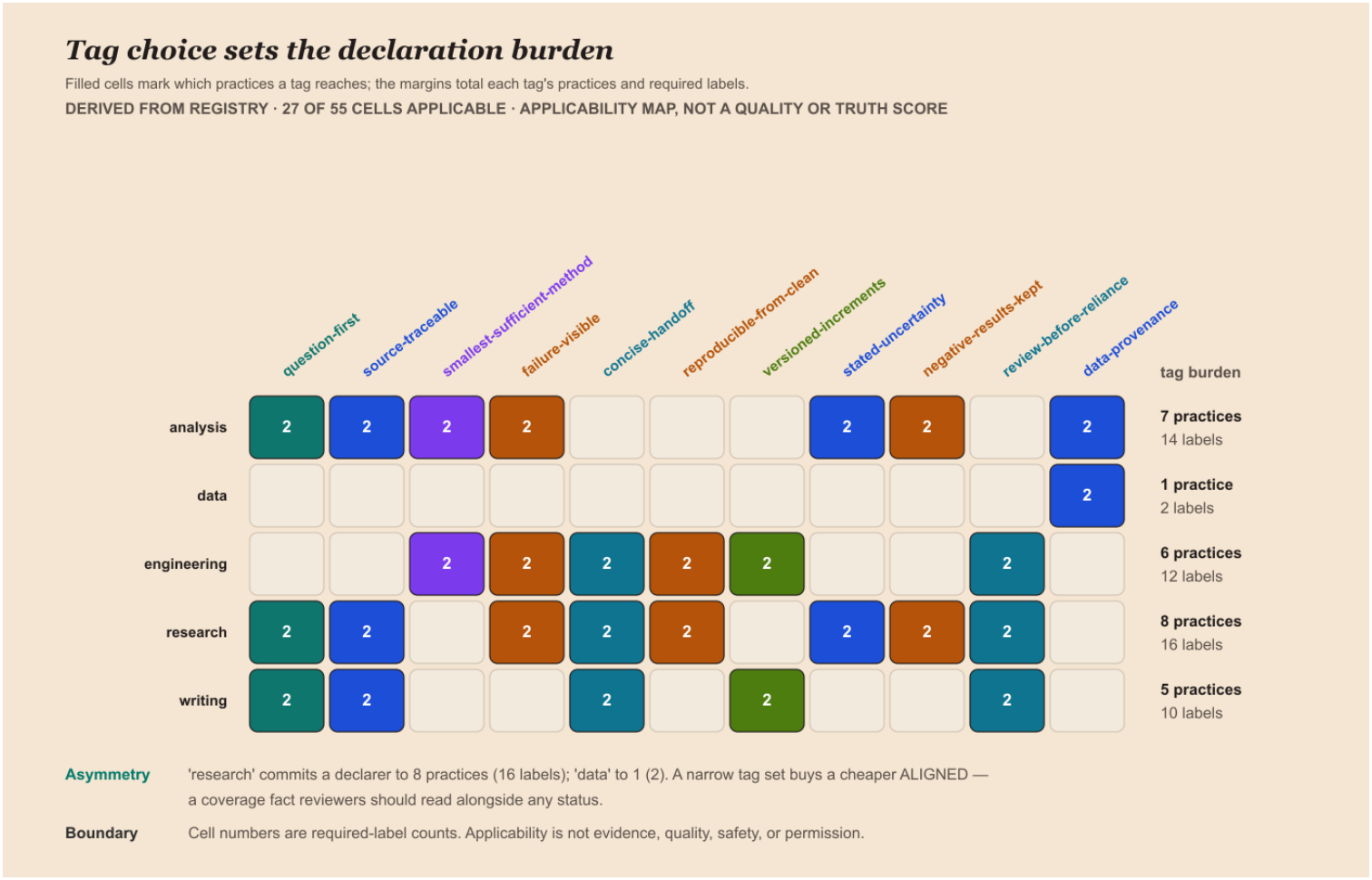


Figure 7: The tag-practice coverage matrix derived from the registry: filled cells mark applicability, cell numbers give each practice's required-label count, and the margin totals each tag's reach and declaration burden. The burden is asymmetric — `research` reaches 8 practices (16 required labels) while `data` reaches 1 (2 labels) — so a narrow tag set buys a cheaper `ALIGNED`. The matrix maps applicability, not evidence quality, safety, or permission.

8 Formal method: the evaluator and its invariants

The formalism below restates the evaluator that produces the coverage properties just shown.

This section states, as definitions and propositions, exactly what the implemented package computes. Every object, status name, and decision rule is taken from the source; the formalism describes the code and does not extend it. Each property is marked as guaranteed by an executable test or as following by inspection of the decision rule. Writing down what would falsify a claim, and then planting a counter-example to confirm the check can fail, is the falsificationist stance of the scholarship section applied to the tool itself [Popper, 1959].

No number below is written in the source. Every definition and proposition carries a label, the numbering is generated in document order, and every cross-reference resolves from the label, so an inserted block cannot leave a reference pointing at the wrong statement.

8.1 Domain objects

Definition 1 (Practice). A *practice* is a frozen record $p = (\text{id}, \text{title}, \text{wire}, \text{tags}, \text{req}, \text{kind})$ where id, title, wire are strings, tags is a finite set of tag strings, req is a finite ordered tuple of required evidence labels, and kind is a craft family. The field kind defaults to `METHOD` when omitted from positional construction.

Definition 2 (Tag vocabulary). The reviewed tag vocabulary is the fixed set

$$V = \{\texttt{analysis}, \texttt{data}, \texttt{engineering}, \texttt{research}, \texttt{writing}\},$$

with $|V| = 5$. Practice tags outside V are unreviewed drift, which is what I3 in Proposition 14 refuses.

Definition 3 (Registry). The *registry* $R = (p_1, \ldots, p_n)$ is the ordered tuple of eleven practices exported as `BLACK_PRACTICES`. Its content is summarized by the deterministic SHA-256 `registry_digest`, so any edit to a wire is visible as a digest change in review. An assessment records this digest so a serialized result can be compared with the registry that produced it.

Definition 4 (Craft families). The family of a practice is a member of

$$K = \{\texttt{FRAMING}, \texttt{TRACEABILITY}, \texttt{METHOD}, \texttt{VERIFICATION}, \texttt{COMMUNICATION}, \texttt{STEWARDSHIP}\},$$

with six families. Families exist so registry balance can be reviewed; I5 in Proposition 14 requires each family to retain at least one practice.

Definition 5 (Work attempt). A *work attempt* is a record $W = (\text{desc}, T, E, D)$ where desc is a description string, T is a declared tag set, E is a declared undated evidence-label set, and D is a tuple of *dated evidence items*. A dated evidence item is a pair (ℓ, τ) of a label ℓ and an optional ISO date τ (with $\tau = \bot$ meaning undated).

Definition 6 (Assessment record). An assessment is a record $A = (s, \mathcal{F}, N, d, h)$ containing an overall status s, ordered practice findings $\mathcal{F}$, intake notes N, review date d, and registry digest h. Each finding is itself a triple of a practice id, a practice status, and an ordered reasons trail. The digest identifies method content only; it is not a signature of the evidence or the work.

8.2 Status codomains

Definition 7 (Practice status). A per-practice outcome lies in

$$S_p = \{\texttt{ALIGNED}, \texttt{NEEDS\_EVIDENCE}, \texttt{NEEDS\_REWORK}\}.$$

Definition 8 (Assessment status). An overall outcome lies in

$$S_a = S_p \cup \{\texttt{OUTSIDE\_SCOPE}\} = \{\texttt{ALIGNED}, \texttt{NEEDS\_EVIDENCE}, \texttt{NEEDS\_REWORK}, \texttt{OUTSIDE\_SCOPE}\}.$$

`OUTSIDE_SCOPE` is available to the overall status only; it is never a per-practice status.

8.3 The staged evaluator

The function under study is

$$\text{evaluate\_work} : (W, R, d, \omega) \mapsto A,$$

where d is the review date (`as_of`, defaulting to today), ω is an optional staleness window in days (`max_evidence_age_days`, with $\omega = \bot$ disabling staleness), and A is a `BlackAssessment`. Two pre-stages run before any scoring: the review configuration is validated, and the supplied registry is checked for a shape that can be scored at all. Only then do the four numbered stages run.

Definition 9 (Review configuration). The review date accepts $\perp$ (meaning today), an ISO date string, or a `datetime.date`. A malformed ISO string raises `ValueError`; a `datetime`, or any other type, raises `TypeError` rather than being coerced. The window accepts $\perp$ or a non-negative `int`; a `bool`, a non-integer, or a negative value raises. This is the one input class the evaluator refuses outright, because a misread review date silently changes every age comparison downstream.

Definition 10 (Intake normalization). *Stage 1* normalizes hostile or malformed input into declarations plus review notes rather than raising. A declared label collection X is normalized by

$$\text{clean}(X) = \{\, \text{lower}(\text{strip}(t)) : t \in X,\ t \text{ a non-blank string} \,\},$$

and every dropped token, non-collection field, or string-valued field is recorded as an intake note. The *blocking predicate* is

$$\text{block}(W) \equiv \neg\, \text{isstr}(\text{desc}) \ \vee\ \text{strip}(\text{desc}) = \varepsilon.$$

Definition 11 (Freshness partition). *Stage 2* splits the dated evidence D, at review date d under window ω, into a fresh set and a stale set:

- a record whose label is missing or is not a non-blank string is dropped with a note;
- an undated item $(\ell, \perp)$ contributes ℓ to *fresh*;
- an item dated $\tau > d$ (future) or with an unparseable τ is counted in neither set and is surfaced as a note;
- an item with $\omega \neq \perp$ and age $(d - \tau) > \omega$ contributes ℓ to *stale*;
- otherwise the item contributes ℓ to *fresh*.

The evaluated evidence sets are the *fresh* set $F = \text{clean}(E) \cup \text{fresh\_dated}$ and the *stale* set Σ.

The staleness comparison is *strict*: an item aged exactly ω days satisfies $(d - \tau) = \omega \not> \omega$ and is therefore still fresh; staleness begins at age $\omega + 1$. Proposition 9 pins this boundary with an executed witness.

Definition 12 (Applicability). *Stage 3* selects the applicable practices by non-empty tag intersection:

$$A(W) = \{\, p \in R : p.\text{tags} \cap \text{clean}(T) \neq \emptyset \,\}.$$

Definition 13 (Practice finding rule). *Stage 4* scores each applicable practice p against (F, Σ). Let present and missing be the subsequences of p.req, in declaration order, whose labels are respectively in and not in F. The finding $\varphi(p, F, \Sigma)$ is given by the first matching rule:

- `NEEDS_EVIDENCE` if $F = \emptyset$ and $\Sigma = \emptyset$ (the attempt declared no usable evidence at all);
- otherwise `ALIGNED` if missing $= \emptyset$ (every required label is fresh);
- otherwise `NEEDS_EVIDENCE` if missing $\subseteq \Sigma$ (every remaining gap is merely a stale item awaiting refresh);
- otherwise `NEEDS_REWORK`.

Each finding also carries a reasons trail naming present, missing, and stale-refresh labels.

The first branch is a statement about the attempt, not about the practice; Proposition 6 gives that reading exactly, because it is the branch most often misread.

Definition 14 (Overall aggregation). *Stage 4, second half* aggregates the set of finding statuses $\text{stat}(\mathcal{F})$ into the assessment status by the first matching rule:

- `NEEDS_REWORK` if `NEEDS_REWORK` $\in \text{stat}(\mathcal{F})$;
- otherwise `NEEDS_EVIDENCE` if `NEEDS_EVIDENCE` $\in \text{stat}(\mathcal{F})$;
- otherwise `ALIGNED` if $\mathcal{F} \neq \emptyset$;
- otherwise `OUTSIDE_SCOPE`.

