

White Line: A Typed Ledger for the Edge of the Claim

Keeping missing evidence, ethical boundaries, and open questions from becoming unsupported claims

Daniel Ari Friedman

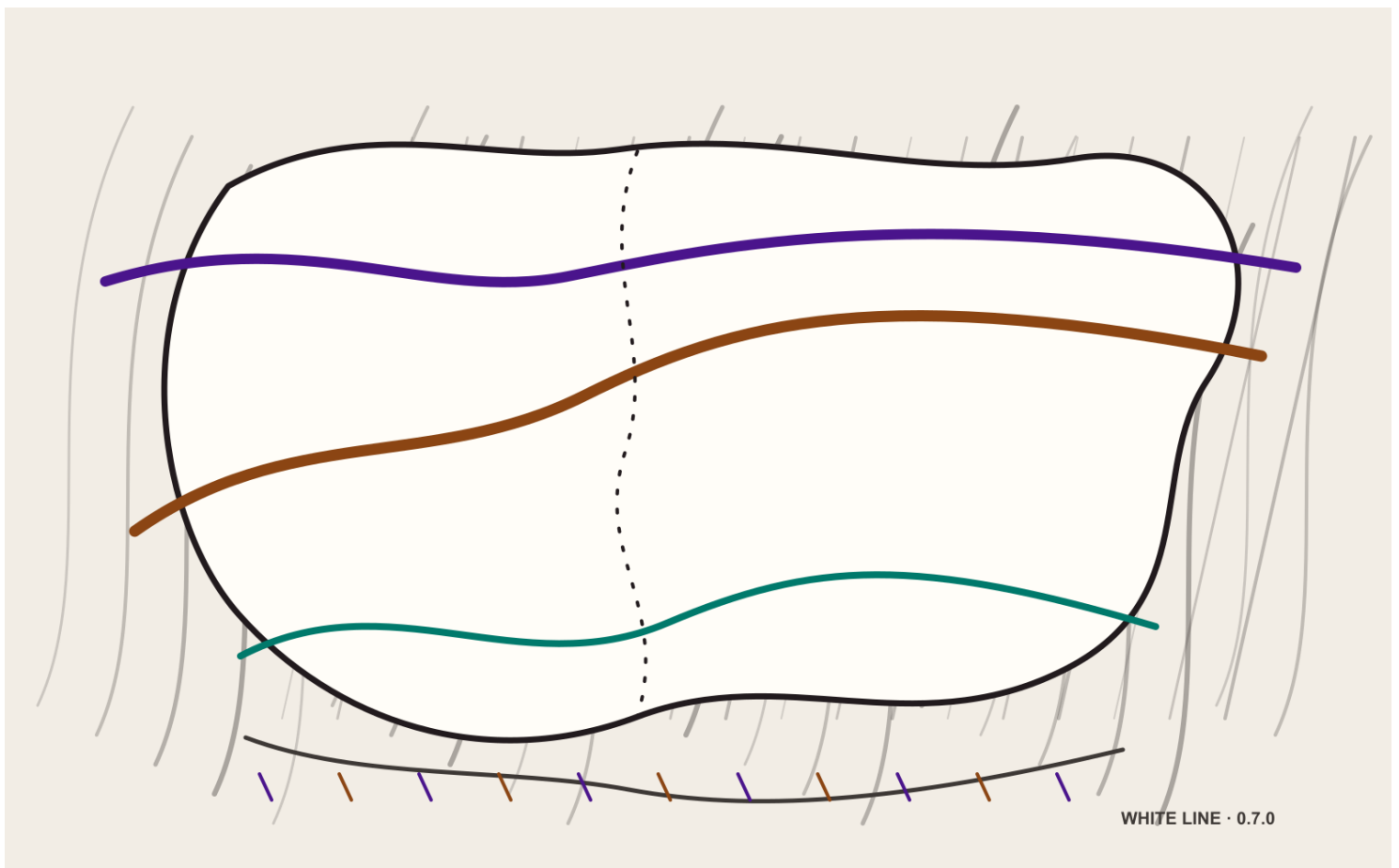
Active Inference Institute

`daniel@activeinference.institute`

ORCID: [0000-0001-6232-9096](https://orcid.org/0000-0001-6232-9096)

DOI: [10.5281/zenodo.21754242](https://doi.org/10.5281/zenodo.21754242)

2026-07-18



Contents

1	Abstract	2
2	Introduction: the discipline of not filling the gap	3
3	Relationship to the line set	4
3.1	Note on the name	4
4	Scholarship: epistemic boundaries, classification, and refusal	5
4.1	Epistemic absence: knowing the edge of what one knows	5
4.2	Classification and information infrastructures	5
4.3	Apophatic and contemplative traditions: the discipline of not saying	6
4.4	Ethical restraint: absence as an obligation to people	6
4.5	The translation test: what the instrument takes, and what it refuses	8
5	Method: three kinds of absence, four ledger states	10
6	Review protocol: from state to responsible follow-up	12
7	Reproducibility boundary and release chain	13
8	The three White Line layers	14
9	Formal method: the evaluator and its invariants	17
9.1	Domain objects	17
9.2	The staged evaluator	17
9.3	Semantic guarantees	19
9.4	The witness layer	24
9.5	Structural invariants	25
9.6	Where each claim is checked	25
10	Worked ledger entries	28
11	Reading one report distributionally	29
11.1	A worked intake under careless and adversarial input	29
11.2	A zero row is untriggered, not unreachable	29
11.3	Where the states sit across the three kinds	31
11.4	What comes due next, and what cannot come due at all	31
11.5	The state as a safe projection: the witness layer	31
12	Limits and safeguards	36
13	Conclusion	37

1 Abstract

White Line is a typed ledger for the edge of a claim. It records what is unknown or unobserved, what is intentionally withheld or unrepresented, and what is left open as contemplative negative space. It asks what a document must refuse to imply when the evidence, the obligation, or the language is not there. A gap is never treated as proof that a hidden cause exists.

The executable instrument is a small, pure-data Python package: a versioned registry of eleven absence records and a staged evaluator, `assess_absence`. It is a tested prototype for structured review, not a validated measurement instrument and not an observation of the world. Each record carries one of three kinds — `EPISTEMIC`, `ETHICAL`, or `CONTEMPLATIVE` — and a dated observation assigns it one of four states: `NAMED`, `UNRESOLVED`, `WITHHELD`, or `NOT_RECORDED`. Intake sets aside any input that fails the contract, whether a non-observation, an unknown record, an untyped state, or an attempt to assert the ledger-reserved `NOT_RECORDED`, and records why. Conflicting and duplicate observations leave the same visible trace.

Staleness is the second stage. A dated naming that ages past its record's review horizon decays back to `UNRESOLVED`, because an old naming is no longer current evidence for a settled state. Six structural invariants guard the registry's shape, and a shipped battery hands each one a registry carrying exactly the defect it guards against, so a check that has never rejected anything is not counted as protection. A bounded follow-up protocol makes the next human action explicit without issuing an approval.

The result is a ledger, not a detector of hidden causes, and not a claim that an absence by itself supports an inference beyond the record's scope. Reports retain their review date and registry digest so a later reader can tell a reproducible rerun from an unanchored assertion. White Line is the fourth work in the line set, deliberately non-redundant with Red Line's security boundary, Black Line's positive practice, and Golden Line's aspirational direction: where those lines fill, commit, and reach, White Line marks the edge and holds it open.

2 Introduction: the discipline of not filling the gap

Research and engineering reward completed surfaces: a full dataset, a clean narrative, a confident conclusion. But the missing observation, the withheld testimony, and the deliberate silence are often part of the truth of a work. I wrote White Line to give those absences a place to remain different from each other, because the pressure I feel while writing is not to hide a gap so much as to let three unlike things blur into one apology.

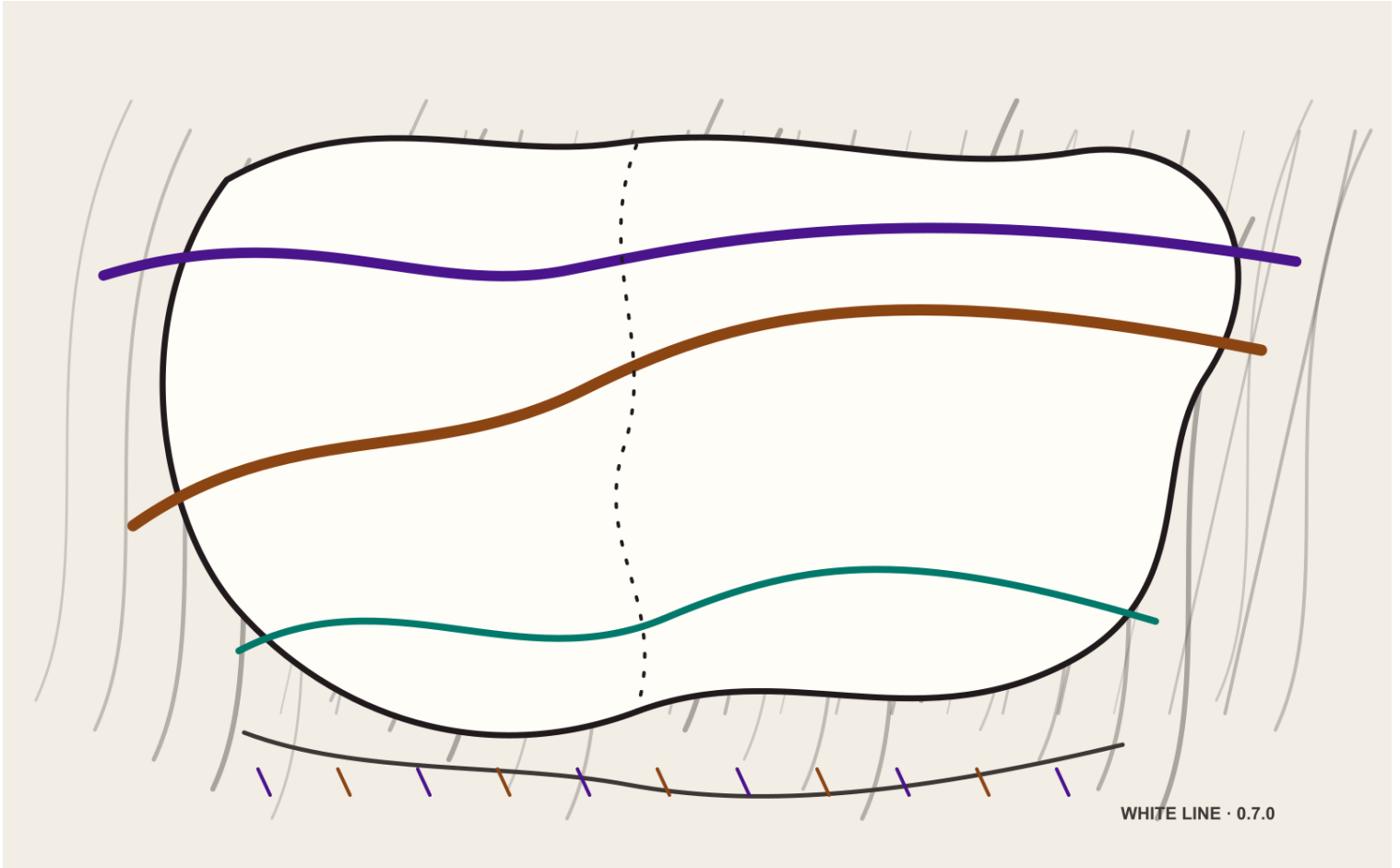


Figure 1: White Line’s cover art: congested graphite marks press toward an irregular open field while three typed traces cross it without closing it. The art is a deterministic conceptual composition for interpretation, not evidence of a hidden cause.

The concern is old. Nāgārjuna warns against turning conceptual absence into a hidden substance [Nāgārjuna, 1995]; Maimonides develops negative ways of speaking about what cannot be positively attributed [Maimonides, 1963]; Du Bois’s veil shows how a social order can make a lived perspective structurally unseen rather than simply missing from a spreadsheet [Du Bois, 1903]. These sources are historically and intellectually distinct, and I do not flatten them into one theory. Contemporary work on produced ignorance, missing-data assumptions, classification, and epistemic injustice puts methodological and ethical pressure on the same design problem, and the scholarship section names those bridges without treating them as interchangeable [Sullivan and Tuana, 2007, Rubin, 1976, Bowker and Star, 1999, Fricker, 2007].

The personal security boundary remains Red Line. Black Line describes how to make work strong and inspectable. Golden Line names what a project hopes to serve. White Line refuses to pretend that all gaps are solvable, that all silence is permission, or that all mystery is a causal explanation.

The paper and the package are separate but linked objects. The package imports as a small library; the manuscript states the scope, limits, and review protocol that keep its output from being overread.

The paper is structured as follows. Section sec. 5 defines the three kinds of absence and the four ledger states. Section sec. 9 states the evaluator and its invariants formally. Section sec. 10 walks through worked ledger entries, and Section sec. 12 closes with the instrument’s safeguards and limits.

3 Relationship to the line set

White Line is the absence work in the four-line set, whose shared framing is declared by the set reader `line_set`. Red Line is the personal security boundary and explicit No document; Black Line describes strong practice; Golden Line describes aspiration. White Line copies none of their registries or evaluators, and a gap or restraint is never evidence that an unseen cause exists.

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 White Line does not import, depend on, or defer to it.

3.1 Note on the name

The four line works — Black, White, Golden, Red — carry colors that openly echo the stages of the alchemical *magnum opus*, and White Line stands for **Albedo**, the whitening. Read only in Carl Jung’s symbolic-psychological register of individuation, albedo is the washing that follows the blackening: a figure for purification, clarification, and cleared negative space — never an empirical, mystical, or causal event. The emblem fits this instrument precisely because it asserts nothing; whitening here is restraint and the discipline of the unwritten, not a sign that some absent or hidden thing is real, safe, or at work. The set’s working order — refuse, method, aspire, absence — is functional, not a reenactment of the opus’s nigredo → albedo → citrinitas → rubedo sequence; the shared palette is a label, not a ritual. The full framing, and the single Jung citation for the whole set, live in the set reader `line_set`.

4 Scholarship: epistemic boundaries, classification, and refusal

I did not invent the idea that absence deserves careful handling. Several conversations have been running on what a serious account should refuse to fill in, and they are not one lineage: they disagree about whether ignorance is a limit, a social achievement, an institutional resource, a harm, or a protected refusal. The disagreement is what makes them useful here, because it stops the instrument from treating every gap as the same kind of object. My contribution is narrower than any of them — a typed, versioned ledger that keeps three kinds of absence from collapsing into one unsupported claim. Throughout this section I separate intellectual resonance from direct derivation, and scholarship from validation. The sources constrain the design; they do not prove that the package’s categories are complete, or that a recorded absence has a cause.