The decision path from intake through per-practice findings to this status is drawn below.

8.4 Surfaces, projection, and the report envelope

The scoring rule of Definition 13 reads its subsequences off an intermediate object worth naming, because a status word is the *last* step of Stage 4, not the whole state. Strong support and strong resistance co-present on one practice are not the same as no evidence, even when both correctly yield the same demanding word for action; the typed surfaces are what keep those situations distinguishable after the word is chosen.

Definition 15 (Evidence surfaces). For an applicable practice p evaluated against (F, Σ), the *evidence surfaces* are the frozen record $\sigma(p) = (\text{practice\_id}, \text{present}, \text{missing}, \text{stale})$ with exactly those four fields: present and missing partition p.req by membership

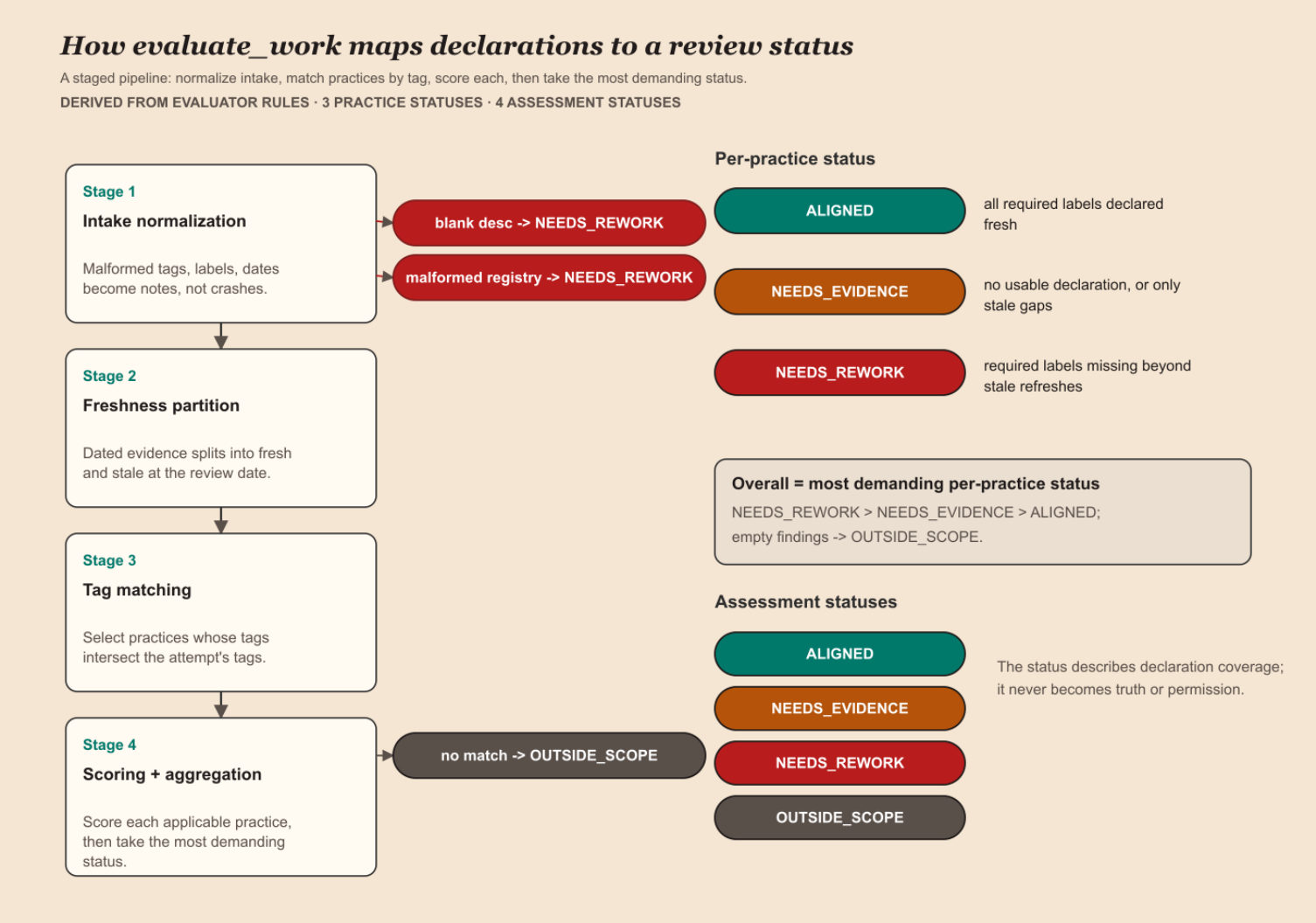


Figure 8: The staged evaluation path. Intake normalization can block on a blank description, and a malformed practice registry fails closed as `NEEDS_REWORK` before any scoring; otherwise practices are matched by tag intersection, each applicable practice is scored `ALIGNED`, `NEEDS_EVIDENCE`, or `NEEDS_REWORK`, and the overall status takes the most demanding per-practice status, or `OUTSIDE_SCOPE` when no practice applies. The diagram is derived from the evaluator rules and status enums; a status reports declaration coverage and freshness, not verified evidence or authorization.

in F, and stale is the subsequence of missing whose labels are in Σ, so stale $\subseteq$ missing always holds. All three tuples keep the practice's declared evidence order. The surfaces are co-present state — support and resistance on one practice are both kept, and neither erases the other — and, like every object in this section, they describe declaration coverage only.

The finding is then a *projection* of that state: precedence selects the most demanding reading for action, and the surfaces preserve what the selection compresses.

Proposition 1 (The finding is a projection of its surfaces). For every applicable practice, the finding's status and its ordered reasons trail are functions of $(\sigma(p), e)$ alone, where $e \equiv (F \cup \Sigma \neq \emptyset)$ is the attempt-wide evidence-declared bit whose reading Proposition 6 gives; every reasons trail is re-derivable from its surfaces. The two public forms report one staged computation: `evaluate_with_surfaces` returns the surfaces beside an assessment that is byte-identical, under canonical serialization, to what `evaluate_work` returns for the same arguments — one staged core, never a second evaluator. On a blocking intake defect or an unscorable registry there are no findings and therefore no surfaces. *Evidence:* `tests/test_witness_surfaces.py::test_both_public_forms_report_one_staged_computation` across a battery reaching every assessment status, `tests/test_witness_surfaces.py::test_surfaces_align_one_to_one_with_findings_and_generate_their_reasons`, and the definition re-derivation in `tests/test_formalism_definitions.py::test_surfaces_projection_matches_the_proposition`.

The panel below draws the projection once, from a single executed call, so the compression is visible rather than asserted: several distinguishable surface shapes share one projected word, and the overall status is one word for all of them.

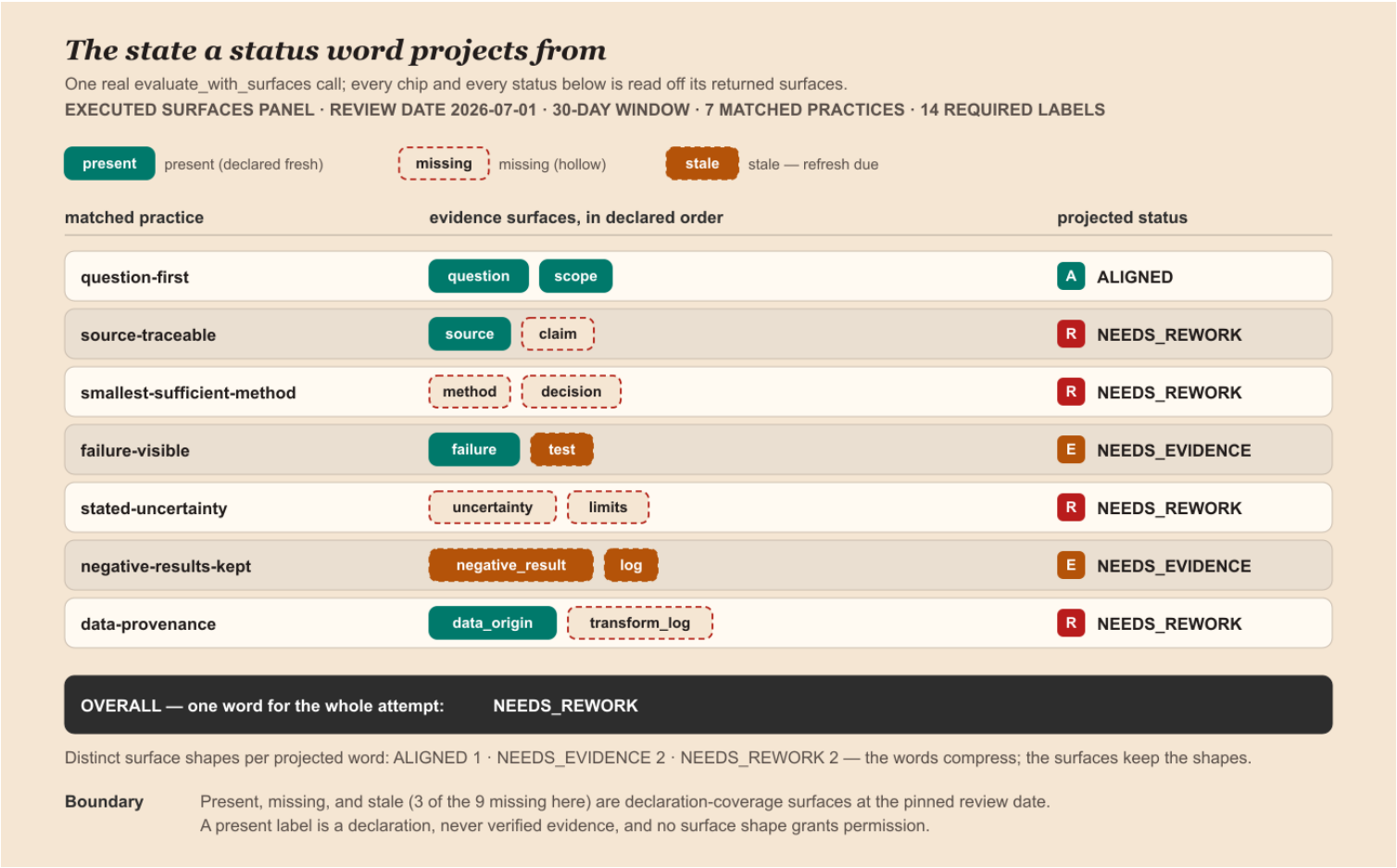


Figure 9: One declaration's typed evidence surfaces beside the statuses projected from them: a single `evaluate_with_surfaces` call at review date 2026-07-01 under a 30-day window matches 7 practices, whose 14 required labels split into 5 present (filled), 9 missing (hollow), and 3 of the missing merely stale (amber refresh marks). One status word covers distinguishable surface shapes: `NEEDS_EVIDENCE` is projected from 2 distinct present/missing/stale shapes and `NEEDS_REWORK` is projected from 2 distinct present/missing/stale shapes, while the single overall word for the whole attempt is `NEEDS_REWORK`. The surfaces keep what the projection compresses — strong support and strong resistance co-present on one practice are not the same as no evidence — and, like the statuses, they describe declaration coverage only: a present label is a declaration, never verified evidence, and no surface shape grants permission.

A reader holding reports from several independent instruments needs one uniform way to say "this instrument, about this subject, at this review moment, said this — and here is the pointer to its complete native report." The envelope is that data contract and

nothing more.