4.1 Epistemic absence: knowing the edge of what one knows

One classical strand is the Socratic disavowal of knowledge one does not possess. In the *Apology*, Socrates locates his only advantage in not imagining he knows what he does not [Plato, 2002]. This is not skepticism for its own sake; it is a working rule for inquiry, and it is exactly the posture White Line encodes in its EPISTEMIC kind. An unobserved dependency or an unmeasured uncertainty is recorded as a named gap, not converted into a confident value. The analogy is deliberately modest: Socratic humility is a philosophical posture, whereas White Line is a software contract for a supplied record.

Feminist epistemology adds a second correction to the idea of a neutral gap. Haraway’s account of situated knowledges treats knowledge as partial and located rather than as a view from nowhere [Haraway, 1988]. The point is not that every perspective is equally reliable; it is that the position, interests, and accountability of an account belong to the conditions under which it can be assessed. White Line records dates, provenance, and review prompts, but it does not model a reviewer’s social position or generate what Harding calls strong objectivity. That missing capacity is a human and institutional responsibility, not a feature the ledger should pretend to possess.

The study of ignorance makes the stronger point that nonknowledge is not always an innocent remainder. Proctor and Schiebinger’s agnotology proposes ignorance as a subject in its own right — something made and unmade, with a history and a politics, rather than the empty space left over where knowledge has not yet arrived [Proctor and Schiebinger, 2008]. Mills’s account of white ignorance and Sullivan and Tuana’s broader epistemologies of ignorance describe ignorance as something that can be organized, reproduced, and attached to racialized power [Mills, 2007, Sullivan and Tuana, 2007]. McGoey further shows how ignorance can serve as an institutional resource, including by distributing responsibility and preserving room for action under uncertainty [McGoey, 2012]. White Line therefore asks a reviewer to name a gap and its provenance when known, but it does not diagnose motive or institutional strategy from a state label. UNRESOLVED is a prompt for investigation, not a theory of how ignorance was produced.

The field’s sharpest cases are the ones a ledger cannot reach. Oreskes and Conway trace how a small network of scientists ran sustained campaigns on tobacco and climate, the manufactured appearance of unsettled science being itself the product [Oreskes and Conway, 2010]. Rayner describes the quieter institutional version: an organization must simplify to function, and uncomfortable knowledge that will not fit is held off by denial, dismissal, diversion, and displacement [Rayner, 2012]. Gross and McGoey’s handbook collects the range between those registers [Gross and McGoey, 2015]. Each describes a process with agents, interests, and a history. White Line records that a category of absence was named, and when; the ledger cannot tell a manufactured gap from an ordinary one, and nothing in a state label is evidence that a campaign or strategy produced it. That question stays with the reviewer, which is why every record carries a prompt rather than a score.

Statistical missing-data theory supplies a complementary warning. Rubin’s framework, and the later treatment by Little and Rubin, make inference depend on assumptions about the process that generated missingness [Rubin, 1976, Little and Rubin, 2019]. In the familiar MCAR, MAR, and MNAR distinctions, the label of a missing value does not by itself identify the mechanism or justify an estimate. Two results sharpen how little the data can settle. Little’s global test can reject MCAR, but rejecting one mechanism is not identifying another [Little, 1988]. Molenberghs, Beunckens, Sotito, and Kenward prove the stronger point: every MNAR model has an MAR counterpart that fits the observed data equally well, so observed data can never adjudicate between them [Molenberghs et al., 2008]. The mechanism is an assumption, argued rather than discovered. Manski’s partial-identification programme takes the disciplined alternative — report the bounds the data and stated assumptions actually support instead of a point estimate that requires more [Manski, 2003]. White Line’s refusal is narrower: it estimates nothing, not even an interval. It records that an observation is absent or unresolved, preserves the date and intake trace, and leaves both mechanism and bound to a method that states its assumptions. This is a scope decision, not a claim that missingness is unmodelable.

4.2 Classification and information infrastructures

Bowker and Star show that categories and standards are not neutral containers: they organize information infrastructures, make some work visible, and render other perspectives difficult to see [Bowker and Star, 1999]. That insight explains why White Line treats its registry as a versioned object with invariants rather than as a loose list of labels. A category has a history, a review horizon, and a non-claim that must remain inspectable.

D’Ignazio and Klein extend this concern into contemporary data practice. Their data-feminist account treats data work as a field of power, emphasizes that classification systems can reproduce hierarchy, and rejects the fantasy that data speak without human and institutional mediation [D’Ignazio and Klein, 2020]. So the counts stay descriptive, colour and marker shape carry the same distinction redundantly, and every figure states what it does not establish. The registry is a small audit surface for classification choices. Versioning can expose a category’s history and make revision inspectable; it cannot make a finite registry neutral, exhaustive, or politically innocent.

4.3 Apophatic and contemplative traditions: the discipline of not saying

A different conversation concerns what should be left unsaid on principle. The apophatic theology of Pseudo-Dionysius proceeds by negation — approaching what exceeds speech by removing predicates rather than adding them [Pseudo-Dionysius the Areopagite, 1987]. Maimonides develops a parallel negative predication, holding that certain subjects are better protected by what one declines to assert than by a confident positive attribution [Maimonides, 1963]. Nāgārjuna’s analysis of emptiness supplies an important caution for any absence ledger: emptiness itself must not be reified into a hidden substance, or the cure becomes the disease [Nāgārjuna, 1995]. These are not historical sources for a software state machine, and they should not be made interchangeable with privacy or justice theory. They offer a limited conceptual resonance for White Line’s CONTEMPLATIVE kind: a boundary can be meaningful without naming an object beyond it.

Two readings of that tradition mark where the resonance stops. Sells argues that apophatic language performs rather than describes: each saying is turned back on itself and unsaid, and the meaning lives in the regress rather than in any statement the regress comes to rest on [Sells, 1994]. Turner presses a related correction — the medieval negative tradition is not a report of an extraordinary inner experience, and reading it as one converts a discipline of language into a claim about a state of mind [Turner, 1995]. Both warn against the mistake an absence ledger is most likely to make. White Line’s contemplative records perform no regress and report no experience. They are ordinary typed rows saying that a question is being held open, and their year-long review horizons are a scheduling decision, not a spiritual one. The tradition contributes the discipline of not filling; it contributes nothing about what, if anything, lies past the boundary, and the ledger asserts nothing there either.

Wittgenstein gives the strand a sharp modern formulation: the *Tractatus* closes on the injunction that whereof one cannot speak, thereof one must be silent [Wittgenstein, 1922]. Keats names the temperamental capacity that makes such restraint bearable rather than anxious — the “Negative Capability” of remaining in uncertainty without irritable reaching after fact and reason [Keats, 1958]. Merton treats silence in a contemplative register, not as an absence of content but as a protected space that a hurried account would destroy by rushing to fill it [Merton, 1961]. John Cage carries the same insight into art: his timed silent composition is not empty but framed, demonstrating that structured negative space is a positive compositional act rather than a failure to produce sound [Cage, 1961]. White Line’s fallow-ground and ineffable-remainder records are the ledger’s small, secular echo of these commitments — a way of writing down that something is deliberately left open.

4.4 Ethical restraint: absence as an obligation to people

A third conversation treats some absences as duties rather than gaps. Glissant’s “right to opacity” argues that the demand for full transparency about another person or culture can itself be a form of violence, and that respecting what one is not entitled to know is an ethical stance, not a limitation to be overcome [Glissant, 1997]. Du Bois’s figure of the veil shows the other side of the same problem: a social order can render a lived perspective structurally unseen, so that its absence from an account reflects the account’s blind spot rather than the perspective’s non-existence [Du Bois, 1903]. These sources ground White Line’s ETHICAL kind, where withheld material and an unrepresented perspective are recorded as obligations — something the ledger asks a reviewer to sit with — rather than as missing data to be recovered at any cost.

Refusal is not merely a privacy preference or a defective data point. Simpson’s account of Mohawk political life treats refusal as a practice of sovereignty against incorporation into categories imposed by settler institutions [Simpson, 2014]. Tuck’s critique of damage-centered research adds a methodological warning: documenting pain can reproduce a one-dimensional account of a community even when the stated purpose is advocacy [Tuck, 2009]. White Line cannot adjudicate sovereignty or decide whether a project is damage-centered, but these works change the review question. Before asking how to complete a record, a reviewer must ask who benefits from completion, who bears its exposure, and whether the request itself repeats the harm it claims to document.

Fricker names two ways epistemic practice can wrong people specifically as knowers: testimonial injustice and hermeneutical injustice [Fricker, 2007]. Dotson’s account of epistemic violence makes the failure to meet a speaker’s vulnerability under conditions of silencing more explicit [Dotson, 2011]. Medina extends this into an account of epistemic resistance and shared responsibility: the problem is not only whether a claim is recorded, but whether social conditions allow people to participate as knowers [Medina, 2013]. These works prevent a dangerous shortcut in the instrument: WITHHELD is not proof of deception, and NOT_RECORDED is not proof that nobody had knowledge. The ledger can preserve a boundary and prompt responsible review; it cannot decide whether a social interaction was just, whether a refusal is politically justified, or whose testimony should carry authority.

Nissenbaum’s contextual-integrity framework gives the privacy side of the same constraint: information flow is appropriate only

relative to the norms of a specific context [Nissenbaum, 2004]. White Line consequently distinguishes “not disclosed” from “not observed” and makes the reason for a follow-up a human question rather than an automatic demand for completion. The state vocabulary is therefore deliberately non-diagnostic: it names the relation of a record to the supplied observation, not the moral character of the person or institution behind it.

4.5 The translation test: what the instrument takes, and what it refuses

The scholarship yields no algorithm. It yields pressure on the design: each source asks what a naïve absence ledger would erase, assume, or expose. I translate only part of that pressure into code, and the incompleteness is the point:

- **Locate the record.** Ask who supplied the observation, from what position, under which institutional conditions, and with what access. The package stores date and provenance but does not encode positionality or produce situated knowledge [Haraway, 1988, Harding, 1992].
- **Type the limit.** Ask whether the boundary is epistemic, ethical, or contemplative; classification is itself consequential. See Definition 1.
- **Model or decline the mechanism.** If the claim requires an explanation of missingness, use a method that states its assumptions. White Line records the unresolved state but does not estimate MCAR, MAR, MNAR, or a causal process [Rubin, 1976, Little and Rubin, 2019].
- **Protect refusal.** Ask who benefits from completion and who bears exposure; a withheld or unrepresented record is not automatically a deficit to repair [Tuck, 2009, Simpson, 2014, Nissenbaum, 2004].
- **Bound and revisit.** Preserve what is absent without converting missingness, opacity, or silence into a cause, motive, or verdict; retain date, provenance, and the next human question where power can make testimony unheard [Fricker, 2007, Dotson, 2011, Medina, 2013, Sullivan and Tuana, 2007].

The package implements a narrow subset of these moves: it types records, bounds state transitions, and returns a bounded follow-up directive. It does not model missingness mechanisms, locate a knower, or protect a refusal by itself. Those remain governance tasks, and the split between them is the central limit that keeps a review instrument from impersonating a social epistemology.



Seven conversations, one bounded translation

White Line does not turn distinct literatures into one theory.
It asks what each prevents a small absence ledger from erasing or exposing.

NOT AN EXHAUSTIVE LITERATURE REVIEW · CONSTRAINTS IN, LIMITED OPERATIONS OUT

FIELD, SOURCES, AND THE QUESTION FORCED		WHITE LINE RESPONSE (LIMITED)
	SITUATED KNOWLEDGE Haraway · Harding where is the account located?	record date and provenance; ask about position reviewer responsibility
	PRODUCED IGNORANCE Proctor · Schiebinger · Mills · McGoey how was the gap made or maintained?	name provenance; do not infer motive reviewer responsibility
	MISSINGNESS Rubin · Little what mechanism would justify inference?	do not impute a mechanism: not MCAR, MAR, or MNAR reviewer responsibility
	CLASSIFICATION Bowker · Star · D'Ignazio · Klein what does this category expose or hide?	version the registry; do not call it neutral reviewer responsibility
	JUSTICE + RESISTANCE Fricker · Dotson · Medina who can participate as a knower?	a state label is not authority, diagnosis, or a verdict reviewer responsibility
	REFUSAL + OPACITY Tuck · Simpson · Nissenbaum · Glissant who bears the cost of completion?	access is not entitlement; refusal is not a deficit reviewer responsibility
	DISCIPLINED RESTRAINT Nāgārjuna · Wittgenstein · Keats · Cage what should remain unasserted?	open space is not a hidden cause; a remainder stays unsaid reviewer responsibility
IMPLEMENTED THIN SLICE		
TYPE what kind of limit	BOUND what must not be inferred	REVISIT what changed context reopens

The literature supplies reasons for restraint; the package supplies a bounded trace,
not a theory of justice or a detector of hidden causes.

Figure 2: The scholarship-to-design matrix connects situated knowledge, produced ignorance, missing-data methodology, classification, epistemic justice, refusal, opacity, and contemplative restraint to White Line's limited operations: type, bound, and revisit. It distinguishes reviewer responsibilities from code behavior; it is not an exhaustive literature review or a claim that the traditions are one theory.

5 Method: three kinds of absence, four ledger states

An absence record has an identifier, a title, a kind, a bounded description, a reviewer’s `prompt`, and a `review_horizon_days` bound. The kind answers what sort of limit is being recorded:

- **Epistemic absence** concerns what has not been observed, measured, accessed, or resolved.
- **Ethical restraint** concerns what is withheld, unrepresented, or not disclosed because a duty of care, consent, or safety boundary matters.
- **Contemplative negative space** concerns what is intentionally left open beyond a useful claim. It does not assert a literal invisible force.

An observation assigns a record one state. `NAMED` says the absence has been made explicit. `UNRESOLVED` says it is known but not settled. `WITHHELD` says material is intentionally not disclosed. `NOT_RECORDED` is emitted when no observation exists — and it is reserved for the ledger itself, so an observation that tries to assert it is set aside at intake. The analyzer does not infer intent from silence, promote a gap to a cause, or use one kind of absence as evidence for another.

The assessment is staged rather than a single lookup. Intake normalizes and screens the input, matching keeps at most one contract-passing observation per record, and scoring assigns each record its state. Two rules run through the whole pipeline. First, nothing is discarded quietly: every set-aside input, resolved conflict, and same-state duplicate leaves a typed event and a note in the report. Second, caution only rises: when observers disagree the ledger keeps the more cautious state, and a dated `NAMED` observation that ages past its record’s horizon is reported `UNRESOLVED` until it is re-reviewed. Under this contract no input can make an absence look more settled than its most cautious observer reported. A malformed custom registry or an invalid review date fails closed. The formal-method section states each of these behaviors, and the six structural invariants that guard the registry’s shape, as definitions and propositions.

The registry is deliberately finite and versioned. Projects can extend it, but an extension should state its kind, its scope, and the non-claim it protects; the evaluator refuses an extension that fails the structural checks. The report’s date and registry digest should be stored with any downstream review.

THE REVIEW ATLAS

From a missing context to a bounded next question

A small instrument with a deliberately visible boundary: type the gap, reconcile the record, then choose a human next step.

11 REGISTRY RECORDS · 3 KINDS · 4 STATES · DIGEST 11047a64

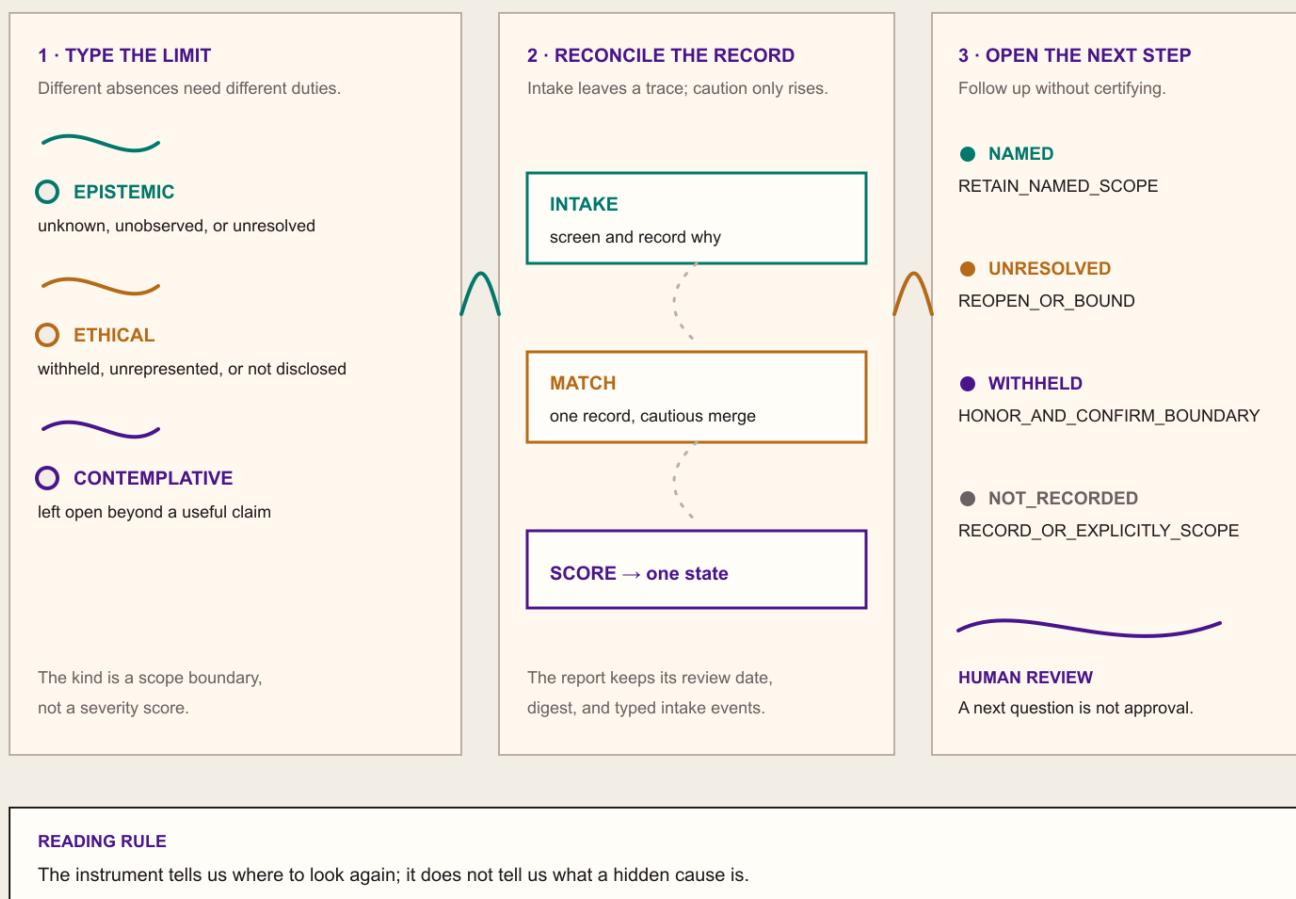


Figure 3: The White Line review atlas: type the limit, reconcile the record, and open a bounded next question. Counts and actions are derived from the live registry, state alphabet, and review protocol; the atlas is not a summary of an observed population.

6 Review protocol: from state to responsible follow-up

The evaluator reports a state; it does not decide what a person or project may do. White Line therefore keeps a second record explicit: a bounded review directive for each finding. `review_directives(report)` (formalized as [Definition 14](#) and [Proposition 7](#)) maps `NOT_RECORDED` to recording or explicitly scoping the unreviewed category, `UNRESOLVED` to reopening or bounding the question, `WITHHELD` to honoring and confirming the boundary without seeking disclosure, and `NAMED` to retaining the naming without treating it as resolution. None of these directives is an approval, compliance, safety, or truth verdict.

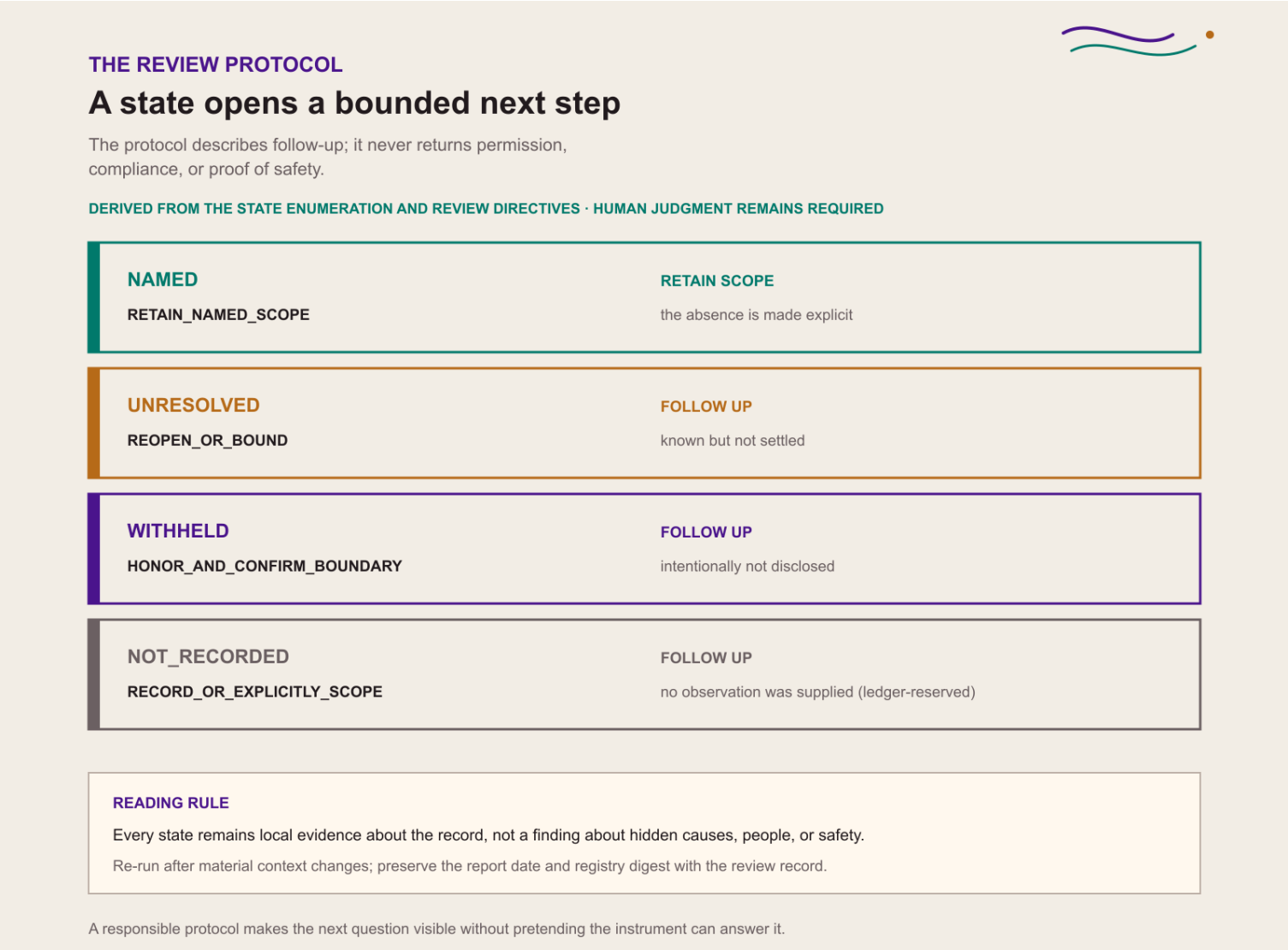


Figure 4: The bounded White Line review protocol maps each state to human follow-up without producing permission, compliance, or safety approval. It is derived from the state enumeration and review directives, not from observed safety data.

The operator loop is deliberately short:

1. Fix a review date and retain the registry digest.
2. Supply only observations the reviewer is authorized to record.
3. Inspect every typed intake event; a set-aside input is a limitation of the review, not an invisible rejection.
4. Read every finding and its directive, preserving withholding where a duty, consent condition, or safety boundary requires it.
5. Re-run after material context changes or after a dated naming passes its review horizon.

The resulting report is useful precisely because it does not collapse these steps into a score. The protocol makes the next human question visible while leaving independent verification, consent, security review, and domain judgment where they belong.

7 Reproducibility boundary and release chain

A White Line result is reproducible only when its context travels with it. Every `WhiteReport` records the ISO `review_date`, the order-independent `registry_digest`, and typed `intake_events` (see [Proposition 6](#)). The date fixes staleness accounting; the digest fixes which registry defined each id. A report without those fields can still be read, but it should not be treated as a complete review record.

The publication artifacts follow the same source-to-output chain. The live registry generates the deterministic figures and their registry JSON; the manuscript names those figures and its bibliography keys; the sibling template renders PDF and HTML; and `scripts/audit_project.py` checks that the generated bundle contains the source headings, figure links, bibliography keys, and required artifacts. A source manifest is written only after that audit passes, and its content fingerprint becomes stale after any manuscript or figure-input change.

The gate is structural, not epistemic. It can detect that a rendered bundle is missing a source section or that a figure no longer matches the ledger. It cannot verify a reviewer’s note, establish independent evidence, justify a withholding, or prove that a project is safe. Those are separate human and domain-governed questions.

8 The three White Line layers

The initial registry keeps three layers distinct:

- 1. **Epistemic:** the ledger records a gap rather than inventing its value or cause. Its records — an unobserved dependency, an unmeasured uncertainty, a missing null result, an unasked question — all name something a reviewer could in principle go and check.
- 2. **Ethical:** withheld material, an unrepresented perspective, uncredited labor, and an unacknowledged limitation concern obligations to people, not merely missing data. Some of these absences are duties to keep rather than gaps to close.
- 3. **Contemplative:** negative space, fallow ground, and an ineffable remainder protect the possibility that a useful account can remain incomplete without being secretly completed by the author. Their long review horizons reflect that they are meant to be revisited slowly, not resolved on a sprint.

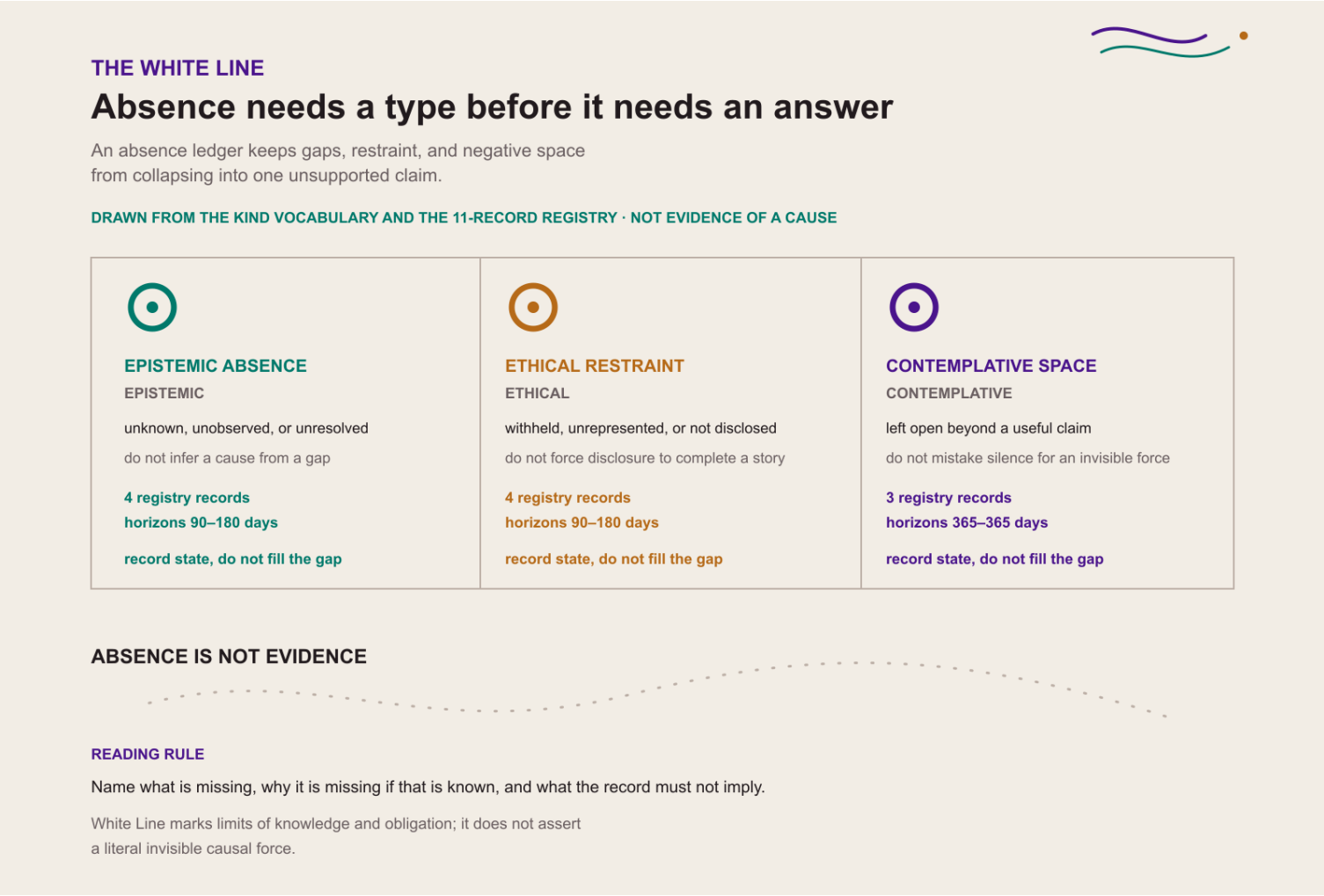


Figure 5: White Line separates three kinds of absence: epistemic absence, ethical restraint, and contemplative space. Each panel is generated from the **AbsenceKind** vocabulary and the shared kind glosses, with its record count and review-horizon range read from the live versioned registry, and each panel keeps a guard against overclaiming. The lower dotted field represents an open boundary rather than a hidden mechanism. The figure is a deterministic schematic of the typology; it is not evidence of a causal force, a completeness score, or a diagnosis of the people involved.