Definition 16 (The report envelope). The *report envelope* is the frozen record `AssessmentEnvelope` with exactly the ten fields, in order, `schema_version`, `line_id`, `subject_id`, `review_date`, `registry_version`, `registry_digest`, `native_status`, `report_ref`, `source_snapshot_refs`, and `scope_and_nonclaims`, declared under the cross-instrument schema string `line.report-envelope/1.0`. report_ref is the SHA-256 digest of the complete `canonical_assessment` serialization of Proposition 8, so the envelope points at the full native derivation — every finding and every intake note — without restating or reinterpreting any of it, and scope_and_nonclaims carries the instrument's non-claims inside the record itself, so a stored envelope cannot quietly outgrow what the instrument was allowed to say. native_status is this line's own status word in this line's own vocabulary; envelopes from different lines must not be compared, ranked, averaged, or merged on it. Sibling instruments that export the same shape do so by publishing the same schema string, never by importing one another. source_snapshot_refs is caller-supplied provenance that the envelope stores and does not verify.

An envelope is a witness record, not a score: it makes one instrument's complete report co-registrable beside the others' without granting any reader a licence to aggregate the status words.

8.5 Propositions

Proposition 2 (A malformed registry fails closed). Before any attempt is read, the supplied registry is checked for entries that are not practice records, for blank or non-text ids, titles, and wires, for duplicate ids, for malformed tag or evidence fields, and for invalid families; the canonical digest is then computed. If either step fails, the assessment is `NEEDS_REWORK` with no findings, an empty digest, and an intake note naming the defect. The direction is deliberate: a registry that cannot be scored must not produce a permissive status. *Evidence:* `tests/test_evaluator.py::test_custom_registry_failures_are_blocking` and `tests/test_evaluator.py::test_malformed_custom_registry_is_blocked_before_tag_matching`.

Proposition 3 (Intake records supported malformed declarations). For a work attempt whose T, E, and D fields are non-iterable, strings, or ordinary iterables yielding malformed tokens, Stage 1 returns normalized sets and notes rather than raising: non-collection or string-valued fields become a single note, malformed tokens are dropped with a note, and unreadable or future dates are excluded with a note. This is defensive normalization, not a sandbox for arbitrary user-defined iterators or properties. *Evidence:* the hostile-input cases in `tests/test_evaluator.py`, and the executed battery in `tests/test_figures.py::test_the_intake_plate_follows_the_executed_battery`.

The battery is drawn below, one row per malformed declaration, so the claim "notes rather than exceptions" can be read off outputs instead of taken on trust.

Two rows of that plate read directly against Definition 10. An evidence collection carrying a blank and a non-text token still reaches `ALIGNED`, because the two usable labels survive normalization while the two dropped tokens become notes. A tag set declared as the bare string `data` is dropped whole, leaving nothing to match, so the attempt is `OUTSIDE_SCOPE` — a coverage statement produced by a typo, which is why intake notes belong in the returned record rather than in a log line.

Proposition 4 (Blocking description short-circuits). If block(W) holds, then A.status = `NEEDS_REWORK`, A.findings = $\emptyset$, and A.notes contains the restatement instruction. No practice is scored, because no honest scoring is possible without a described work item. *Evidence:* by inspection of Stage 1's early return, exercised in `tests/test_evaluator.py::test_blank_description_is_a_blocking_intake_defect`.

Proposition 5 (Scope characterization). For a non-blocking attempt, A.status = `OUTSIDE_SCOPE` if and only if $A(W) = \emptyset$: no practice's tags intersect the declared tags. An `OUTSIDE_SCOPE` status is therefore a statement about *coverage*, never about safety or permission. *Evidence:* follows from Definition 12 and Definition 14, exercised in `tests/test_evaluator.py::test_no_matching_tags_is_outside_scope_with_review_date`.

Proposition 6 (Global versus local emptiness). The first branch of Definition 13 tests the *global* evidence sets F and Σ, not the practice's own requirements: "no evidence was declared" fires only when the attempt supplied no usable fresh or stale evidence at all. When some evidence exists but none of it satisfies p, control reaches the fourth branch — `NEEDS_REWORK`, all required labels missing and none stale — not the first. An attempt that has *begun* work but omitted a practice's evidence is therefore told to rework it, not that it declared nothing. *Evidence:* `tests/test_manuscript_bindings.py::test_method_prose_matches_the_global_emptiness_branch`.

Proposition 7 (Conditional fresh-evidence monotonicity). Fix p, Σ, and let $F \subseteq F'$ be two fresh sets. Once $F \cup \Sigma \neq \emptyset$, adding fresh labels cannot move a finding toward a more demanding status under the order `NEEDS_REWORK` $\prec$ `NEEDS_EVIDENCE` $\prec$ `ALIGNED`; adding required labels can only shrink missing. There is one intentional boundary case: when both F and Σ are empty, adding an irrelevant fresh label changes the global branch from `NEEDS_EVIDENCE` to `NEEDS_REWORK`, because the attempt has now declared something while still omitting the practice's requirements. *Evidence:* the conditional claim follows from Definition 13 and is exercised as a seeded sampled property by `tests/test_analytics.py::test_sampled_permutations_never_regress_after_first_declaration` and `tests/test_analytics.py::test_declaration_path_under_staleness_never_regresses_after`

Figure 10: Stage 1 executed over 9 deliberately malformed declarations at review date 2026-07-01: each is one real evaluate_work call, together they return 10 intake notes, and none raises. 2 block scoring outright (description is blank and description is not text) and 1 reaches no practice once its dropped tag declaration leaves nothing to match, while the other 6 are scored normally; across the battery the outcomes are ALIGNED, NEEDS_EVIDENCE, NEEDS_REWORK, OUTSIDE_SCOPE. Surviving malformed input is a robustness property of the intake stage, not tolerance of a bad declaration, and a note asks the declarer to fix something rather than verifying anything declared correctly.

_first_step; the boundary case is exercised by tests/test_evaluator.py::test_irrelevant_evidence_only_needs_rework_without_present_reason. The same sweep is drawn as the monotonicity lattice in the worked examples, where the per-row monotonicity mark and the inversion count are read off executed paths rather than asserted.

Proposition 8 (Determinism and archivability). For fixed ordinary data values (W, R, d, ω) with a materialized registry tuple, evaluate_work is a pure function, and canonical_assessment serializes the result — including the registry digest — to byte-identical JSON across runs, so an assessment can be diffed and cited in a review record. *Evidence:* tests/test_serialization.py::test_canonical_assessment_is_deterministic_and_complete.

Proposition 9 (Strict staleness boundary). Fix $\omega \neq \perp$ and a dated item (ℓ, τ) with $\tau \leq d$. The item is stale if and only if $(d - \tau) > \omega$; the equality case $(d - \tau) = \omega$ is fresh. Executed witness: a full data declaration aged exactly 51 days is fresh under $\omega = 51$ (overall status ALIGNED) and stale under $\omega = 50$ (overall status NEEDS_EVIDENCE, every gap being merely stale); with $\omega = \perp$ staleness is disabled and the status is again ALIGNED. The boundary is a fact about date arithmetic inside the evaluator, not a judgment that 51-day-old evidence is trustworthy. *Evidence:* tests/test_analytics.py::test_staleness_profile_pins_the_strict_inequality_boundary and tests/test_evaluator.py::test_fully_stale_evidence_needs_refresh_not_rework.

Proposition 10 (Window monotonicity, sampled). Fix an attempt whose dated evidence parses (no future or unreadable dates). Widening the freshness window never lowers the declaration-coverage rank of Proposition 13: enlarging ω can only move labels from Σ to F. For the aged-51 witness of Proposition 9, sweeping every integer window $\omega \in \{0, \ldots, 90\}$ and then $\omega = \perp$ yields NEEDS_EVIDENCE for every $\omega < 51$ and ALIGNED for every $\omega \geq 51$, including $\omega = \perp$, and the coverage rank is non-decreasing along the sweep. Beyond this fully-enumerated witness the claim is exercised as a seeded sampled property, not proven for all inputs; and a wider window is a more permissive review setting, not better work. *Evidence:* tests/test_analytics.py::test_widening_the_window_never_regresses_coverage_rank and the sweep re-derivation in tests/test_manuscript_bindings.py::test_formalism_window_sweep_witness_re_derives.

Proposition 11 (Coverage-matrix algebra). For the shipped registry R ($n = 11$) with tag vocabulary V ($|V| = 5$), coverage_matrix returns one row per declared tag, sorted alphabetically, each row carrying the tag's *reach* (practices selected) and *burden* (total required labels a declarer of only that tag is scored against). Two identities hold by execution: (i) the rows fill $\sum_t \text{reach}(t) = \sum_{p \in R} |p.\text{tags}|$, which is 27 of the 55 tag-practice cells; (ii) since every shipped practice requires exactly two labels, $\text{burden}(t) = 2 \cdot \text{reach}(t)$ for every tag. The executed rows, as (tag, reach, burden), are (analysis, 7, 14), (data, 1, 2), (engineering, 6, 12), (research, 8, 16), and (writing, 5, 10). The matrix describes declared applicability only — a heavily reached tag is a costlier declaration, not a safer or better-reviewed domain. *Evidence:* tests/test_analytics.py::test_coverage_matrix_pins_the_registry_reach_and_burden and tests/test_analytics.py::test_coverage_matrix_total_cells_match_tag_declarations.

Proposition 12 (Digest order-independence and drift visibility). canonical_registry sorts practices by id before serializing, so registry_digest is invariant under any permutation of the registry tuple. The digest is a SHA-256 value rendered as 64 lowercase hexadecimal characters, and editing any single practice field changes it, which is what makes silent method drift visible in review. The universal clause is checked field by field: the test table is closed against dataclasses.fields(BlackPractice), so a field added to the record but omitted from canonical() fails rather than quietly making the proposition false. The digest identifies method content only; it is not a signature of evidence, of work, or of any person. *Evidence:* tests/test_serialization.py::test_digest_is_order_independent_and_hex_shaped, tests/test_serialization.py::test_digest_changes_when_any_practice_field_changes, and tests/test_serialization.py::test_field_edit_table_covers_every_serialized_field.

Proposition 13 (The coverage ladder is partial). The exported DECLARATION_STATUS_ORDER fixes NEEDS_REWORK $\prec$ NEEDS_EVIDENCE $\prec$ ALIGNED with ranks 0, 1, and 2 under status_rank. OUTSIDE_SCOPE has no rank: status_rank raises ValueError rather than comparing it, because an outside-scope assessment says no practice applied — a statement about tag coverage, not a position below or above any coverage status. *Evidence:* tests/test_analytics.py::test_declaration_status_order_ranks_rework_lowest_and_aligned_highest and tests/test_analytics.py::test_status_rank_rejects_outside_scope.

8.6 Structural invariants

The invariants battery checks the *shape* of the registry rather than any single attempt. Each check returns a RegistryCheck, and each is validated by a *proof-of-detection* pair: a test asserting it passes on the real registry and at least one test planting a counter-example that makes it fail. A green check that never saw a bad input does not count.

Proposition 14 (Invariants with proof of detection). The battery all_invariants runs exactly the following seven checks, in order, and the real registry passes all seven:

- **I1 — ids distinct.** Every practice id is a distinct, non-blank string. Planted: a duplicated, blank, and non-string id.