The current registry is small, finite, and reviewable in full. Each record names one bounded category of absence, states the reviewer’s question, and declares how long a dated naming stays fresh before it must be looked at again. The review horizons are graded by kind: epistemic gaps and ethical restraints both come due within 90–180 days of a dated naming, while contemplative spaces are given a full year because they are meant to ripen rather than be closed.

Stating the grading as a per-kind range understates how differently the three behave. Placed on a shared days axis, the epistemic and ethical records spread across three horizon values each, while all three contemplative records sit on one. That uniformity is a commitment rather than an accident: there is no useful sense in which one ineffable remainder ripens faster than another, so the ledger asks about all of them on the same annual cadence.

THE WHITE LINE REGISTRY

11 absence records across 3 kinds

Finite, versioned, and deliberate: each record names one bounded gap, restraint, or negative space.

GENERATED FROM THE VERSIONED ABSENCE REGISTRY · NOT A COMPLETENESS SCORE

EPISTEMIC

unknown, unobserved, or unresolved

4 records

- **Unobserved dependency**

id: unobserved-dependency

review horizon: 90 days

- **Unmeasured uncertainty**

id: unmeasured-uncertainty

review horizon: 120 days

- **Missing null results**

id: missing-null-results

review horizon: 180 days

- **Unasked question**

id: unasked-question

review horizon: 90 days

ETHICAL

withheld, unrepresented, or not disclosed

4 records

- **Withheld material**

id: withheld-material

review horizon: 180 days

- **Unrepresented perspective**

id: silenced-perspective

review horizon: 120 days

- **Uncredited labor**

id: uncredited-labor

review horizon: 180 days

- **Unacknowledged limitation**

id: unacknowledged-limit

review horizon: 90 days

CONTEMPLATIVE

left open beyond a useful claim

3 records

- **Contemplative negative space**

id: negative-space

review horizon: 365 days

- **Fallow ground**

id: fallow-ground

review horizon: 365 days

- **Ineffable remainder**

id: ineffable-remainder

review horizon: 365 days

A record's presence names a category worth reviewing; it is never a claim that the category is populated in any given work.

Figure 6: The current versioned absence registry, grouped by kind, with every record's review horizon in days. The figure is generated directly from the registry data structure; the audit gate binds the figure registry, its digest, and the raster bytes of each shipped image, so a stale or hand-edited plate fails the gate rather than passing quietly. A record's presence names a category worth reviewing; it is never a claim that the category is populated in any particular work.

THE HORIZON DISPERSION

How often each kind comes back for review

One dot per registry record, positioned by its `review_horizon_days`.
Kinds differ by colour and by marker shape; a horizon is a review cadence,
not a rate at which the world changes.

READ FROM THE 11-RECORD REGISTRY · NOT A PRIORITY, SEVERITY, OR COMPLETENESS SCORE

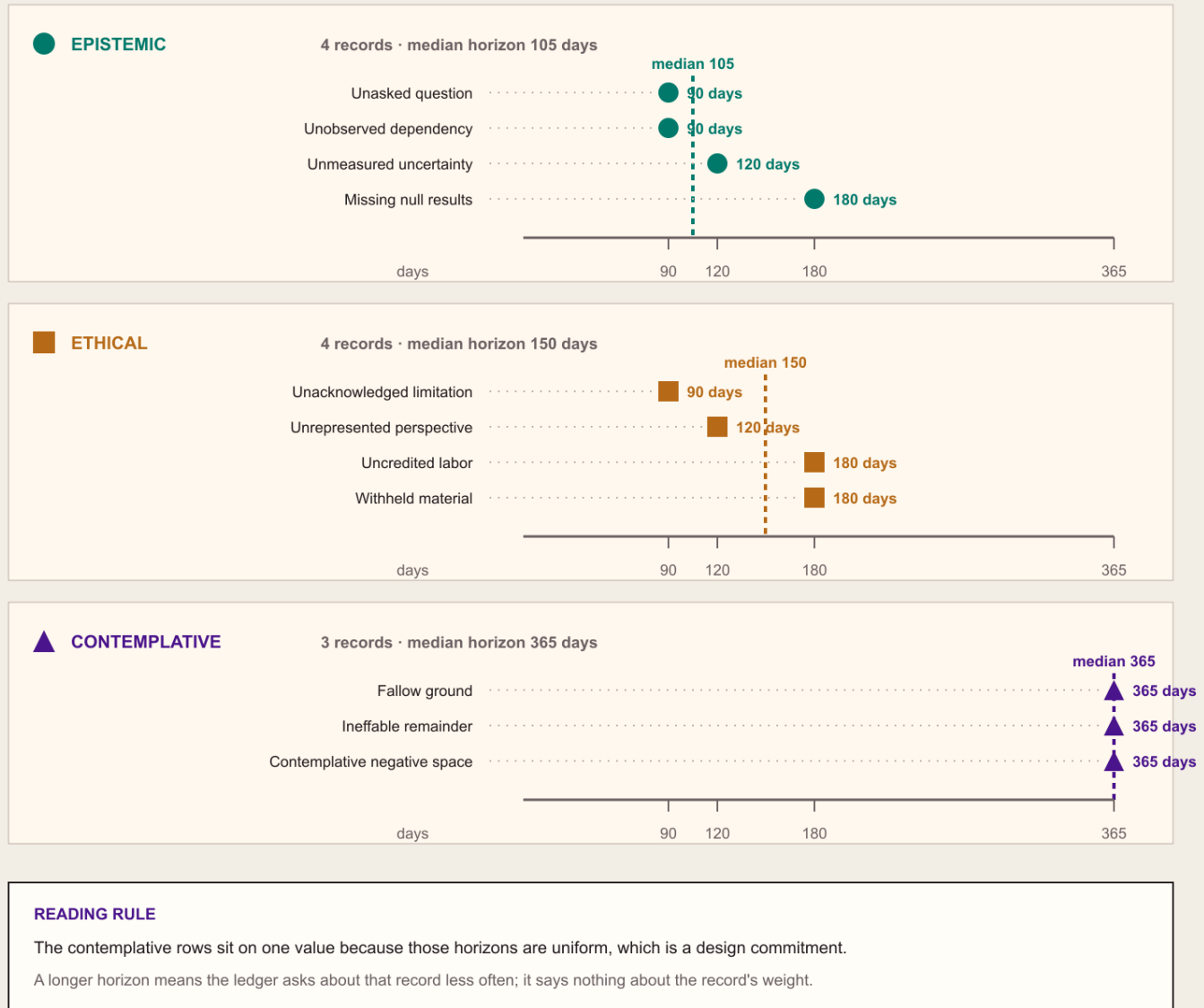


Figure 7: Every registry record placed on a shared days axis by its `review_horizon_days`, grouped into one band per `AbsenceKind` and keyed by colour and by marker shape, with each kind's median horizon marked. Read from `WHITE_RECORDS`. The dispersion shows a structural asymmetry the per-kind range compresses away: the contemplative horizons are uniform at one value while the epistemic and ethical horizons spread across several. A horizon is the cadence on which the ledger asks about a record again; it is not a claim about how fast the world changes, and a longer horizon is not a lower priority.

9 Formal method: the evaluator and its invariants

The three kinds (Definition 1) and four states (Definition 2) just described are restated formally below.

This section states, as definitions and propositions, exactly what the executable instrument computes. The formalism describes the code; it does not extend it. Every enumerated value, decision branch, and structural check below is present in the `white_line` package, each named guarantee is exercised by a test, and where the prose states a count, that count is re-derived from the registry and the enumerations rather than typed. Numbering is generated at render time from the labels, so a block inserted here rennumbers every reference to it.

9.1 Domain objects

Definition 1 (Kinds). The kind alphabet is the set $K = \{\text{EPISTEMIC}, \text{ETHICAL}, \text{CONTEMPLATIVE}\}$, with $|K| = 3$. A kind answers *what sort of limit* a record describes.

Definition 2 (States). The state alphabet is $S = \{\text{NAMED}, \text{UNRESOLVED}, \text{WITHHELD}, \text{NOT_RECORDED}\}$, with $|S| = 4$. `NOT\_RECORDED` is reserved for the ledger itself and marks the meta-gap of a record for which no observation was supplied; an observation may not assert it.

Definition 3 (Absence record). An absence record is a frozen tuple $r = (id, title, kind, description, prompt, review_horizon_days)$ with $kind \in K$ and a review horizon $review_horizon_days \in \mathbb{Z}^+$ measured in days. The `prompt` is the question a reviewer sits with; the horizon bounds how long a dated naming stays fresh.

Definition 4 (Registry). The registry is the finite ordered tuple $R = (r_1, \dots, r_{11})$ of 11 records. Partitioned by kind it contains exactly 4 epistemic, 4 ethical, and 3 contemplative records, and every horizon lies in $\{90, 120, 180, 365\}$ days. The registry is versioned; its canonical digest is a change-review instrument with no safety semantics.

Definition 5 (Caution order). Caution is the total order induced by the rank map $c : S \rightarrow \{0, 1, 2, 3\}$,

$$c(\text{NOT_RECORDED}) = 0 < c(\text{NAMED}) = 1 < c(\text{UNRESOLVED}) = 2 < c(\text{WITHHELD}) = 3.$$

Withholding outranks an open question, which outranks a settled naming, which outranks silence.

Definition 6 (Observation). An observation is a tuple $o = (record_id, state, note, observed_on)$, where `observed_on` is an optional ISO date. Undated observations are accepted but cannot participate in staleness accounting.

Definition 7 (Finding). A finding is a frozen tuple $f = (record_id, kind, state, reasons, observed_on)$ carrying the assessed state together with the ordered trail of reasons that produced it.

Definition 8 (Report). A report is a tuple $(findings, intake_notes, review_date, registry_digest, intake_events)$. Its `tally` maps every state in S to a count, including states that occur zero times. The review date anchors staleness accounting; the registry digest identifies the record schema that was assessed; typed intake events preserve how input was handled.

Definition 9 (Intake event). An intake event is a frozen tuple $e = (code, message, position, record_id, state, displaced_state)$ recording one intake or reconciliation decision. The `code` ranges over the intake code alphabet the intake stage partitions; the `position` names the input position when one exists. For the two match-stage codes, `displaced_state` carries the asserted state of the observation the reconciliation replaced, so a conflict survives as two co-present typed surfaces rather than only as a note that a conflict occurred; it is empty for every set-aside and dating event, where nothing was displaced. An event is descriptive audit data, never a verdict on whoever supplied the input.

9.2 The staged evaluator

Definition 10 (Assessment map). The evaluator is the function

$$\text{assess_absence} : \text{Obs}^* \cup \{\perp\} \times R^* \times (\text{ISO} \cup \text{Date} \cup \{\perp\}) \longrightarrow \text{Report},$$

whose three parameters are `observations`, `records`, and `as_of`. The third resolves to a review date d (today when $\perp$), and an invalid date fails closed. Before intake, a custom registry must satisfy all 6 structural checks. The computation then runs in three stages: intake normalization, matching, then scoring.

Definition 11 (Intake stage). Intake II consumes the raw input in positional order and produces a map $A : id \rightarrow o$ of at most one accepted observation per record, together with typed events and one note per event. An input that cannot be iterated at all is refused whole, yielding an empty A and a single `NON_ITERABLE_INPUT` event. Otherwise an input at position i is set aside, with an event, when any of the following holds:

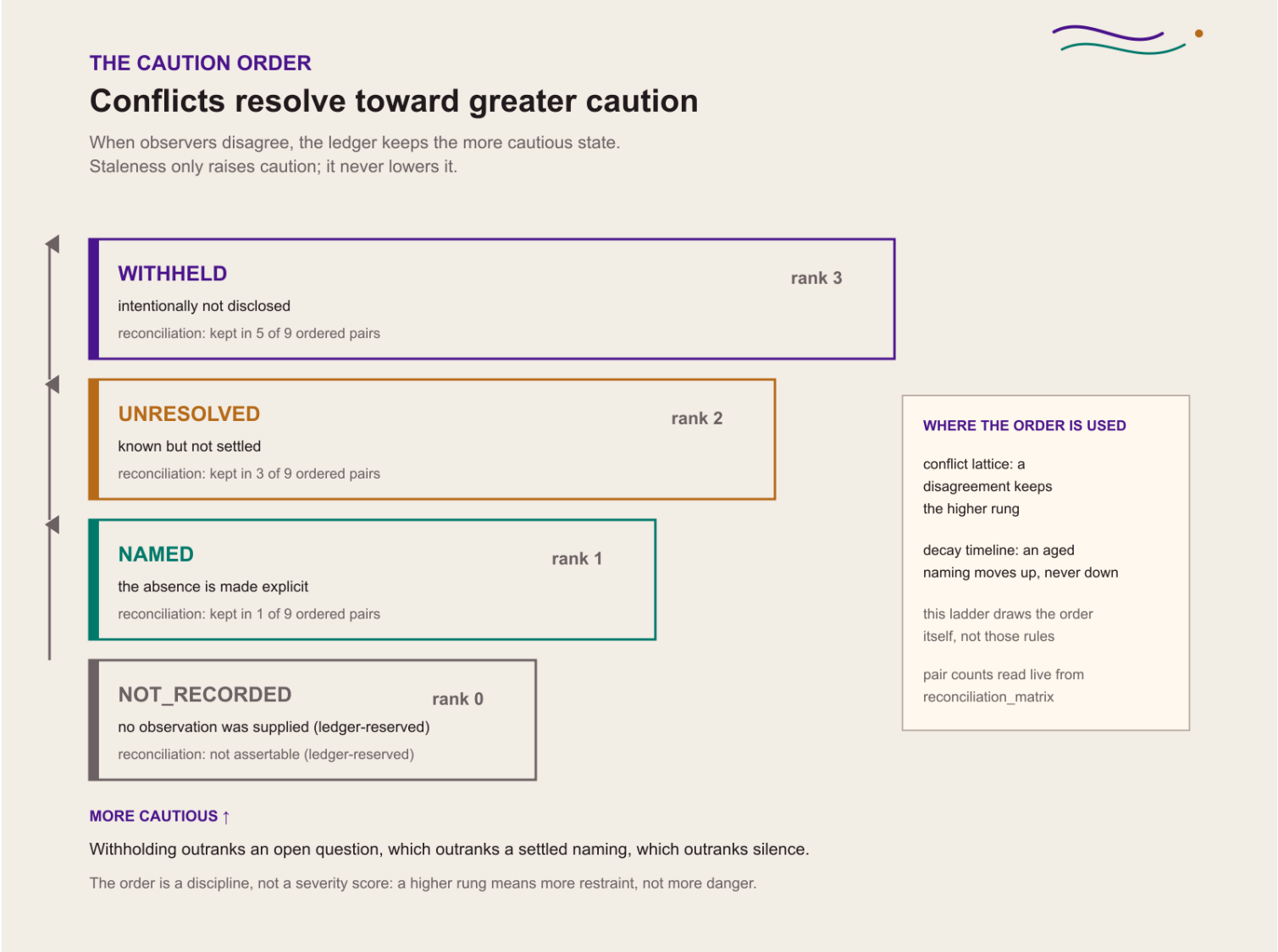


Figure 8: The defined caution order, read directly from `caution_rank`. When observers of one record disagree, the evaluator keeps the higher rung; staleness moves a naming from `NAMED` to `UNRESOLVED`, toward greater caution. Each rung also reports, live from `reconciliation_matrix`, in how many of the nine ordered pairs it is the kept state; the rules that use the order are drawn in the conflict-lattice and decay-timeline figures. A higher rung means more restraint, not more danger. The figure is a deterministic schematic generated from the state enumeration, not a severity score.