- **I2 — fields populated.** Title and wire are non-blank, and required evidence is a non-empty tuple of non-blank strings. Planted: a blank title, an empty evidence tuple, a blank label, and a string evidence field.
- **I3 — tags reachable.** Every practice has at least one tag, and every tag is in V. Planted: a zero-tag practice, a string tag field, and an out-of-vocabulary tag.
- **I4 — kind valid.** Every kind is a real family member. Planted: a string kind.
- **I5 — family coverage.** Every family in K retains at least one practice. Planted: a registry with all `STEWARDSHIP` practices removed.
- **I6 — labels matchable.** Required labels are distinct per practice and already lowercase-normalized (the evaluator lowercases declared evidence, so an uppercase registry label could never match), and the evidence field keeps its declared tuple shape. Planted: a duplicate label, an uppercase label, and a non-tuple field.
- **I7 — digest computable.** Canonical serialization and digesting succeed; a registry that cannot be digested cannot be reviewed for drift. Planted: a non-serializable tag field.

Evidence: `tests/test_invariants.py` contains the pass-on-real and fail-on-planted tests for each check, and asserts the battery has exactly seven uniquely named checks that are all green with detail `ok` on the real registry. `invariants_hold` is the conjunction of the seven and is `True` on R.

The plate below runs that battery eight times — once on the shipped registry and once on each planted registry — so proof of detection is visible as a matrix rather than asserted as a policy.

The off-diagonal cells are the honest part. A kind that is not a `PracticeKind` member drops its practice out of family coverage *and* breaks canonical serialization, so that one plant fails three checks; a `None` tag field is both unreachable and unserializable. Plants chosen to trip exactly one check each would have produced a cleaner diagonal and a less accurate figure.

8.7 Claim-to-test binding

Each definition and proposition above names the executable test that verifies it. The two tables below collect those bindings so a reader can go from claim to failing condition without searching. A claim whose test cannot fail is not admitted — the invariants battery makes the same demand of itself through planted counter-examples. Every row is a statement about code behaviour under declared inputs; none is a claim about the world, about safety, or about persons.

Definition	What the code must still do	Verifying test
Definition 1	Field names, order, and the `METHOD` default	`tests/test_formalism_definitions.py::test_practice_record_matches_the_definition`
Definition 2	V is exactly the five reviewed tags	`tests/test_formalism_definitions.py::test_tag_vocabulary_matches_the_definition`
Definition 3	R is an ordered tuple of eleven practices with a digest	`tests/test_formalism_definitions.py::test_registry_matches_the_definition`
Definition 4	K is exactly the six declared families	`tests/test_formalism_definitions.py::test_craft_families_match_the_definition`
Definition 5	W and the dated-item pair keep their shape	`tests/test_formalism_definitions.py::test_work_attempt_matches_the_definition`
Definition 6	A and its finding triples keep their shape	`tests/test_formalism_definitions.py::test_assessment_record_matches_the_definition`
Definition 7	S_p has exactly three members	`tests/test_formalism_definitions.py::test_practice_status_codomain_matches_the_definition`
Definition 8	$S_a = S_p \cup \{\texttt{OUTSIDE\_SCOPE}\}$	`tests/test_formalism_definitions.py::test_assessment_status_codomain_matches_the_definition`
Definition 9	Each named bad configuration raises rather than coerces	`tests/test_formalism_definitions.py::test_review_configuration_refuses_what_the_definition_names`

Definition	What the code must still do	Verifying test
Definition 10	`clean` strips, lowercases, and notes every drop	`tests/test_formalism_definitions.py::test_intake_normalization_matches_the_definition`
Definition 11	Each of the five partition rules, executed	`tests/test_formalism_definitions.py::test_freshness_partition_matches_the_definition`
Definition 12	Selection is exactly non-empty tag intersection	`tests/test_formalism_definitions.py::test_applicability_matches_the_definition`
Definition 13	All four branches, in order, with ordered trails	`tests/test_formalism_definitions.py::test_practice_finding_rule_matches_the_definition`
Definition 14	All four aggregation branches, in order	`tests/test_formalism_definitions.py::test_overall_aggregation_matches_the_definition`
Definition 15	Field names, order, stale $\subseteq$ missing, declared evidence order	`tests/test_formalism_definitions.py::test_evidence_surfaces_match_the_definition`
Definition 16	The ten fields, the digest pointer, the travelling non-claims	`tests/test_formalism_definitions.py::test_report_envelope_matches_the_definition`

Proposition	Statement essence	Verifying test	Boundary kept
Proposition 1	Status and reasons are projections of the typed surfaces; both public forms report one staged computation	`tests/test_witness_surfaces.py::test_both_public_forms_report_one_staged_computation`, `tests/test_witness_surfaces.py::test_surfaces_align_one_to_one_with_findings_and_generate_their_reasons`	surfaces of declarations, never evidence quality or a cross-line rank
Proposition 2	An unscoreable registry blocks, with a note	`tests/test_evaluator.py::test_custom_registry_failures_are_blocking`	shape of the registry, not merit of its practices
Proposition 3	Malformed intake is noted, never raised	`tests/test_evaluator.py::test_malformed_label_tokens_are_dropped_but_good_ones_kept`, `tests/test_evaluator.py::test_string_tag_declaration_is_ignored_with_a_note`	supported hostile branches only, not a sandbox
Proposition 4	Blank description blocks all scoring	`tests/test_evaluator.py::test_blank_description_is_a_blocking_intake_defect`	no honest scoring without a described work item
Proposition 5	`OUTSIDE_SCOPE` $\iff$ no tag intersects	`tests/test_evaluator.py::test_no_matching_tags_is_outside_scope_with_review_date`	coverage statement, never safety or permission
Proposition 6	The first branch is global, not per-practice	`tests/test_manuscript_bindings.py::test_method_prose_matches_the_global_emptiness_branch`	rule about the code path, not about the work

Proposition	Statement essence	Verifying test	Boundary kept
Proposition 7	Fresh labels never demote a finding (conditional)	`tests/test_analytics.py::test_sampled_permutations_never_regress_after_first_declaration`, `tests/test_evaluator.py::test_irrelevant_evidence_only_needs_rework_without_present_reason`	sampled property; empty-declaration boundary is intentional
Proposition 8	Evaluation and serialization are deterministic	`tests/test_serialization.py::test_canonical_assessment_is_deterministic_and_complete`	byte-identity of records, not correctness of work
Proposition 9	Staleness is strict: age $> \omega$, equality fresh	`tests/test_analytics.py::test_staleness_profile_pins_the_strict_inequality_boundary`	date arithmetic, not trustworthiness of old evidence
Proposition 10	Widening ω never lowers coverage rank	`tests/test_analytics.py::test_widening_the_window_never_regresses_coverage_rank`	permissiveness of review setting, not work quality
Proposition 11	Coverage rows: 27 of 55 cells; burden $= 2\cdot$ reach	`tests/test_analytics.py::test_coverage_matrix_pins_the_registry_reach_and_burden`	declared applicability, not domain safety
Proposition 12	Digest is permutation-invariant and drift-visible in every field	`tests/test_serialization.py::test_digest_is_order_independent_and_hex_shaped`, `tests/test_serialization.py::test_digest_changes_when_any_practice_field_changes`	identifies method content, signs nothing else
Proposition 13	Ladder ranks $0 \prec 1 \prec 2$; `OUTSIDE_SCOPE` unranked	`tests/test_analytics.py::test_status_rank_rejects_outside_scope`	ordering of statuses, not of people or work
Proposition 14	Registry shape checks, each with planted failures	`tests/test_invariants.py::test_battery_runs_every_check_once_and_passes_on_real_registry`	shape of the registry, not merit of its practices

The named tests are themselves checked: a suite test re-reads this section and fails if any referenced `tests/…::function` does not exist, so a renamed or deleted binding surfaces as a red test rather than silent prose drift. A second test refuses any formalism block without a label, any reference to a label no block declares, and any hand-written block number anywhere in the manuscript.

These claims bound what the instrument establishes under its declared inputs and implementation. The tests establish that the registry is well-shaped and that scoring is deterministic, staged, and conditionally monotone in fresh evidence. They do not establish that a declared source is real, that a `test` label corresponds to a passing test, or that an `ALIGNED` status means the work is correct — only that the declared Black labels are present.

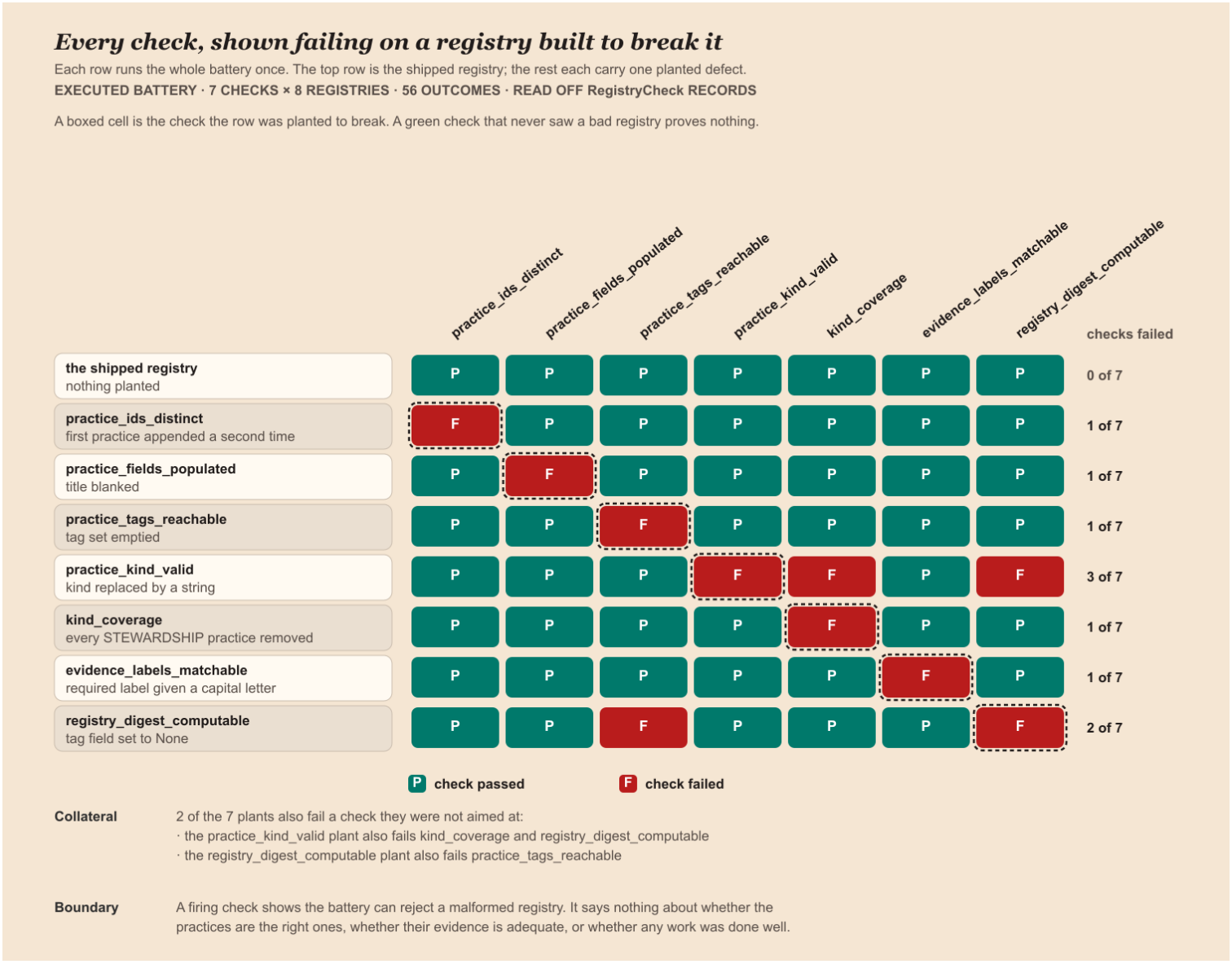


Figure 11: The 7-check structural battery run over 8 registries: the shipped registry, which passes every check, and 7 registries each carrying one planted defect. Every plant fails the check it targets, boxed in its row, which is what makes the battery a proof of detection rather than a record of greenness. 2 plants also fail a check they were not aimed at — the practice_kind_valid plant also fails kind_coverage and registry_digest_computable; the registry_digest_computable plant also fails practice_tags_reachable — because a single planted value can break more than one structural property at once; those cells are drawn rather than designed away. A firing check shows the battery can reject a malformed registry, not that the practices are the right ones, that their evidence is adequate, or that any work was done well.