1. it is not an absence observation (`NON_OBSERVATION`);
2. its `record_id` is not the id of some record in R (`UNKNOWN_RECORD`);
3. its `state` is not a member of S (`UNTYPED_STATE`);
4. its `state` is `NOT_RECORDED`, the reserved meta-state (`RESERVED_STATE`).

An accepted observation whose `observed_on` is neither absent nor a string is kept and scored but emits `UNDATED_OBJECT`: acceptance is not dating. When two accepted observations name the same record with differing states, Π keeps $\arg \max_o c(o.state)$ — the more cautious — and records a `CONFLICTING_STATES` event. When states agree, the last-listed observation is kept and `DUPLICATE_STATE` records that replacement. The rule is positional, not chronological: it depends on input order, not on either observation’s date.

Intake therefore uses the whole code alphabet and partitions it: `NON_ITERABLE_INPUT`, `NON_OBSERVATION`, `UNKNOWN_RECORD`, `UNTYPED_STATE`, and `RESERVED_STATE` refuse an input; `UNDATED_OBJECT` keeps one whose date it cannot use; `CONFLICTING_STATE` and `DUPLICATE_STATE` reconcile two that were kept. No input is ever dropped without a note. When Π reconciles two accepted observations, the resulting event also carries the displaced observation’s asserted state as typed data ([Definition 9](#)), preserving the conflict as co-present surfaces; the cautious resolution is unchanged.

Definition 12 (Date resolution). Given `observed_on` and review date d , the resolution rule δ returns a pair (parsed-date-or- $\perp$, issue-note). Absent input — `None` or the empty string — yields $(\perp, \varepsilon)$: an undated observation is ordinary, and there is nothing to report. A non-string object yields $(\perp, \text{UNDATED_OBJECT})$: the observation is accepted and scored, but its date cannot age. An unparseable ISO string yields $(\perp, \text{“unreadable”})$, and a date t with $t > d$ yields $(\perp, \text{“after the review date”})$. Otherwise δ yields (t, ε) . A future date never counts as evidence.

δ has exactly one implementation, `white_line.dates.usable_observation_date`. Scoring calls it, and so does the `expiry_horizon` view when it reads a report back, so no view can age an observation the evaluator refused to age.

Definition 13 (Scoring rule). For a record r with horizon H , its accepted observation o (possibly absent), and review date d , the scoring rule φ reports:

1. **No observation** ($o = \perp$): state `NOT_RECORDED`, reason `no absence note was recorded`.
2. $o.state = \text{WITHHELD}$: state `WITHHELD`, reason `material is intentionally withheld`; and if the note is blank, the added reason `no boundary is stated for the withholding; restraint should name its duty`.
3. $o.state = \text{UNRESOLVED}$: state `UNRESOLVED`, reason `the absence is named but not resolved`.
4. $o.state = \text{NAMED}$ with $\delta = (t, \cdot)$ and $age = (d - t)$ in days: reason `the absence is explicitly named when the naming is current`, and otherwise the decay stated below.

$$\varphi_{\text{NAMED}}(r, o, d) = \begin{cases} \text{UNRESOLVED} & \text{if } t \neq \perp \text{ and } age > H \quad (\text{staleness decay}), \\ \text{NAMED} & \text{otherwise.} \end{cases}$$

Any date issue note from δ and any recorded note are appended to the finding’s reasons in that order.

Definition 14 (Review directive). The protocol maps each finding to one bounded action ρ without producing an approval state:

$$\begin{aligned} \rho(\text{NAMED}) &= \text{RETAIN_NAMED_SCOPE} \\ \rho(\text{UNRESOLVED}) &= \text{REOPEN_OR_BOUND} \\ \rho(\text{WITHHELD}) &= \text{HONOR_AND_CONFIRM_BOUNDARY} \\ \rho(\text{NOT_RECORDED}) &= \text{RECORD_OR_EXPLICITLY_SCOPE} \end{aligned}$$

The action is a prompt for human review, not a claim that the observation is true, safe, authorized, or complete.

9.3 Semantic guarantees

Each proposition below is a property of the maps just defined and is covered by the evaluator test suite.

Proposition 1 (Reserved meta-state). No observation can cause the report to assign `NOT_RECORDED` to a record. It is emitted only in the first branch of φ , for a record with no accepted observation. *Reason:* Π discards any observation whose state is `NOT_RECORDED` ([Definition 11](#), clause 4), so the state can never reach φ through an input.

Proposition 2 (Caution monotonicity). The state a record receives is never less cautious than the most cautious accepted observer of that record. Conflict resolution keeps the higher-caution state ([Definition 11](#)), and the only state rewrite in scoring is the staleness decay `NAMED` $\rightarrow$ `UNRESOLVED`, which raises caution from 1 to 2. The ledger therefore never resolves a disagreement, or an aging naming, by making an absence look *more settled* than it was reported.



THE STAGED ASSESSOR

Intake, then dating, matching, and scoring

Every set-aside input leaves a typed note; a conflict resolves toward the more cautious state; nothing is dropped silently.

1 · INTAKE — set aside what cannot be trusted (with a typed note)

NON_ITERABLE_INPUT	whole input is not iterable
NON_OBSERVATION	not an absence observation
UNKNOWN_RECORD	unknown record id
UNTYPED_STATE	untyped state
RESERVED_STATE	asserts the ledger-reserved state

2 · MATCH — keep at most one observation per record

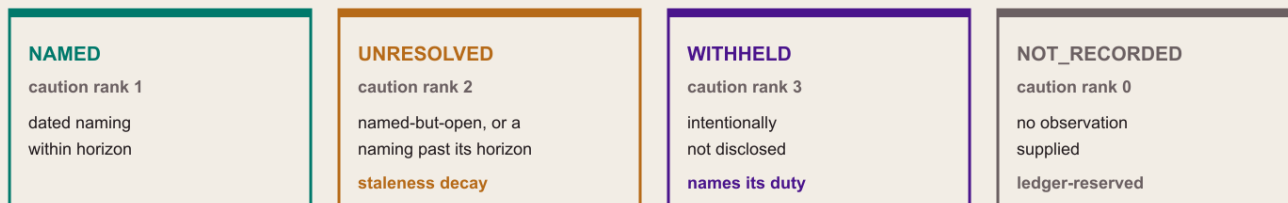
conflicting states for one record resolve toward the more cautious (higher caution rank)

typed match events: **CONFLICTING_STATES** keeps the more cautious state · **DUPLICATE_STATE** keeps the last-listed note

3 · DATE — retain ISO dates for staleness accounting

typed dating event: **UNDATED_OBJECT** keeps the observation; drops non-ISO dating

4 · SCORE — assign one ledger state per record



STALENESS DECAY — a dated **NAMED** past its review horizon reports **UNRESOLVED**

THE RULE UNDER THE DEFINED ORDER

No input can make an absence look more settled than its most cautious observer reported.

A dated naming older than its review horizon is reported **UNRESOLVED**, not **NAMED**.

The ledger records state; it never infers why something is absent.

Figure 9: The staged evaluator's decision path: intake sets aside untrusted input while leaving a typed note, matching keeps at most one — and the more cautious — observation per record, and scoring assigns exactly one of the four ledger states. The set-aside and match categories name every **IntakeCode** member, and the figure build fails if that enumeration changes, so the panels cannot silently drift from the intake rules. The labeled staleness arrow shows a dated **NAMED** naming decaying to **UNRESOLVED** once it ages past its horizon. The figure is generated from the evaluator contract, not from observed safety data; the reading rule beneath it is essential when the visual is read without the surrounding prose.



THE REASON TRAIL

Every branch, and what it wrote down

One input per branch of the scoring rule, each assessed live: 6 branches, 6 findings.

A reason explains one scoring decision; it is never a judgment of the observation.

COMPUTED THROUGH `assess_absence` · REVIEW DATE 2026-07-18 · A DECISION TRAIL, NOT EVIDENCE ABOUT ANY WORK

BRANCH AND SUPPLIED INPUT	THE ORDERED REASONS THE ASSESSOR RETURNED	
no observation supplied no observation for this record → NOT_RECORDED	1 no absence note was recorded	EPISTEMIC
withheld, boundary stated AbsenceObservation('withheld-material', WITHHELD) → WITHHELD	1 material is intentionally withheld 2 recorded note: Identifying details stay out of the publication.	ETHICAL
withheld, note blank AbsenceObservation('silenced-perspective', WITHHELD) → WITHHELD	1 material is intentionally withheld 2 no boundary is stated for the withholding; restraint should name its duty	ETHICAL
named but not resolved AbsenceObservation('unmeasured-uncertainty', UNRESOLVED) → UNRESOLVED	1 the absence is named but not resolved 2 recorded note: The variance is still being estimated.	EPISTEMIC
named, within horizon AbsenceObservation('negative-space', NAMED, '2026-06-30') → NAMED	1 the absence is explicitly named 2 observed 18 days before review, within the 365-day horizon 3 recorded note: Left open on purpose this cycle.	CONTEMPLATIVE
named, past horizon AbsenceObservation('missing-null-results', NAMED, '2025-09-01') → UNRESOLVED	1 the naming is 320 days old, past the 180-day review horizon; treated as unresolved until re-reviewed 2 recorded note: The dead ends were written up once.	EPISTEMIC
READING RULE The trail records why the ledger reported a state, and its order is the order the rule appended. A long trail is not a worse record than a short one; it is a record with more context attached.		

Figure 10: One row per branch of the scoring rule, each computed by a live `assess_absence` call and drawn with the ordered reason trail the evaluator returned. The two conditional reasons appear only in the branch that adds them: a blank-note withholding gains the duty reason, and a dated naming gains either its age line or the decay line. The plate quotes what the evaluator wrote, so a reworded reason moves the figure rather than leaving the prose behind. A reason is an explanation of one scoring decision, never a judgment of the observation or of whoever supplied it.

The reconciliation half of **Proposition 2** is small enough to exhibit exhaustively. The pure helper `reconciliation_matrix` in `white_line.analysis` returns the full ordered-pair resolution table over the three assertable states — 9 ordered pairs, each resolving to $\arg\max$ under c — and the test suite checks every cell against a real two-observer `assess_absence` call in both arrival orders. The table is symmetric with an identity diagonal, and `NOT_RECORDED` appears in no pair and no cell, because intake screens it out before reconciliation can see it.

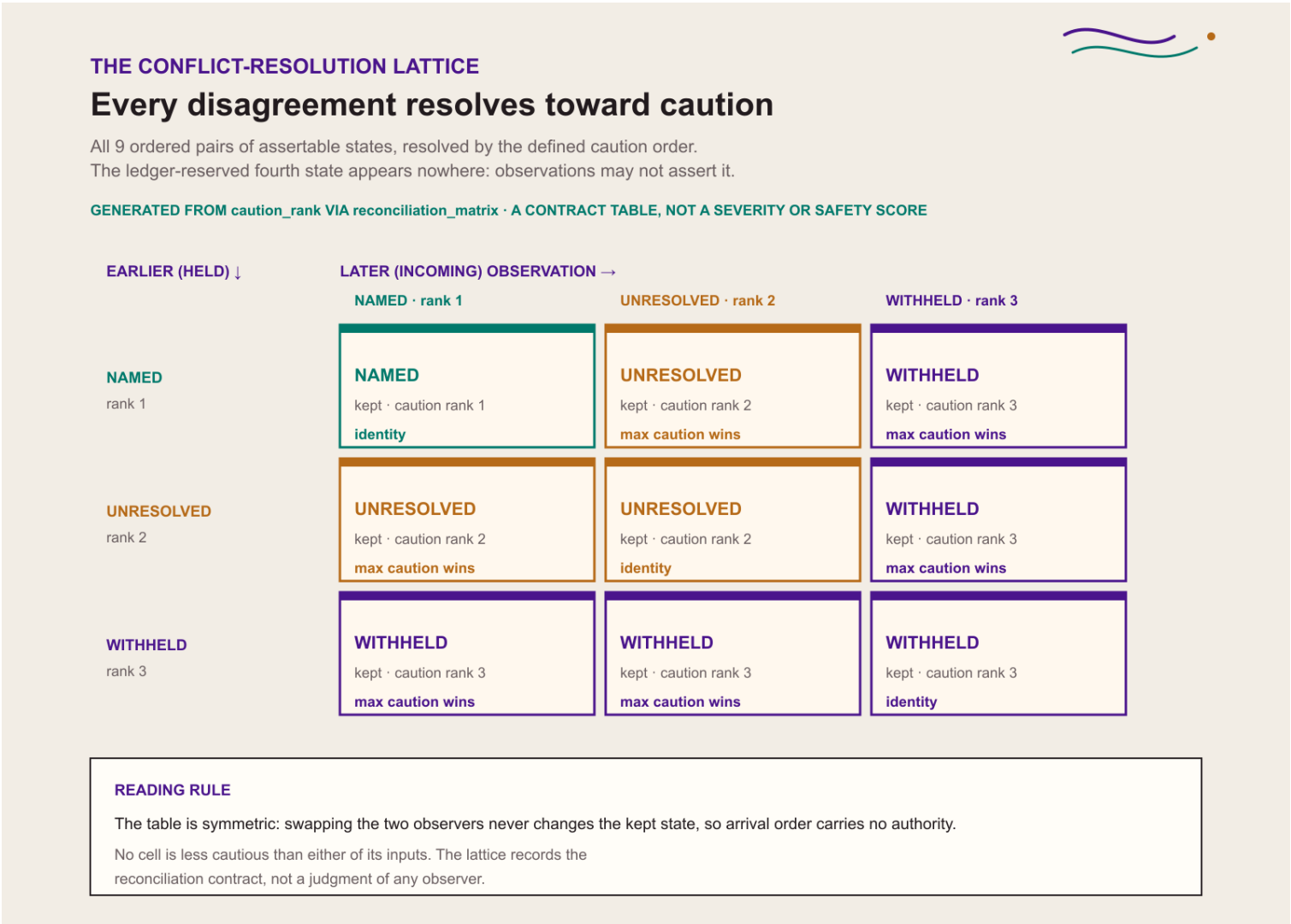


Figure 11: The full ordered-pair conflict-resolution table for assertable states, generated from `caution_rank` via `reconciliation_matrix`. Every cell keeps the more cautious input, the table is symmetric under swapping the observers — arrival order carries no authority — and the ledger-reserved `NOT_RECORDED` appears nowhere because observations may not assert it. The lattice records the reconciliation contract, not a severity score or a judgment of any observer.