9 Executed examples and boundaries

Every status in this section is the output of a real `evaluate_work` call — the same public API a reviewer would run — with the exact inputs stated so the transitions can be reproduced. None of the runs changes what a status means: each is a report on declaration coverage under the registry's tag contract, never a judgment of work quality, and never a permission.

9.1 An executed incremental-declaration path

The first worked example traces one attempt from an empty declaration to full coverage. The attempt is tagged `research`, which selects 8 of the 11 practices and commits the declarer to 16 required evidence labels (see the coverage matrix). `declaration_status_path` evaluates the same attempt 17 times — once with no evidence, then once after each label is added — through the ordinary evaluator. The table reports every step at which the declaration completes a practice, plus the two boundary steps:

Step	Labels added since previous row	Labels declared	Practice completed	Practices `ALIGNED`	Overall status
0	—	0	—	0 of 8	`NEEDS_EVIDENCE`
1	`question`	1	—	0 of 8	`NEEDS_REWORK`
2	`scope`	2	`question-first`	1 of 8	`NEEDS_REWORK`
4	`source`, `claim`	4	`source-traceable`	2 of 8	`NEEDS_REWORK`
6	`failure`, `test`	6	`failure-visible`	3 of 8	`NEEDS_REWORK`
8	`uncertainty`, `limits`	8	`stated-uncertainty`	4 of 8	`NEEDS_REWORK`
10	`negative_result`, `log`	10	`negative-results-kept`	5 of 8	`NEEDS_REWORK`
12	`environment`, `rerun`	12	`reproducible-from-clean`	6 of 8	`NEEDS_REWORK`
14	`next_step`, `handoff`	14	`concise-handoff`	7 of 8	`NEEDS_REWORK`
16	`reviewer`, `review_note`	16	`review-before-reliance`	8 of 8	`ALIGNED`

Three properties of the path are worth reading directly off the table. First, the overall status is the most demanding per-practice status, so it stays `NEEDS_REWORK` from step 1 through step 15 even as completed practices accumulate from 0 to 7; only the sixteenth label — completing the last applicable practice — yields `ALIGNED`. The per-practice findings, not the overall status, are where intermediate progress is visible. Second, the step 0 to step 1 transition is the boundary Proposition 6 names: an empty declaration is `NEEDS_EVIDENCE`, and the first label, although it adds information, moves the overall status to `NEEDS_REWORK`. Third, executed on this path, `no_status_regression` returns `True` for steps 1 through 16 and `False` only when the empty step 0 is included, which is exactly the conditional monotonicity of Proposition 7: once a declaration is non-empty, adding fresh labels never regresses the status. As Proposition 1 establishes, an `ALIGNED` reports declared labels — never that any source is real, any test passed, or any claim is correct.

The same path is drawn below as an executed status grid, one column per `evaluate_work` call, so the per-practice accumulation the table can only summarize is visible cell by cell:

One executed order is a trace, not a property. `declaration_status_path` is cheap enough to run over many orders, so the lattice below sweeps twelve seeded orders of the same sixteen labels — the registry's own order as row 0, then eleven permutations of it — and reports `no_status_regression` per row together with a count of every rank decrease in the sweep:

The sweep is what makes the claim falsifiable rather than illustrative: a single non-monotone row, or a rank decrease anywhere but the first step, would appear as a `NO` in the right-hand column and a larger count in the footer. It also shows something the single trace could not — the overall path is the same whichever order the labels arrive in, because the aggregation takes the most demanding per-practice status and the last applicable practice completes at step 16 in every order. That is an aggregation property, not a claim that declaration order is unimportant to the person doing the work.

That first transition is intentional, and Proposition 6 says why: an attempt with no usable evidence returns `NEEDS_EVIDENCE` because there is nothing to inspect, while one irrelevant label moves it to `NEEDS_REWORK` because the attempt has begun declaring and still omits every required label. The change reports a more specific request, not worse work, which is why the monotonicity

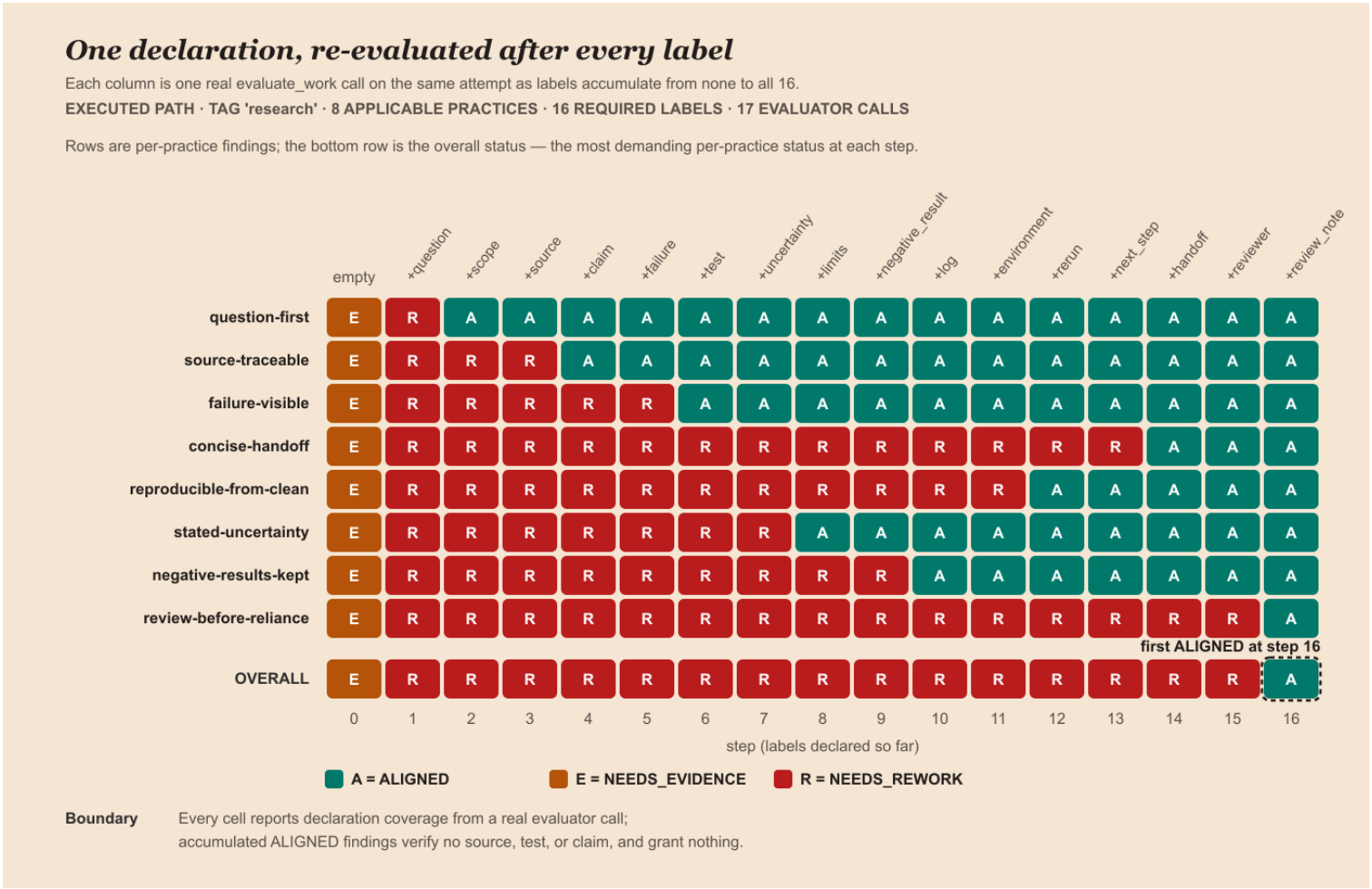


Figure 12: The executed incremental-declaration path as a status grid: the `research`-tagged attempt selects 8 practices (16 required labels), and every cell is one real `evaluate_work` call as labels accumulate one per step from an empty declaration (step 0, `NEEDS_EVIDENCE`) to full coverage (step 16). Per-practice findings flip to `ALIGNED` as each practice's labels complete, while the overall status — the most demanding per-practice status — stays `NEEDS_REWORK` from step 1 through step 15 and first reaches `ALIGNED` at step 16: conditional fresh-evidence monotonicity made visible. An `ALIGNED` cell records declared labels only; it does not show any source is real, any test passed, or any claim is true.