Proposition 3 (Staleness). A dated `NAMED` observation whose age exceeds its record’s horizon is reported `UNRESOLVED`, never `NAMED` (**Definition 13**, clause 4). The old naming is no longer current evidence for a settled state.

Because the report is anchored by an explicit review date, **Proposition 3** can be made visible by holding one observation set fixed and sweeping the review date — the `decay_timeline` helper in `white_line.analysis` re-runs `assess_absence` at each date and tallies the resulting ledgers. For an illustrative cohort in which every record receives a `NAMED` observation dated on day 0, the swept ledger changes on exactly four days — offsets 91, 121, 181, and 366 — one day after each of the registry’s four horizon values (`{90, 120, 180, 365}` from **Definition 4**) expires. The `NAMED` count steps $11 \rightarrow 8 \rightarrow 6 \rightarrow 3 \rightarrow 0$ while `UNRESOLVED` gains the same records, and at every review date the four state counts still sum to 11 (**Proposition 5**). The sweep is a view of the contract applied to supplied observations; it is not observed data about any work, and a fully decayed cohort is not a finding that anything is wrong — only that the namings are due for re-review.

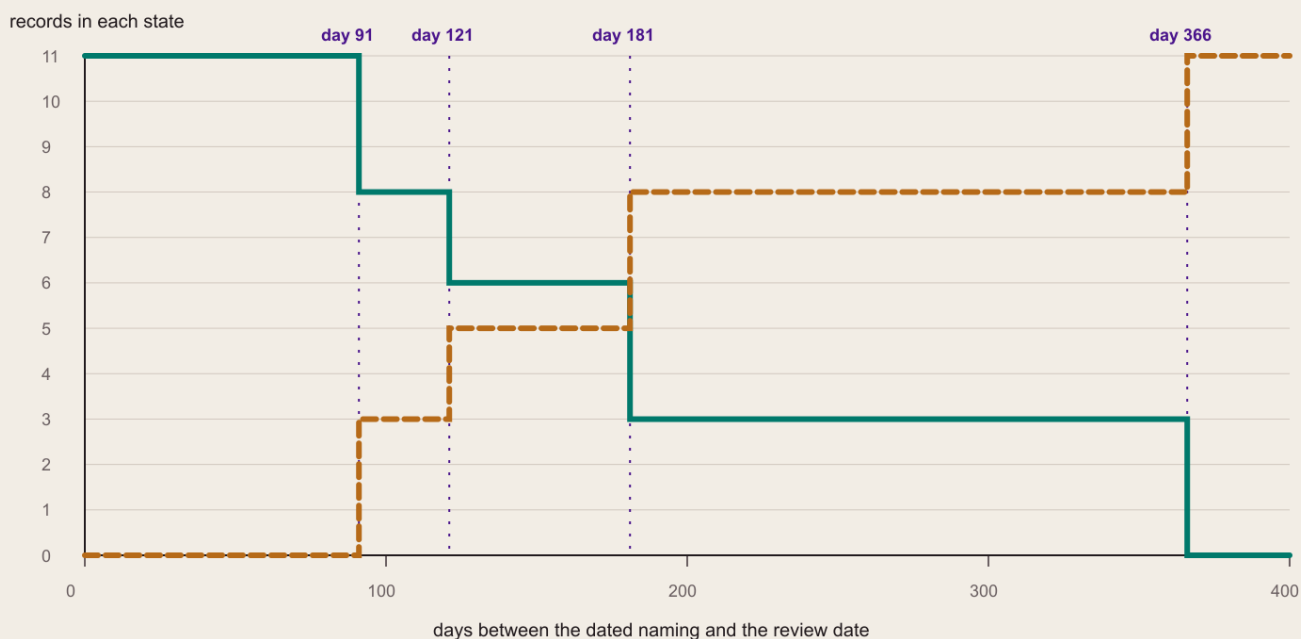
Proposition 4 (No silent discard). Every set-aside input, resolved conflict, and same-state duplicate contributes a typed event and a note to `intake_notes`. Discarding is therefore always visible in the report.

THE DECAY TIMELINE

A dated naming does not stay settled forever

All 11 records receive a NAMED observation dated 2026-01-01; the same input is re-assessed at every later review date.

COMPUTED BY SWEEPING `assess_absence` OVER REVIEW DATES · AN ILLUSTRATIVE COHORT, NOT OBSERVED PROJECT DATA



Each dashed crossing is one review horizon expiring: the naming is one day past its horizon and decays to UNRESOLVED.

Decay only raises caution; no review date makes an aged naming look more settled.
The curve describes the contract, not the world.

Figure 12: Staleness decay computed by sweeping `assess_absence` over review dates via `decay_timeline`: every registry record receives a NAMED observation dated 2026-01-01, and the same input is re-assessed at each later date. Each dashed crossing is one review horizon expiring — days 91, 121, 181, and 366, one day past the horizons 90, 120, 180, and 365 — moving records from NAMED to UNRESOLVED. Decay only raises caution; the curve describes the staleness contract applied to an illustrative cohort, not observed data about any work.

Proposition 5 (Totality). Every record in R yields exactly one finding, and `tally` accounts for all four states of S . The report is a total map over the current registry, not a complete account of the world or a filtered list of hits.

Proposition 6 (Context traceability). Every report produced by the evaluator carries the resolved review date and the canonical digest of the validated registry used to produce it. Two reports with different review dates or registry contents are therefore not silently interchangeable.

Proposition 7 (Bounded follow-up). `review_directives` maps every finding to exactly one action in [Definition 14](#) and never maps a state to `COMPLIANT`, `SAFE`, or `APPROVED`. The package describes a review boundary; it does not enforce a decision.

9.4 The witness layer

The caution order of [Definition 5](#) is a safe projection, not the whole state space: a single ordered value compresses how settled the record is, whether disclosure is being withheld, and how current the recorded observation is. The definitions below state that compression exactly — the facets it loses, the rule that recovers the shipped state from them, the obligation a review horizon reads forward to, and the envelope that makes a complete report transportable. Nothing in this layer changes what the evaluator records or emits; every object is derived from data a report already carries.

Definition 15 (Witness facets). The facet alphabets are the epistemic settlement $E = \{\text{NOT_RECORDED}, \text{NAMED}, \text{UNRESOLVED}, \text{UNDISCLOSED}\}$, the disclosure stance $D = \{\text{OPEN}, \text{WITHHELD}\}$, and the temporal status $T = \{\text{CURRENT}, \text{STALE}, \text{UNDATED}\}$, with $|E| = 4$, $|D| = 2$, and $|T| = 3$. A witness facet record is a frozen tuple $w = (\text{record_id}, \text{settlement}, \text{disclosure}, \text{temporality})$. `UNDISCLOSED` is the settlement behind a stated withholding — a statement about what the ledger can see, never a suspicion — and `STALE` describes the age of a recording under the record’s horizon, through the same date-resolution rule δ of [Definition 12](#), never the world the record points at. The facets describe recorded data only: `finding_facets` derives them from a finding the evaluator produced and fails closed on a finding this evaluator could not have written.

Definition 16 (Projection rule). The projection $\pi : E \times D \times T \rightarrow S$ is partial and restates the caution order as a projection of the facets:

$$\pi(w) = \begin{cases} \text{WITHHELD} & \text{if } w.\text{disclosure} = \text{WITHHELD}, \\ \perp \text{ (fails closed)} & \text{if } w.\text{settlement} = \text{UNDISCLOSED} \text{ otherwise,} \\ \text{NOT_RECORDED} & \text{if } w.\text{settlement} = \text{NOT_RECORDED}, \\ \text{UNRESOLVED} & \text{if } w.\text{settlement} = \text{UNRESOLVED}, \\ \text{UNRESOLVED} & \text{if } w.\text{temporality} = \text{STALE} \quad (\text{staleness decay}), \\ \text{NAMED} & \text{otherwise.} \end{cases}$$

Of the 24 combinations in $E \times D \times T$, 21 project and 3 fail closed: an `UNDISCLOSED` settlement without a `WITHHELD` stance is not a state this ledger can have recorded, and `project_facets` refuses it rather than guessing. The full table is pinned cell by cell in the test suite and drawn, with the fail-closed cells hatched, in the facets-lattice figure of the distributional-views section.

Proposition 8 (Factorization). For every finding f a live `assess_absence` call produces, with record r and review date d , $\pi(\text{finding_facets}(f, r, d)) = f.\text{state}$. In particular the staleness decay of [Definition 13](#) is recovered as a projection rather than a rewrite: a dated naming aged past its horizon factors as settlement `NAMED` with temporality `STALE` and projects to `UNRESOLVED`, so the prior naming and its age survive the decay as facets instead of being replaced by asserted doubt. The property is proven branch by branch over the live scoring battery, not against a restatement of the rules.

Definition 17 (Return contract). A return contract is a frozen tuple $(\text{record_id}, \text{state}, \text{action}, \text{trigger}, \text{due_on}, \text{expected_return}, \text{acceptance})$ one per finding, derived deterministically from the finding, its record, and the report’s own review date through the same δ of [Definition 12](#) the evaluator scored with. The action is the protocol’s own directive from [Definition 14](#); the trigger, expected return, and acceptance condition are the pinned module constants a live contract carries; and the review date and registry digest point back at the producing report, so a return answers an exact prior state rather than rewriting history. Fulfilling a contract earns a re-review, never a pass: an `UNRESOLVED` return may honestly leave the question open, and a `WITHHELD` contract’s return is boundary confirmation, never disclosure.

A review horizon, by itself, is a clock: it says when to look again and nothing else. A return contract is the same horizon read forward as an obligation — what must come back, what would count as an earned change, and which review state it returns to. For a current dated naming the two meet at one boundary: under the strict $\text{age} > H$ rule of [Definition 13](#), `due_on` is the observation date plus the horizon — the *last* review date on which the naming is still current, since a review one day later decays it. The binding test proves the boundary through the evaluator itself, assessing the same observation on `due_on` (still `NAMED`) and one day after (`UNRESOLVED`), rather than restating the arithmetic.

Definition 18 (Report envelope). The canonical report is a stable serialization of a complete report — findings, reasons, intake events, and notes — whose SHA-256 digest is the report’s `report_digest`. The report envelope is the tuple $(\text{schema_version}, \text{line_id}, \text{subject_id}, \text{review_date}, \text{registry_version}, \text{registry_digest}, \text{native_status}, \text{report_ref}, \text{source_snapshot_refs}, \text{source_report_refs})$ exported under the schema string `line.report-envelope/1.0`, where `report_ref` is that digest: the envelope points at the complete native report and never reinterprets it. `native_status` is deliberately typed in this line’s own vocabulary — the complete ordered per-record states, because this instrument has no single overall verdict — and the transportable non-claims ride inside, so a stored envelope cannot quietly outgrow what the instrument was allowed to say. Sibling instruments export the same shape by publishing the same schema string, never by importing one another, and envelopes from different lines must not be compared, ranked, averaged, or merged.

9.5 Structural invariants

Beyond any single assessment, six pure-compute checks validate the *shape* of the registry. Let $\text{ledger_sound}(R) = \bigwedge_{i=1}^6 P_i(R)$.

- P_1 — **distinct ids**. No two records share an id; a duplicate makes findings ambiguous and makes the digest order-dependent.
- P_2 — **canonical slug ids**. Every id matches the reviewed spelling $\sim[\text{a-z}][\text{a-z0-9}]*(-[\text{a-z0-9}]+)*\$$.
- P_3 — **inhabited fields**. `id`, `title`, `description`, and `prompt` are non-blank strings; a blank prompt gives a reviewer nothing to sit with.
- P_4 — **valid, covered kinds**. Every kind is a genuine member of K (a frozen dataclass does not type-check its fields), and all three kinds are present, so no whole kind of absence is silently lost.
- P_5 — **positive horizons**. Every horizon is a positive integer and not a boolean, so staleness arithmetic is well defined.
- P_6 — **stable serialization**. The canonical form builds, parses as JSON, and produces a digest independent of record order.

Proposition 9 (Defined-defect detection). Each invariant P_i is accompanied by a planted-defect registry: a registry carrying exactly the defect P_i guards against is constructed, and P_i fails on it while passing on the real registry R . A check that has never rejected a bad input is not counted as protection. This is the whole of the claim — the battery establishes detection of the planted defects, not the absence of every conceivable structural fault.

The battery is not only a test fixture. `invariant_defect_battery` in `white_line.invariants` constructs each defective registry, runs the whole check suite over it, and refuses to return unless the targeted check failed and every other check that the defect does not implicate still passed. The plate below draws what that call returned, so the claim above is visible rather than merely asserted.

9.6 Where each claim is checked

A claim in prose is not a checked claim. Every definition and proposition above names the test that exercises it, keyed by the block’s own label, and `tests/test_formalism.py` asserts that each row names a test that exists and that no block is missing a row — so a renamed test breaks the build rather than leaving a claim quietly unbound.