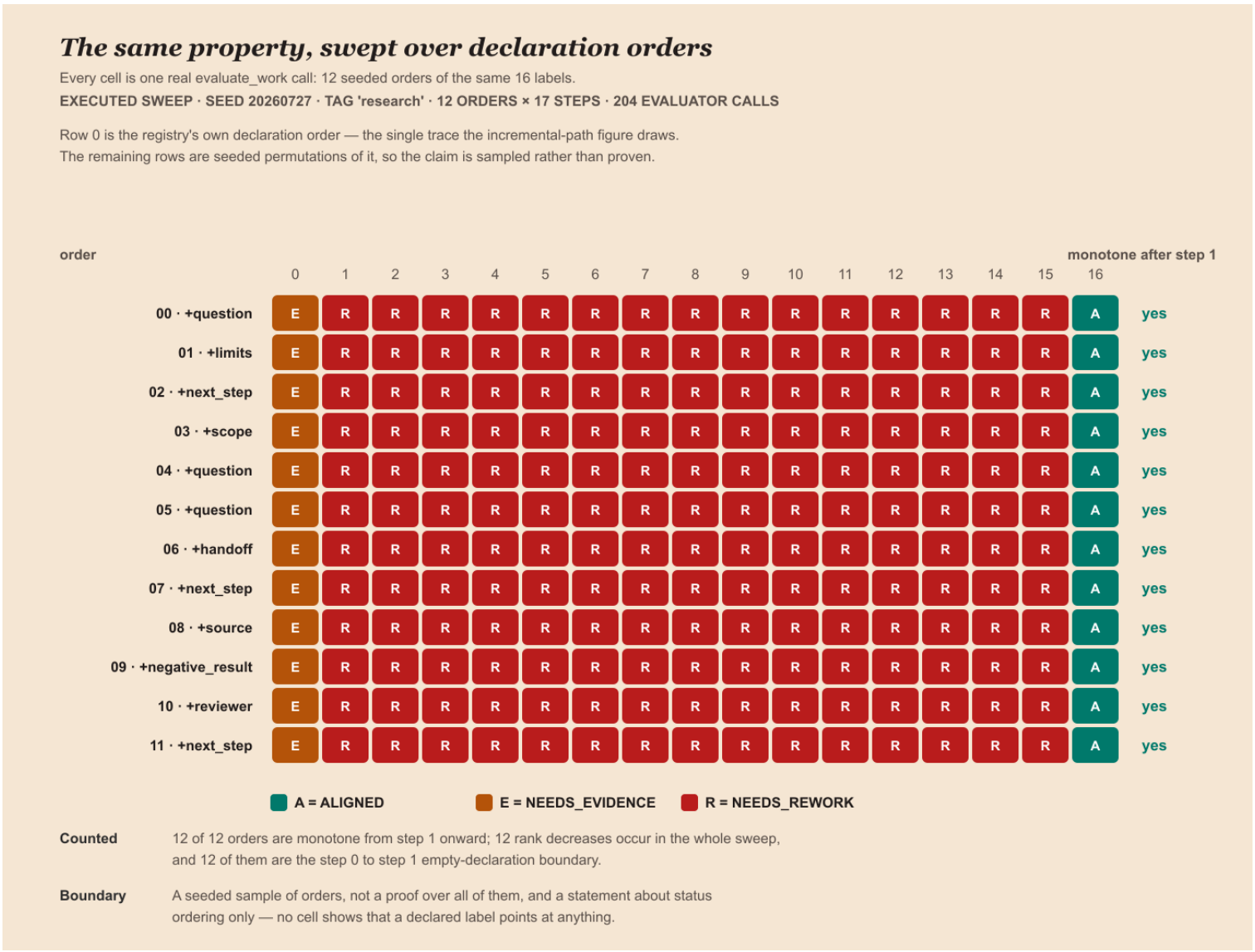


Figure 13: Conditional fresh-evidence monotonicity executed as a sweep rather than a single trace: 12 seeded orders (seed 20260727) of the same 16 labels, each evaluated at all 17 steps through the real evaluator, for 204 calls. Every row is monotone from step 1 onward (12 of 12), and all 12 rank decreases in the sweep are the step 0 to step 1 empty-declaration boundary. The overall status path is identical across every sampled order, which is a property of taking the most demanding per-practice status, not evidence that order is irrelevant to a reviewer. The sample is seeded and finite; it is not a proof over all 16! orders.

claim is scoped to the non-empty branch. The ALIGNED at step 16 carries only its declared labels and the registry digest that pins the method version behind them.

9.2 The decay sweep, executed

The staleness window shows how the instrument distinguishes decay from absence. Suppose a practice's evidence was fully declared but one supporting observation carries a date older than the configured window. Under staleness the aged observation stops counting as fresh, and because the practice's only gap is that one stale item, the finding is NEEDS_EVIDENCE — a request to refresh — rather than NEEDS_REWORK. The same evidence with no window, or with a recent date, returns ALIGNED. A date in the future, or a date the parser cannot read, is never counted and is reported as an intake note so the declarer can correct it.

The threshold is a strict inequality — evidence aged exactly the window is still fresh; one day older is stale. The sweep below executes staleness_profile over the data-provenance practice (required labels data_origin and transform_log) with review date 2026-07-01, re-reviewing the same declaration as its evidence ages. Each cell is one real evaluate_work call:

Evidence age (days)	Window 30	Window 51	No window	One label never declared, window 30
0	ALIGNED	ALIGNED	ALIGNED	NEEDS_REWORK
30	ALIGNED	ALIGNED	ALIGNED	NEEDS_REWORK
31	NEEDS_EVIDENCE	ALIGNED	ALIGNED	NEEDS_REWORK
51	NEEDS_EVIDENCE	ALIGNED	ALIGNED	NEEDS_REWORK
52	NEEDS_EVIDENCE	NEEDS_EVIDENCE	ALIGNED	NEEDS_REWORK
70	NEEDS_EVIDENCE	NEEDS_EVIDENCE	ALIGNED	NEEDS_REWORK

The three contrasts fix the semantics. A full declaration flips from ALIGNED to NEEDS_EVIDENCE exactly one day past its window — at age 31 under a 30-day window, at age 52 under a 51-day window. With no window, dated evidence never goes stale. And a declaration that never included one required label is NEEDS_REWORK at every age, because an absent label is missing work rather than aged work. The figure below traces the same boundary continuously over ages 0–70.

9.3 The refresh queue, executed

Decay describes one label at a time. A reviewer holding a whole declaration has a different question: which part of it expires first? refresh_horizon answers that by ordering the dated labels from nearest to furthest from the freshness boundary. The figure below runs it on an attempt tagged data and research at review date 2026-07-01 under a 120-day window.

Three declarations are absent from the queue, and the band names each one rather than dropping it. An undated label is treated as current, so it has no boundary to reach. An already-stale label is a refresh request now, not a schedule. And a label no practice in the registry requires — dashboard_link here — cannot move any status, so scheduling it would be misleading. The third case is why refresh_horizon takes the practice registry as an argument: the queue is a view of the registry's demands on a declaration, not an inventory of the declaration's dates.

9.4 A batch, executed

Every example so far follows one attempt. A reviewer holding a quarter's work holds many, and the question changes again: across differently-tagged attempts, which wires stay open? summarize_assessments answers the first half by counting statuses and attributing every non-ALIGNED finding to its craft family; recounting the same findings per practice answers the second. The panel below runs both over a pinned battery of eight attempts — one for each reviewed tag, one dated declaration aged past the window, one attempt declaring two tags at once, and one tagged outside the vocabulary — at review date 2026-07-01 under a 30-day window.

The batch shows something no single attempt can. data-provenance and versioned-increments are open most often not because they are harder practices but because of which tags reach them — analysis and data for the first, engineering and writing for the second — and the attempts carrying those tags here rarely declared the matching labels. question-first is the only practice the batch never leaves open, which says that the attempts declaring analysis, research, or writing all named a question and a scope, and nothing more. Read as a review artifact, the panel is a prompt: it names where declarations are thin across a body of work, and it stops there. It does not say those wires were done badly, or that the two ALIGNED attempts were done well.

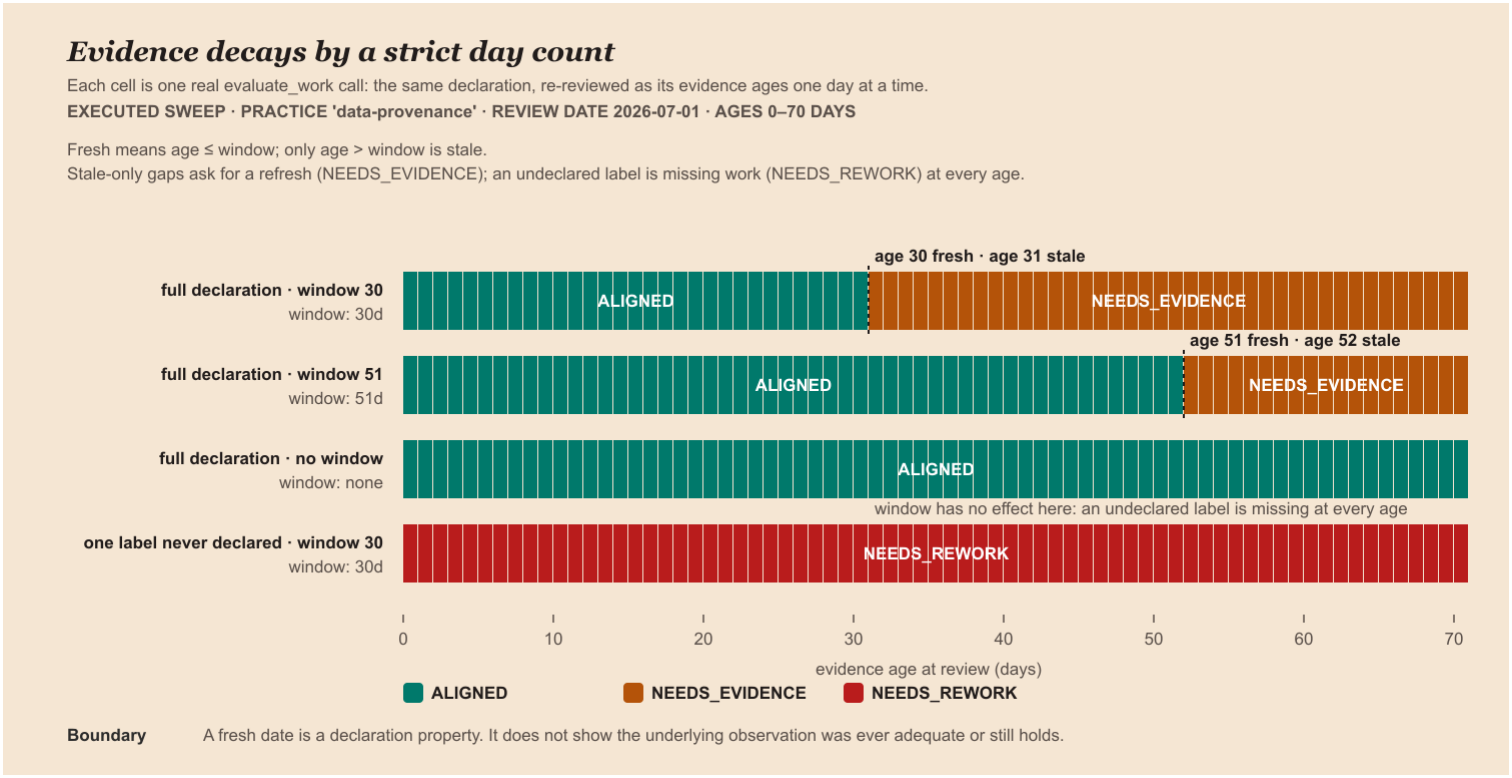


Figure 14: Evidence decay from executed `evaluate_work` sweeps over the `data-provenance` practice with review date 2026-07-01: a full declaration stays `ALIGNED` while its evidence age is at most the freshness window and flips to `NEEDS_EVIDENCE` (a refresh request) exactly one day past it — the threshold is a strict inequality — while a declaration that never included one required label is `NEEDS_REWORK` at every age, and a declaration with no window never goes stale. A fresh date is a declaration property; it does not show the underlying observation was ever adequate or still holds.

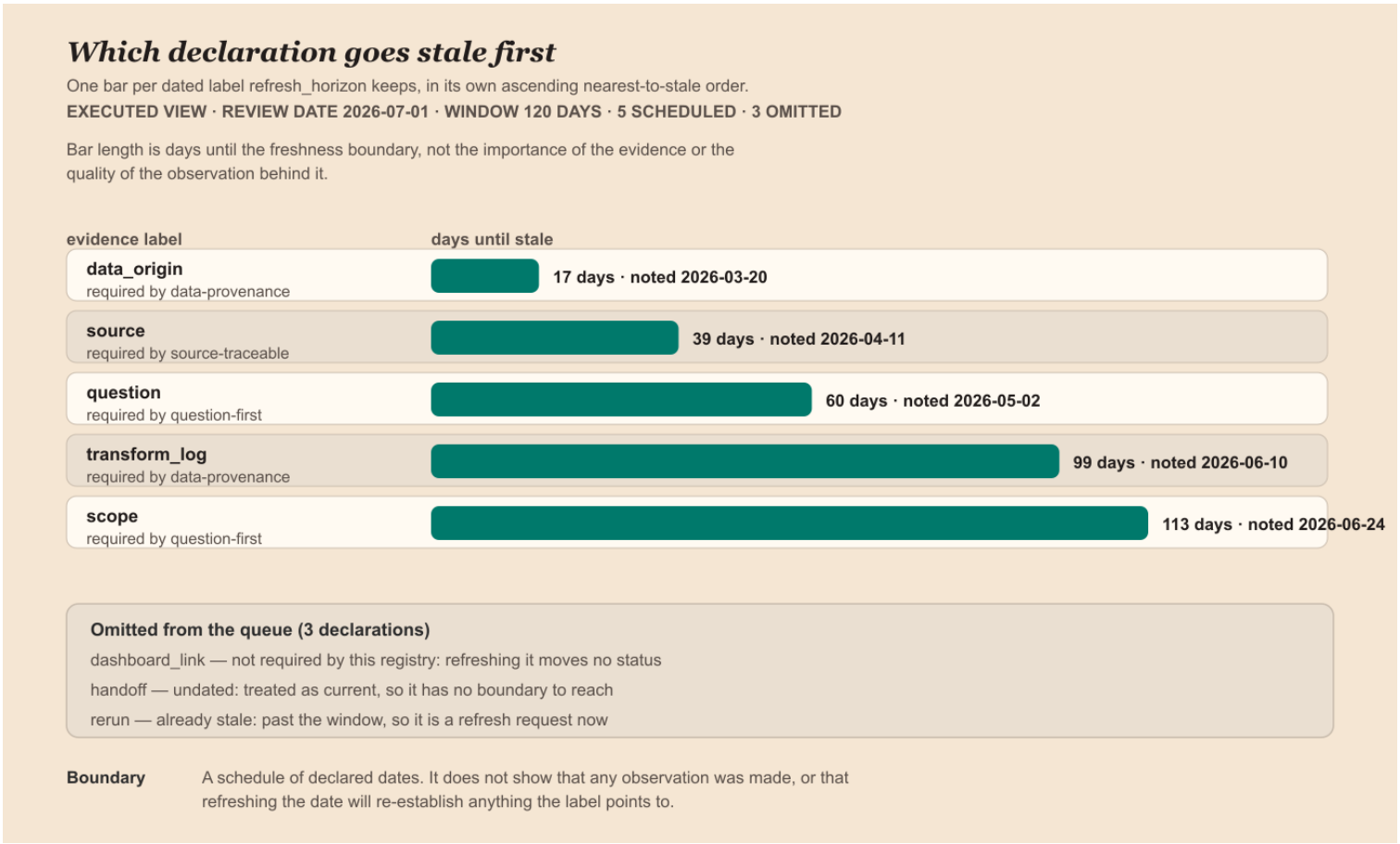


Figure 15: The refresh queue for one declaration, drawn from `refresh_horizon` in its own ascending nearest-to-stale order: at review date 2026-07-01 under a 120-day window, 5 dated labels are scheduled, from ‘data_origin’ at 17 days to ‘scope’ at 113, each naming the practice that requires it. The named band below records the 3 declarations the queue omits and why — one of them undated, which is itself one of the omitted classes — so the omission rule is visible rather than implicit. Bar length is days until a declared date crosses the window; it is not a measure of how much the underlying observation matters or how good it was.

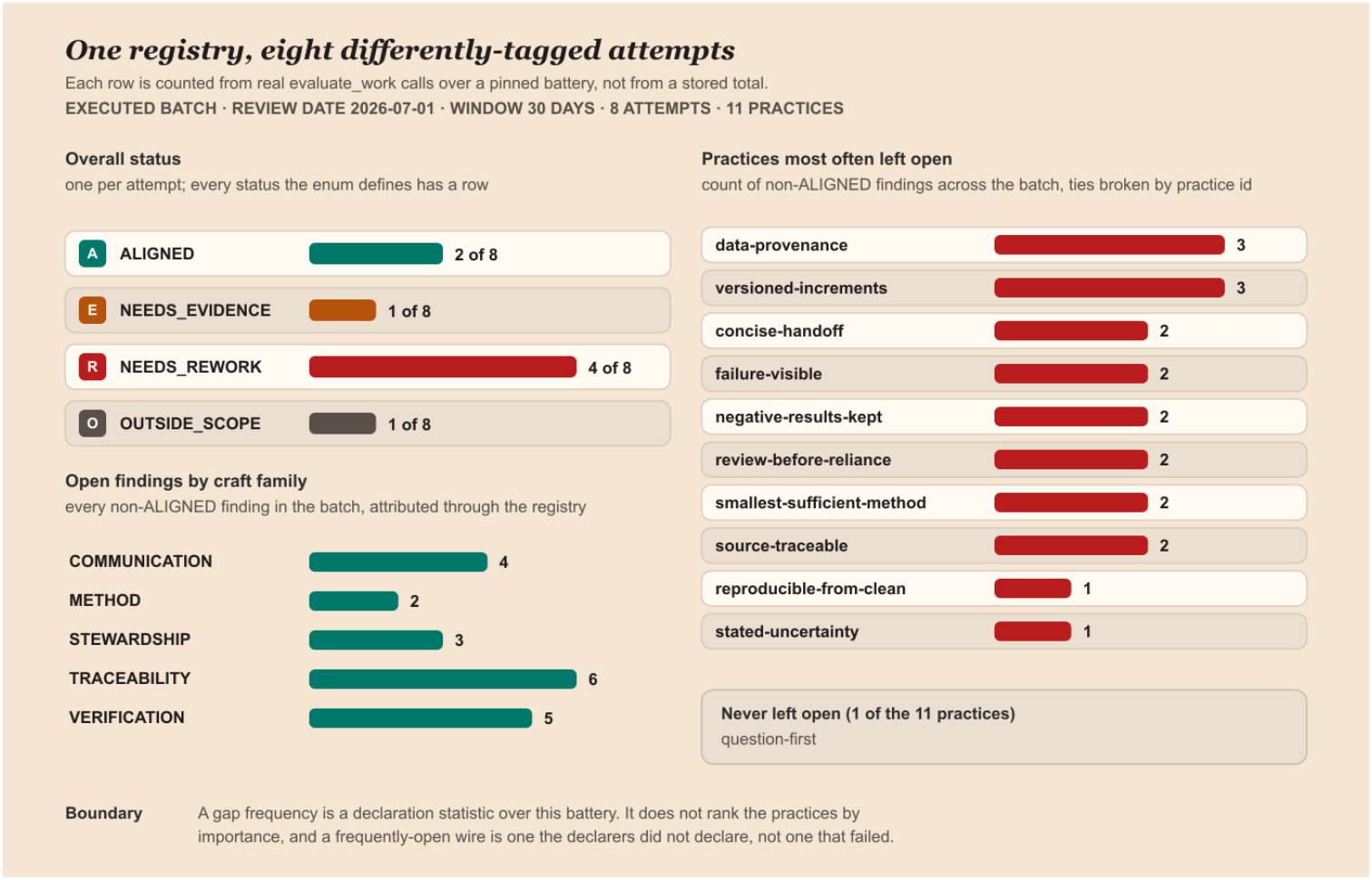


Figure 16: The registry's demands across a batch rather than one attempt: 8 pinned work attempts spanning the 5-tag vocabulary and one tag outside it, each evaluated at review date 2026-07-01 under a 30-day window. Every assessment status the enum defines occurs in the batch (`ALIGNED` 2, `NEEDS_EVIDENCE` 1, `NEEDS_REWORK` 4, `OUTSIDE_SCOPE` 1), open findings concentrate in TRACEABILITY (6), and 10 of the 11 practices are left open at least once, led by `data-provenance` and `versioned-increments` at 3. A gap frequency is a declaration statistic over this battery; it is not a ranking of the practices by importance, and a frequently-open wire is one these declarers did not declare, not one that failed.

9.5 Boundary cases

Two boundary cases fix the instrument's scope. A work item whose description is blank or non-text is a blocking intake defect: no practice is scored, the overall status is `NEEDS_REWORK`, and the single note asks the author to restate the work before assessment. An action tagged only `music` — a tag outside the reviewed vocabulary and matching no practice — returns `OUTSIDE_SCOPE`. That result does not mean the action is good, safe, or permitted; it means this positive-practice registry does not assess it. Red Line remains the separate refusal boundary, and an `OUTSIDE_SCOPE` verdict from Black Line grants nothing.

Finally, invalid review configuration is not silently normalized: a malformed ISO `as_of` value or a negative/non-integer freshness window raises before scoring. This is a configuration defect rather than a work finding, and it prevents a caller from mistaking an accidental date interpretation for a valid review.

10 Limits and Epistemic Boundaries

Black Line is self-declared and lexical. A person can choose the wrong tags, provide weak sources, write a ceremonial limitation, or declare a handoff that another reader cannot use. The evaluator matches labels; it does not inspect semantic truth, power relations, labor conditions, or downstream harm. An `ALIGNED` status is therefore a statement about the *presence* of declared evidence and nothing more — the strongest honest reading of it is "this work has laid out the pieces a reviewer would want," not "this work is correct." The formal propositions are careful about exactly this: Proposition 1 states that the finding is a projection of its surfaces — a reading of declared labels, not of the work — and Proposition 8 pins the scoring as deterministic. Under fixed inputs, the tests establish registry shape and deterministic scoring. Conditional monotonicity is exercised rather than asserted (Proposition 7): seeded permutation sweeps over incremental declarations confirm that once a declaration is non-empty the status never regresses, with the empty-declaration boundary pinned as the one intentional exception (Proposition 9), and the same sweep is drawn as the monotonicity lattice. None of that establishes anything about the world the labels point to (developed in the scholarship section).

The registry digest narrows one archival ambiguity but does not solve it. It shows which practice content was used and makes disagreement visible; it is not a signature, an external timestamp, or proof that the evidence record was not altered. Independent provenance would require a separate trust boundary and is left explicitly deferred.

10.1 Adversarial declarations

Because every input is self-declared, the instrument can be gamed by construction, and the honest response is to demonstrate the attacks rather than deny them. Each of the following was executed against the real evaluator.

Label-stuffing. The registry's evidence vocabulary contains 22 distinct labels. A `research`-tagged attempt that simply declares all 22 — with no artifact behind any of them — returns `ALIGNED`. The evaluator matches declared labels against required ones; it has no access to whether a declared `rerun` was ever run or a declared `reviewer` ever read anything. A stuffed declaration is lexically indistinguishable from a diligent one.

Tag-minimization. Tags select the practices an attempt is scored against, so narrowing the declared tags shrinks the review surface (see the coverage matrix). Executed: the same description with the same two declared labels (`data_origin`, `transform_log`) returns `NEEDS_REWORK` across 7 applicable practices when tagged `analysis` and `data`, and `ALIGNED` against the single applicable practice when tagged `data` alone. Both statuses are true statements about declaration coverage; the second is simply earned against a sevenfold smaller burden — 7 practices and 14 required labels narrowed to 1 and 2. (The registry's widest spread is eightfold, `research` at 8 practices against `data` at 1; this executed contrast starts from `analysis`, which reaches 7.) Whether the narrow tag set honestly describes the work is not a question the evaluator can pose.

Refresh-date laundering. Staleness reads declared dates, and dates are declarations too. Executed with the `data-provenance` practice under a 30-day window and review date 2026-07-01: a full two-label declaration whose evidence is 70 days old returns `NEEDS_EVIDENCE`, and the identical declaration with its dates rewritten to the review date returns `ALIGNED`. The evaluator cannot distinguish a genuine refresh — re-observing the data origin — from an edit to a date string.

These are instances of a well-documented dynamic, not defects unique to this design. Campbell observed that the more a quantitative indicator is used for decision-making, the more subject it becomes to corruption pressures that distort the very process it monitors [Campbell, 1979]; Strathern compressed the same dynamic into the aphorism that when a measure becomes a target, it ceases to be a good measure [Strathern, 1997]; and Power's study of audit cultures shows how systems built on checkable declarations drift toward producing auditable form rather than the substance the audit was meant to secure [Power, 1997]. A practice registry that certified quality would make these failure modes catastrophic, because a gamed status would launder bad work into apparent good work. Black Line's design response is to refuse the certifying role entirely: a status reports declaration coverage, so a gamed `ALIGNED` overstates nothing but coverage. The attacks also stay inspectable rather than hidden — the declared tag set, the declared labels, and the declared dates are the very record a reviewer reads, so a reviewer who asks "do 22 labels correspond to 22 artifacts?", "do these tags describe this work?", or "what changed at this refresh?" is asking questions the declaration itself exposes. The instrument narrows what gaming can counterfeit; it cannot remove the need for the human judgment those questions require, and it never converts any status into a safety score, an accreditation, or a permission.