Claim	Test
Definition 1	<code>test_formalism.py::test_the_kind_alphabet_in_prose_is_the_enumeration</code>
Definition 2	<code>test_formalism.py::test_the_state_alphabet_in_prose_is_the_enumeration</code>
Definition 3	<code>test_formalism.py::test_every_tuple_definition_lists_its_dataclass_fields</code>
Definition 4	<code>test_manuscript_bindings.py::test_registry_definition_counts_re_derive_from_the_registry</code>
Definition 5	<code>test_formalism.py::test_the_caution_order_in_prose_is_caution_rank</code>
Definition 6	<code>test_formalism.py::test_every_tuple_definition_lists_its_dataclass_fields</code>
Definition 7	<code>test_formalism.py::test_every_tuple_definition_lists_its_dataclass_fields</code>
Definition 8	<code>test_formalism.py::test_every_tuple_definition_lists_its_dataclass_fields</code>
Definition 10	<code>test_formalism.py::test_the_assessment_map_matches_the_assessor_signature</code>
Definition 11	<code>test_formalism.py::test_the_intake_definition_partitions_the_live_code_alphabet</code>

Claim	Test
Definition 12	<code>test_formalism.py::test_every_date_resolution_branch_matches_the_shared_helper</code>
Definition 13	<code>test_formalism.py::test_every_quoted_scoring_reason_is_produced_by_the_assessor</code>
Definition 14	<code>test_formalism.py::test_the_directive_map_in_prose_is_the_protocol_map</code>
Proposition 1	<code>test_assessor.py::test_observations_cannot_assert_not_recorded</code>
Proposition 2	<code>test_analysis.py::test_reconciliation_matrix_agrees_with_the_assessor_in_both_orders</code>
Proposition 2 at higher arity	<code>test_assessor.py::test_three_or_more_observers_resolve_to_the_most_cautious</code>
Proposition 3	<code>test_assessor.py::test_stale_naming_decays_to_unresolved_past_the_horizon</code>
Proposition 3 swept over review dates	<code>test_analysis.py::test_decay_timeline_migrates_named_records_at_each_horizon_crossing</code>
Proposition 4	<code>test_analysis.py::test_events_tally_counts_real_intake_events_per_code</code>
Proposition 5	<code>test_assessor.py::test_report_covers_every_record_and_tally_counts_all_states</code>
Proposition 6	<code>test_serialization.py::test_digest_is_order_independent_and_repeatable</code>
Proposition 7	<code>test_protocol.py::test_review_protocol_maps_states_to_bounded_follow_up</code>
Proposition 9	<code>test_invariants.py::test_the_defect_battery_rejects_every_planted_defect</code>
Definition 9	<code>test_formalism.py::test_every_tuple_definition_lists_its_dataclass_fields</code>
Definition 15 alphabets	<code>test_formalism.py::test_the_facet_alphabets_in_prose_are_the_enumerations</code>
Definition 15 as a tuple	<code>test_formalism.py::test_every_tuple_definition_lists_its_dataclass_fields</code>
Definition 16 cell counts	<code>test_formalism.py::test_the_projection_cell_counts_re_derive_from_the_code</code>
Definition 16 full table	<code>test_witness_layer.py::test_the_projection_truth_table_is_total_except_the_unrecordable_cell</code>
Proposition 8	<code>test_formalism.py::test_the_factorization_proposition_holds_over_a_live_battery</code>
Definition 17 as a tuple	<code>test_formalism.py::test_every_tuple_definition_lists_its_dataclass_fields</code>
Definition 17 at the boundary	<code>test_witness_layer.py::test_the_due_date_is_the_last_current_day_proven_through_the_assessor</code>
Definition 18 as a tuple	<code>test_formalism.py::test_every_tuple_definition_lists_its_dataclass_fields</code>
Definition 18 points, never reinterprets	<code>test_witness_layer.py::test_the_envelope_points_at_the_native_report_without_reinterpreting_it</code>



THE INVARIANT BATTERY

Every check, actually rejecting

All 6 structural checks pass on the live registry and are then run over a registry carrying exactly the defect each one guards against.

DETECTION OF THE PLANTED DEFECTS · NOT A CLAIM THAT SIX CHECKS EXHAUST THE FAULTS A REGISTRY COULD CARRY

CHECK AND PLANTED DEFECT	WHAT THE CHECK RETURNED WHEN IT REFUSED
P1 · ids_are_distinct record 2 is an exact copy of record 1, so one id names two entries	live registry: ok · planted registry: rejected duplicate record ids: ["unobserved-dependency"]
P2 · ids_are_canonical_slugs record 1's id is written in the unreviewed spelling 'Unobserved_Dependency'	live registry: ok · planted registry: rejected non-canonical record ids: ["Unobserved_Dependency"]
P3 · fields_are_inhabited record 1's reviewer prompt is blank, so a reviewer is given nothing to sit with	live registry: ok · planted registry: rejected blank or non-string fields: ["unobserved-dependency'.prompt"]
P4 · kinds_are_valid_and_covered record 1's kind is the bare string 'EPISTEMIC', which a frozen dataclass accepts	live registry: ok · planted registry: rejected untyped kinds on: ["unobserved-dependency"]; uncovered kinds: [] the same registry also trips: serialization_is_stable
P5 · horizons_are_positive record 1's review horizon is 0 days, which makes every dated naming instantly stale	live registry: ok · planted registry: rejected non-positive or non-integer horizons: ["unobserved-dependency'=0"]
P6 · serialization_is_stable records 1 and 2 share an id but differ in content, so the digest follows input order	live registry: ok · planted registry: rejected digest depends on record order the same registry also trips: ids_are_distinct

READING RULE

A row shows that this check refuses this defect, which is narrower than saying the registry is sound.

A defect no check was written for would leave every row here unchanged.

Figure 13: One row per structural invariant, computed by running the whole check battery over the live registry and then over a registry carrying exactly the defect that invariant guards against. Each row shows the check's verdict on the real registry and the detail string it returned when it rejected the planted defect, so a check that stopped rejecting would empty its own row rather than pass quietly. The plate demonstrates detection of the planted defects; it is not a claim that the six checks exhaust the structural faults a registry could carry.

10 Worked ledger entries

An absence can be recorded directly:

```
from white_line import AbsenceObservation, AbsenceState, assess_absence

report = assess_absence([
    AbsenceObservation(
        "withheld-material",
        AbsenceState.WITHHELD,
        "The identifying details remain outside the publication.",
    ),
])
```

The result distinguishes `WITHHELD` from `NOT_RECORDED`. A record for an unobserved dependency can remain `UNRESOLVED` while investigation continues, and a named contemplative space can remain `NAMED` without implying that an unseen agent, field, or force exists.

The staging shows itself under adversarial or careless input. Suppose two reviewers disagree about the same record — one calls it `NAMED`, another `UNRESOLVED`. The ledger keeps `UNRESOLVED`, the more cautious of the two, and writes a note explaining that it did so; it never quietly upgrades an open question into a settled naming. Suppose an input asserts `NOT_RECORDED` directly: intake sets it aside, because that state is reserved for records no one observed, and again leaves a note rather than absorbing the claim.

Staleness is dated, not assumed. A `NAMED` observation carrying an `observed_on` date is compared against the review date:

```
report = assess_absence(
    [AbsenceObservation("unobserved-dependency", AbsenceState.NAMED,
        "Checked the upstream feed.", observed_on="2025-01-01")],
    as_of="2026-07-18",
)
```

Because the naming is far older than the record's ninety-day horizon, it is reported `UNRESOLVED` with a reason that states its age and horizon. Nothing about the record has been discovered to be wrong; the ledger has simply stopped treating an eighteen-month-old check as though it were made this morning.

The note is contextual evidence for the ledger, not a license to expose what was withheld. In a real project, access controls and consent obligations live outside this small pure-data instrument.

The state-to-action boundary is explicit rather than implied:

```
from white_line import review_directives

for directive in review_directives(report):
    if directive.requires_follow_up:
        print(directive.record_id, directive.action.value)
```

This emits bounded follow-up such as `REOPEN_OR_BOUND` or `HONOR_AND_CONFIRM_BOUNDARY`; it never emits permission or a safety verdict. The report also carries `review_date`, `registry_digest`, and typed `intake_events`, which should travel with the human review record.

11 Reading one report distributionally

The worked entries above read a report record by record. A reviewer auditing a whole ledger usually needs the complementary view: *how was the input handled in aggregate, and where do the states currently sit across the three kinds of absence?* The `white_line.analysis` module answers with three distributional summaries, each computed either through the public `assess_absence` call or directly from the defined caution order: a per-code tally of intake events (`events_tally`), a kind-by-state cross-tabulation of findings (`coverage_matrix`), and the review-date sweep (`decay_timeline`) presented alongside [Proposition 3](#) in the formal-method section. All three are pure and additive. Nothing in the module changes what the ledger records or how a state is assigned; the summaries only make the existing contract inspectable. They are descriptive displays in Tukey’s sense — arrangements of what happened, offered before and instead of inference [[Tukey, 1977](#)] — not scores of any kind.

11.1 A worked intake under careless and adversarial input

The intake stage is easiest to see when the input deliberately exercises its screening rules. The following seven inputs — assessed against the live registry with `as_of="2026-07-18"` — include a two-observer conflict, a same-state duplicate, a typo’d record id, an attempt to assert the reserved state, and one object that is not an observation at all:

```
from white_line import (
    coverage_matrix,
    events_tally,
    worked_intake_observations,
    worked_report,
)
```

```
observations = worked_intake_observations()
report = worked_report()
```

`events_tally(report)` returns a count for **every** intake code, including the codes that did not fire — a zero is reported, never omitted. For this input the distribution is:

Intake code	Count	What happened here
NON_ITERABLE_INPUT	0	The input container itself was iterable.
NON_OBSERVATION	1	The bare string was set aside; it is not an observation.
UNKNOWN_RECORD	1	no-such-record names no registry id.
UNTYPED_STATE	0	Every remaining state was a genuine <code>AbsenceState</code> member.
RESERVED_STATE	1	The asserted <code>NOT_RECORDED</code> was screened out (Proposition 1).
UNDATED_OBJECT	0	Every dated observation carried a parseable ISO string.
CONFLICTING_STATES	1	The <code>NAMED/UNRESOLVED</code> disagreement kept <code>UNRESOLVED</code> .
DUPLICATE_STATE	1	The agreeing <code>WITHHELD</code> pair kept the last-listed note.

Eight rows for eight codes, five events, and exactly five matching intake notes — [Proposition 4](#) made countable. The conflict row is [Proposition 2](#) in one line: the two observers of `unobserved-dependency` disagreed, and the record’s reported state is `UNRESOLVED`, the more cautious of the pair, exactly as the conflict-lattice figure in the formal-method section tabulates for all nine ordered pairs. The resulting state tally is `NAMED 0`, `UNRESOLVED 1`, `WITHHELD 1`, and `NOT_RECORDED 9`: eleven findings, one per record, as [Proposition 5](#) requires.

An important reading rule follows directly from the table: an event count is audit data about *input handling*, never a verdict on whoever supplied the input. A ledger whose event tally is all zeros has clean input mechanics; it has not thereby earned trust for its content. And a nonzero `UNKNOWN_RECORD` row may mean a typo, a stale registry checkout, or an extension that was never registered — the code names what the intake stage did, not why the mismatch exists.

11.2 A zero row is untriggered, not unreachable

Three of the eight rows above are zero, which leaves a reader with a question the table cannot answer: is the code merely untriggered by this input, or is it dead code that nothing can reach? A reporting convention that prints zeros honestly is worth very little if some of those zeros can never become anything else. The battery below settles it. Each row supplies one input constructed to reach exactly one code, runs it through `assess_absence`, and prints the input beside the typed note the evaluator returned. Every member of the enumeration fires, so the zeros in the worked tally mean untriggered here.

The battery is a demonstration of reachability and nothing more. It shows that each code *can* be produced, not that any real review would produce it, and certainly not that a review producing it has gone wrong. `NON_ITERABLE_INPUT` appears only when



THE INTAKE BATTERY

Every intake code, actually reached

One constructed input per code, run through `assess_absence`: all 8 of the live `IntakeCode` members fire.
A zero in the worked tally therefore means untriggered here, never unreachable.

A CONSTRUCTED BATTERY · REVIEW DATE 2026-07-18 · NOT OBSERVED INPUT AND NOT A VERDICT ON ANY SUPPLIER

CODE, TRIGGERING INPUT, AND THE TYPED NOTE PRODUCED

NON_ITERABLE_INPUT

input: 42

typed note: observations is not iterable and was set aside

NON_OBSERVATION

input: 'not an observation at all'

typed note: input 0 is not an absence observation and was set aside

UNKNOWN_RECORD

input: `AbsenceObservation('no-such-record', NAMED)`

typed note: input 0 names an unknown record id and was set aside

UNTYPED_STATE

input: `AbsenceObservation('unasked-question', 'NAMED')`

typed note: input 0 (unasked-question) carries an untyped state and was set aside

RESERVED_STATE

input: `AbsenceObservation('negative-space', NOT_RECORDED)`

typed note: input 0 (negative-space) asserts `NOT_RECORDED`; that state is reserved for the ledger itself and was set aside

UNDATED_OBJECT

input: `AbsenceObservation('fallow-ground', NAMED, date(2026, 1, 5))`

typed note: input 0 (fallow-ground) has an observation date object that is not an ISO date string; the observation was accepted at intake and its date cannot participate in staleness accounting

CONFLICTING_STATES

input: `AbsenceObservation('missing-null-results', NAMED) + AbsenceObservation('missing-null-results', WITHHELD)`

typed note: conflicting states for missing-null-results; kept the more cautious `WITHHELD`

DUPLICATE_STATE

input: `AbsenceObservation('silenced-perspective', WITHHELD) + AbsenceObservation('silenced-perspective', WITHHELD)`

typed note: duplicate `WITHHELD` observations for silenced-perspective; kept the last-listed

READING RULE

This plate establishes reachability of each code, not that any real input would ever produce it.

A typed note describes what intake did with an input; it is never a judgment of whoever supplied an input.

Figure 14: A positive control for the intake contract: one constructed input per `IntakeCode`, each run through `assess_absence`, drawn as one row with the input quoted beside the typed note the evaluator produced for it. Every member of the live enumeration fires, so a zero in the worked tally above means the code was not triggered by that input rather than that the code is unreachable. The battery is built to exercise the contract; it is not observed input, and no note is a verdict on whoever supplied an input.

the whole input, rather than one of its members, cannot be iterated, which is why the battery supplies whole inputs rather than a single list.

11.3 Where the states sit across the three kinds

`coverage_matrix(report)` cross-tabulates the same eleven findings by kind and state, with every cell present even at zero. For the worked input above:

Kind	NAMED	UNRESOLVED	WITHHELD	NOT_RECORDED	Row sum
EPISTEMIC	0	1	0	3	4
ETHICAL	0	0	1	3	4
CONTEMPLATIVE	0	0	0	3	3
Column sum	0	1	1	9	11

The margins are not decoration; they are the consistency check. Row sums (4, 4, 3) equal the registry’s partition by kind from [Definition 4](#), and column sums equal the report’s own **tally** — the test suite asserts both identities against real assessments. The same helper applied to an empty-input assessment gives the baseline every review starts from: all eleven records NOT_RECORDED, distributed 4/4/3 down the final column, every other cell zero.