10.2 Legibility bias

The instrument also has a bias toward legibility. Some important work is slow, embodied, tacit, relational, or not safely compressible into evidence labels, and a discipline that rewards what is easy to declare can quietly devalue what is hard to. The coarse-label design is a deliberate guard against false precision — it refuses to pretend it can score truth — but it cannot recover the aspects of strong work that resist declaration at all. Polanyi's account of tacit knowledge is a useful warning here: participation and skilled judgment are not merely missing fields waiting to be added to a form [Polanyi, 1958]. Situated action also means that a written plan cannot determine all later action [Suchman, 1987]. The White Line work exists partly to record what such a method leaves out.

10.3 A design claim, not an outcome claim

I am making a design and implementation claim, not an outcome claim. I have not run a user study, compared teams working with and without Black Line, or measured whether the practices improve correctness, speed, equity, or downstream decisions. Those are empirical questions needing a defined population, a comparator, an outcome measure, and governance review, and none of that is here. The evidence in this repository supports what the package computes and what its documentation asks a reviewer to inspect. It does not support a causal claim that using the instrument improves anything.

10.4 Bounded by the line set

Finally, Black Line is bounded by the rest of the line set on purpose. Golden Line addresses direction and aspiration; Red Line holds the refusal boundary; White Line marks absence, restraint, and unknowability. Black Line should not absorb those questions merely because they are difficult to measure, and a passing Black Line assessment never licenses anything a Red Line refusal would forbid. The discipline's contribution is to make ordinary rigor inspectable — not to guarantee it.

11 Conclusion

Black Line turns good-work intentions into small questions a collaborator can inspect: what is the question, where are the claims from, why is this method enough, what can fail, can it be rerun and reviewed, and what should happen next? Each question is a wire with declared evidence, each wire is situated against an older tradition of rigorous practice, and the evaluator that scores them is staged, deterministic under fixed inputs, and explicit about its own reach.

I built it to stay small. The tests establish that the registry is well-shaped and that fixed inputs produce a deterministic assessment carrying the registry digest; they establish nothing about whether a source is real or a result true. That gap is the design rather than a shortfall in it. A registry that certified quality would make every attack in the limits section catastrophic, because a gamed status would launder bad work into apparent good work; one that reports declaration coverage lets a gamed status overstate only coverage. Read alongside the formal method and the intellectual lineage, what a reader is left holding is a set of declarations another person can follow, and a plain account of everything those declarations do not settle (see the scholarship section).

Red Line remains the No document. Golden Line holds the higher thread. White Line marks absence, restraint, and unknowability. Black Line is the middle work: the positive discipline that helps an allowed project become clear enough to examine. It lives at `docxology/black_line`, beside the rest of that index.

References

Aristotle. *Nicomachean Ethics*. Hackett Publishing Company, Indianapolis, 2 edition, 1999. ISBN 9780872204645. URL https://archive.org/details/isbn_9780872204645. Translated, with introduction, notes, and glossary, by Terence Irwin. Accessed 2026-07-28.

Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, 2016. doi: 10.1038/533452a. URL https://www.nature.com/articles/533452a. Survey of 1,576 researchers. Accessed 2026-07-28.

Alberto Cairo. *The Truthful Art: Data, Charts, and Maps for Communication*. New Riders, 2016. URL https://www.oreilly.com/library/view/the-truthful-art/9780133440492/. Accessed 2026-07-18.

Donald T. Campbell. Assessing the impact of planned social change. *Evaluation and Program Planning*, 2(1):67–90, 1979. doi: 10.1016/0149-7189(79)90048-X. URL https://doi.org/10.1016/0149-7189(79)90048-X. Accessed 2026-07-22.

Jon F. Claerbout and Martin Karrenbach. Electronic documents give reproducible research a new meaning. In *SEG Technical Program Expanded Abstracts 1992*, pages 601–604. Society of Exploration Geophysicists, 1992. doi: 10.1190/1.1822162. URL https://library.seg.org/doi/abs/10.1190/1.1822162. Accessed 2026-07-28.

Harry Collins. *Tacit and Explicit Knowledge*. University of Chicago Press, Chicago, 2010. ISBN 9780226113807. URL https://press.uchicago.edu/ucp/books/book/chicago/T/bo8461024.html. Accessed 2026-07-28.

Hubert L. Dreyfus and Stuart E. Dreyfus. *Mind over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*. Free Press, New York, 1986. ISBN 9780029080610. URL https://archive.org/details/mindovermachinep00drey. Accessed 2026-07-28.

Richard P. Feynman. Cargo cult science. *Engineering and Science*, 37(7):10–13, 1974. URL https://calteches.library.caltech.edu/51/2/CargoCult.htm. Caltech commencement address. Accessed 2026-07-18.

Steven N. Goodman, Daniele Fanelli, and John P. A. Ioannidis. What does research reproducibility mean? *Science Translational Medicine*, 8(341):341ps12, 2016. doi: 10.1126/scitranslmed.aaf5027. URL https://pubmed.ncbi.nlm.nih.gov/27252173/. Accessed 2026-07-18.

Andrew Hunt and David Thomas. *The Pragmatic Programmer: From Journeyman to Master*. Addison-Wesley, Boston, 1999. URL https://www.oreilly.com/library/view/the-pragmatic-programmer/9780135956977/. Accessed 2026-07-18.

Sheila Jasanoff. Technologies of humility: Citizen participation in governing science. *Minerva*, 41:223–244, 2003. doi: 10.1023/A:1025557512320. URL https://doi.org/10.1023/A:1025557512320. Accessed 2026-07-18.

Donald E. Knuth. Literate programming. *The Computer Journal*, 27(2):97–111, 1984. doi: 10.1093/comjnl/27.2.97. URL https://academic.oup.com/comjnl/article/27/2/97/343244. Accessed 2026-07-18.

Robert K. Merton. The normative structure of science. In Norman W. Storer, editor, *The Sociology of Science: Theoretical and Empirical Investigations*, pages 267–278. University of Chicago Press, Chicago, 1973. URL https://press.uchicago.edu/ucp/books/book/chicago/S/bo28173694.html. Essay originally published 1942. Accessed 2026-07-18.

Marcus R. Munafò, Brian A. Nosek, Dorothy V. M. Bishop, et al. A manifesto for reproducible science. *Nature Human Behaviour*, 1:0021, 2017. doi: 10.1038/s41562-016-0021. URL https://www.nature.com/articles/s41562-016-0021. Accessed 2026-07-18.

National Academies of Sciences, Engineering, and Medicine. *Reproducibility and Replicability in Science*. National Academies Press, Washington, DC, 2019. doi: 10.17226/25303. URL https://nap.nationalacademies.org/catalog/25303/reproducibility-and-replicability-in-science. Consensus Study Report. Accessed 2026-07-18.

Brian A. Nosek, George Alter, George C. Banks, et al. Promoting an open research culture. *Science*, 348(6242):1422–1425, 2015. doi: 10.1126/science.aab2374. URL https://pubmed.ncbi.nlm.nih.gov/26113702/. Accessed 2026-07-18.

Open Science Collaboration. Estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716, 2015. doi: 10.1126/science.aac4716. URL https://www.science.org/doi/10.1126/science.aac4716. Accessed 2026-07-28.

Naomi Oreskes, Kristin Shrader-Frechette, and Kenneth Belitz. Verification, validation, and confirmation of numerical models in the earth sciences. *Science*, 263(5147):641–646, 1994. doi: 10.1126/science.263.5147.641. URL https://doi.org/10.1126/science.263.5147.641. Accessed 2026-07-18.

Elinor Ostrom. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, 1990. doi: 10.1017/CBO9780511807763. URL https://www.cambridge.org/core/books/governing-the-commons/7AB7AE11BADA84409C34815CC288CD79. Accessed 2026-07-18.

Roger D. Peng. Reproducible research in computational science. *Science*, 334(6060):1226–1227, 2011. doi: 10.1126/science.1213847. URL https://www.science.org/doi/10.1126/science.1213847. Accessed 2026-07-18.

Hans E. Plesser. Reproducibility vs. replicability: A brief history of a confused terminology. *Frontiers in Neuroinformatics*, 11:76, 2018. doi: 10.3389/fninf.2017.00076. URL https://www.frontiersin.org/journals/neuroinformatics/articles/10.3389/fninf.2017.00076/full. Accessed 2026-07-28.

Michael Polanyi. *Personal Knowledge: Towards a Post-Critical Philosophy*. University of Chicago Press, Chicago, 1958. URL https://press.uchicago.edu/ucp/books/book/chicago/P/bo19722848.html. Accessed 2026-07-18.

Karl R. Popper. *The Logic of Scientific Discovery*. Hutchinson & Co., London, 1959. URL https://www.routledge.com/The-Logic-of-Scientific-Discovery/Popper/p/book/9780415278447. English translation of Logik der Forschung (1934). Accessed 2026-07-18.

Michael Power. *The Audit Society: Rituals of Verification*. Oxford University Press, Oxford, 1997. ISBN 9780198289470. Accessed 2026-07-22.

Gilbert Ryle. *The Concept of Mind*. Hutchinson's University Library, London, 1949. URL https://archive.org/details/conceptofmind0000ryle. Accessed 2026-07-28.

Geir Kjetil Sandve, Anton Nekrutenko, James Taylor, and Eivind Hovig. Ten simple rules for reproducible computational research. *PLOS Computational Biology*, 9(10):e1003285, 2013. doi: 10.1371/journal.pcbi.1003285. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003285. Accessed 2026-07-18.

Donald A. Schön. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, New York, 1983. ISBN 9780465068760. URL https://www.hachettebookgroup.com/titles/donald-a-schon/the-reflective-practitioner/9780465068784/. Accessed 2026-07-28.

Richard Sennett. *The Craftsman*. Yale University Press, New Haven, 2008. ISBN 9780300119091. URL https://yalebooks.yale.edu/book/9780300151190/the-craftsman/. Accessed 2026-07-28.

Susan Leigh Star and James R. Griesemer. Institutional ecology, translations and boundary objects: Amateurs and professionals in berkeley's museum of vertebrate zoology, 1907–39. *Social Studies of Science*, 19(3):387–420, 1989. doi: 10.1177/030631289019003001. URL https://doi.org/10.1177/030631289019003001. Accessed 2026-07-18.

Marilyn Strathern. 'Improving ratings': Audit in the British university system. *European Review*, 5(3):305–321, 1997. doi: 10.1017/S1062798700002660. URL https://www.cambridge.org/core/journals/european-review/article/abs/improving-ratings-audit-in-the-british-university-system/FC2EE640C0C44E3DB87C29FB666E9AAB. Accessed 2026-07-22.

Lucy A. Suchman. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press, Cambridge, 1987. ISBN 0521331374. URL https://books.google.com/books?id=AJ_eBJtHxmsC. Accessed 2026-07-18.

Greg Wilson, Jennifer Bryan, Karen Cranston, Justin Kitzes, Lex Nederbragt, and Tracy K. Teal. Good enough practices in scientific computing. *PLOS Computational Biology*, 13(6):e1005510, 2017. doi: 10.1371/journal.pcbi.1005510. URL https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005510. Accessed 2026-07-18.