What the matrix supports is a bounded reviewer question, asked kind by kind: *the ethical row currently holds one WITHHELD and three NOT_RECORDED records — have those three simply not been reviewed this cycle, or is the review itself avoiding them?* The matrix cannot answer that question, and it is not meant to. A full NOT_RECORDED column is not negligence, a full NAMED row is not diligence, and no cell count is a completeness or coverage *score* of any work. The numbers describe records in a ledger; the judgment about what the distribution means belongs to the humans conducting the review, with the report’s **review_date** and **registry_digest** attached so a later reviewer can tell which registry and which day the distribution describes ([Proposition 6](#)).

The third view — sweeping the review date while holding the observations fixed — appears with the staleness proposition in the formal-method section, where its four transition days (91, 121, 181, 366) are read off the registry’s horizon multiset. The three views cover the report’s three axes of variation: how input became findings, how findings distribute over the typology, and how a fixed input’s findings move as the review date advances. None of them adds information the report did not already carry, which is exactly what makes them safe to automate.

11.4 What comes due next, and what cannot come due at all

A reviewer holding a report also needs a prospective view: of the namings recorded today, which one reaches its review boundary first? `expiry_horizon(report)` answers that, returning one entry per current dated NAMED finding, ordered nearest-boundary-first, each carrying the record’s horizon, the naming’s age, and the days that remain. Its omissions carry as much information as the queue. A naming the evaluator could not date cannot expire, so it never enters — not because it is fresh, but because staleness accounting has nothing to measure against. A naming already past its horizon is reported UNRESOLVED rather than NAMED, so it has left by decaying. Both are drawn as labelled bands beneath the queue rather than dropped, because a silently absent row would be a strange failure in a work about absence, and the undated naming is the one a reviewer is most likely to assume is current.

The queue ranks by days remaining, not by weight. A record near the top is one the ledger will ask about soon, which is a statement about the cadence its horizon declares and about nothing else. Two records with identical remaining days sort by id, so the order is total and the figure is reproducible.

11.5 The state as a safe projection: the witness layer

Every view above reads the single caution-ordered state, and that state is a deliberate compression. Strong support and strong resistance co-present in a record are not the same situation as no evidence at all, yet a lone ordered value cannot say so. The witness layer ([Definition 15](#) through [Definition 18](#) in the formal-method section) makes the compression inspectable without widening what the evaluator emits: **finding_facets** factors each finding into its epistemic settlement, disclosure stance, and temporal status; **project_facets** states the rule that recovers the shipped state from them; **return_contracts** reads each record’s review horizon forward as an obligation with a due date proven at the decay boundary; and **report_envelope** wraps the complete report for co-registration beside the other lines, pointing at the canonical report by digest and never reinterpreting it. A conflict likewise survives reconciliation as typed data: the intake event keeps the displaced observation’s asserted state, so the more cautious resolution stands without erasing what it displaced.

The lattice below draws the whole projection at once. Each cell is one facet combination and carries the state **project_facets** returns for it; the three cells for an undisclosed settlement without a stated withholding fail closed, because that is not a state this



TWO DISTRIBUTIONAL VIEWS

How input was handled, and where the states sit

The worked seven-input example assessed against the live registry: 5 intake events became 11 findings, one per record.

COMPUTED THROUGH `assess_absence` · REVIEW DATE 2026-07-18 · COUNTS DESCRIBE INPUT HANDLING, NOT ANY PERSON OR WORK

INTAKE EVENTS BY CODE — `events_tally`

every code is reported, including zeros

NON_ITERABLE_INPUT	0
NON_OBSERVATION	1
UNKNOWN_RECORD	1
UNTYPED_STATE	0
RESERVED_STATE	1
UNDATED_OBJECT	0
CONFLICTING_STATES	1
DUPLICATE_STATE	1

FINDINGS BY KIND × STATE — `coverage_matrix`

row sums = registry partition · column sums = report tally

	NAMED	UNRESOLVED	WITHHELD	NOT_RECORDED	row sum
EPISTEMIC	0	1	0	3	4
ETHICAL	0	0	1	3	4
CONTEMPLATIVE	0	0	0	3	3
column sum	0	1	1	9	11

READING RULE

An event count is audit data about input handling, never a verdict on whoever supplied the input.

No cell is a completeness or coverage score; a zero is an absence of the event, not evidence about the world.

Figure 15: The two distributional views of the worked seven-input example, computed live through `assess_absence`, `events_tally`, and `coverage_matrix` at review date 2026-07-18: intake-event counts per code in the upper panel, with zeros reported rather than omitted, and the kind-by-state coverage matrix in the lower panel, with row sums matching the registry's partition by kind and column sums matching the report's own tally. The figure draws the same computation as the two tables above, over the same eight intake codes. Neither an event count nor a cell count is a completeness score, a safety score, or a verdict on whoever supplied the input.



THE REFRESH QUEUE

Which naming comes due first

expiry_horizon over one assessed report: 4 current dated namings, ranked nearest-boundary-first.
A queue position is a review cadence, not a measure of importance or risk.

COMPUTED THROUGH `assess_absence` · REVIEW DATE 2026-07-18 · A REFRESH SCHEDULE, NOT A COMPLETENESS OR SAFETY SCORE

RECORD	AGE AND REMAINING DAYS AGAINST THE RECORD'S HORIZON	
Unobserved dependency <code>EPISTEMIC</code> · observed 2026-05-20	 59 days old of 90 · 31 days remaining	#1
Withheld material <code>ETHICAL</code> · observed 2026-03-11	 129 days old of 180 · 51 days remaining	#2
Unmeasured uncertainty <code>EPISTEMIC</code> · observed 2026-07-02	 16 days old of 120 · 104 days remaining	#3
Contemplative negative space <code>CONTEMPLATIVE</code> · observed 2026-06-30	 18 days old of 365 · 347 days remaining	#4

UNDATED NAMING — cannot expire, so it never enters the queue

1 in this input: `Uncredited labor`

ALREADY DECAYED — past its horizon, so the ledger reports UNRESOLVED

1 in this input: `Unasked question`

READING RULE

The two bands are drawn, not dropped: an omitted naming is still a naming a reviewer supplied.

A short remaining count means the naming is due for re-review, never that the underlying work is wrong.

Figure 16: The refresh queue computed by `expiry_horizon` over one assessed report: every current dated **NAMED** finding, ranked nearest-boundary-first, with its kind shown by colour and by marker shape, its age drawn as a filled portion of its own review horizon, and its remaining days stated in words. The helper's two documented omissions are drawn as labelled bands rather than dropped: a naming the evaluator could not date cannot expire, and a naming already past its horizon is reported **UNRESOLVED**. Queue position is a review cadence, not importance, risk, or a completeness score.

ledger can have recorded; and the marked cells are where the live scoring battery's findings land when each is factored through `finding_facets` at build time. The one cell in which a `NAMED` settlement projects to `UNRESOLVED` is the staleness decay, recovered here as a projection rather than a rewrite — the prior naming and its age survive as facets ([Proposition 8](#)).

None of this ranks anything. The facets are derived views of recorded data, never new observations; a return contract earns a re-review, never a pass; and envelopes from different lines must not be compared, averaged, or merged — the envelope exists so that a separate register, if one is ever built, can co-register complete reports without any line reinterpreting another.



THE FACETS LATTICE

Every combination, and what it projects to

project_facets over all 24 settlement × stance × temporality combinations: 21 project, 3 fail closed.
Marked cells are inhabited by the live scoring battery, factored through finding_facets at build time.

COMPUTED THROUGH project_facets AND finding_facets · REVIEW DATE 2026-07-18 · RECORDED-DATA FACETS, NOT A RICHER WORLD

SETTLEMENT AND STANCE	CURRENT	STALE	UNDATED
NOT_RECORDED stance OPEN	NOT_RECORDED	NOT_RECORDED	NOT_RECORDED ● live battery ×1
NOT_RECORDED stance WITHHELD	WITHHELD	WITHHELD	WITHHELD
NAMED stance OPEN	NAMED ● live battery ×1	UNRESOLVED the staleness decay ● live battery ×1	NAMED
NAMED stance WITHHELD	WITHHELD	WITHHELD	WITHHELD
UNRESOLVED stance OPEN	UNRESOLVED	UNRESOLVED	UNRESOLVED ● live battery ×1
UNRESOLVED stance WITHHELD	WITHHELD	WITHHELD	WITHHELD
UNDISCLOSED stance OPEN	FAILS CLOSED not recordable	FAILS CLOSED not recordable	FAILS CLOSED not recordable
UNDISCLOSED stance WITHHELD	WITHHELD	WITHHELD	WITHHELD ● live battery ×2

READING RULE

A cell names the one caution-ordered state the ledger reports for records whose facets land there.

The lattice is a projection table of recorded-data facets; it is not a claim that richer states exist in the world.

Figure 17: The full facet lattice computed by running `project_facets` over every settlement, disclosure-stance, and temporal-status combination: each cell is coloured and labelled by the single caution-ordered state the projection returns, the cells for an `UNDISCLOSED` settlement without a stated withholding are drawn hatched because that combination fails closed rather than projecting, and the cells inhabited by the live scoring battery are marked by factoring each of its findings through `finding_facets` at build time. The one cell where a `NAMED` settlement projects to `UNRESOLVED` is the staleness decay recovered as a projection. The lattice is a projection table of recorded-data facets; it is not a claim that richer states exist in the world, and an inhabited cell is a fact about the battery's constructed inputs, never about any observed work.

12 Limits and safeguards

White Line is unusually easy to misuse, because people project meaning into gaps. A blank space invites a story, and the most dangerous story reads absence as confirmation. The scope is narrow on purpose:

- **NOT_RECORDED** means only that no observation was supplied to this function. It does not mean absent, false, secret, or irrelevant.
- **UNRESOLVED** is not proof that a hidden cause exists. It is how the ledger holds a question open, including after a naming has gone stale.
- **WITHHELD** must never be reverse-engineered into disclosure. The accompanying note is context for review, not a key to the material.
- **NAMED** does not make an absence harmless, complete, or spiritually significant.
- A record may describe an unrepresented perspective without speaking for it. Naming a gap is not filling it.

The registry’s finiteness sits behind all of these. Eleven records name eleven categories a reviewer is prompted about; absences outside that list are not tracked, not flagged, and not counted as missing. Taleb’s point about consequential events falling outside an anticipated possibility space applies directly to a fixed category list [Taleb, 2007], and it cuts against this instrument rather than for it. A full ledger with every record reviewed and every horizon current is evidence about eleven questions, not about the ones no one thought to add. Extending the registry is the only remedy, and an extension is a human judgment the package cannot make.

The machinery has its own boundaries. The caution order resolves disagreement; it is not a severity score, and a higher rung means more restraint was reported rather than that more danger is present. The registry digest lets a later checkout be compared against a stored manifest; it carries no security or safety semantics and proves nothing about authorship or intent. The staleness horizon is a review cadence, not a truth about when knowledge expires. The six structural invariants check the registry’s shape, and the shipped defect battery shows each one refusing the defect it was written for — which establishes detection of that defect and nothing about faults no check was written for.

None of this replaces security review, consent processes, source criticism, accessibility work, or consultation with affected people. Red Line sets explicit prohibitions and White Line cannot weaken them. Golden Line’s aspirations cannot fill an absence, and Black Line’s evidence discipline cannot turn non-disclosure into data. White Line’s only promise is to keep the negative space legible so the other lines cannot silently paint over it.

The distributional views in `white_line.analysis` inherit every boundary above and add one. A cross-tabulation or an event tally rearranges a report; it is not new evidence, and the rearrangement invites a specific misreading — a filled **NAMED** row as diligence, a filled **NOT_RECORDED** column as negligence, a zero event count as trustworthy input. None of those readings is supported. The decay sweep is the same contract applied at many review dates, so a fully decayed cohort means the namings are due for re-review, not that anything was discovered to be wrong. Anyone ranking teams, projects, or people by these counts has left the instrument’s scope entirely.

The refresh queue is the sharpest case, because it looks forward. It orders by days remaining under a declared horizon, which is a statement about cadence and not about importance, risk, or how fast the underlying situation moves. It is also not a picture of everything due: a naming the evaluator could not date cannot expire and never enters the queue, and a naming already past its horizon has left by decaying. Both omissions are drawn as labelled bands rather than dropped, because in a work about absence a silently missing row is the worst failure available, and the undated naming is exactly the one a reader assumes is current. The two positive-control batteries carry the mirror-image limit: each is constructed so that every code, and every check, fires. They establish reachability and detection, and say nothing about how often a real review would reach one.

Reproducibility has the same shape. A matching registry digest and a fresh rendered bundle show that the local artifacts agree with one another; they do not turn local notes into independent evidence. The source-to-artifact audit is a release-consistency gate, not a research-validity certificate.

13 Conclusion

White Line makes one modest promise: a careful work can say what it does not know, what it will not disclose, and what it chooses not to close with a claim. That is a practice of restraint rather than an aesthetic of vagueness. The difference is that this restraint is typed, dated, and reviewable — an absence is named, given a kind, and assigned a state the machinery is forbidden from quietly making more settled than it was reported.

Typed absence keeps three duties apart. Epistemic gaps call for humility and further inquiry; the ledger records them and lets a stale naming lapse back into an open question rather than pretending it still holds. Ethical restraint protects people and boundaries; the ledger can hold a withholding or an unrepresented perspective without speaking for what it declines to expose. Contemplative space lets an account stay open without inventing an invisible cause. The instrument's strongest honest output is often a `NOT_RECORDED`, an `UNRESOLVED`, or a note explaining what it set aside.

That is White Line's place in the set. Red Line refuses, Black Line builds, and Golden Line reaches; White Line marks the edge where knowledge, obligation, and language run out, and keeps that edge from being painted over. The discipline is old — the Socratic disavowal, the apophatic traditions, the right to opacity (see [the scholarship section](#)) — and my contribution is only a small, explicit, testable place for absence to remain itself. The report date, registry digest, typed intake events, and bounded review directives make that restraint inspectable. None of them makes it enforceable, and I would distrust a version of this work that claimed otherwise.

The project is indexed in my public research graph ([github docxology/docxology](#)); its repository is `docxology/white_line`.

References

- Geoffrey C. Bowker and Susan Leigh Star. *Sorting Things Out: Classification and Its Consequences*. The MIT Press, 1999. ISBN 9780262024617. URL <https://mitpress.mit.edu/9780262024617/sorting-things-out/>. Analysis of classification and standards as consequential information infrastructures. Accessed 2026-07-18.
- John Cage. *Silence: Lectures and Writings*. Wesleyan University Press, 1961. Includes Cage’s account of the silent composition 4’33” and structured negative space. Accessed 2026-07-18.
- Catherine D’Ignazio and Lauren F. Klein. *Data Feminism*. The MIT Press, 2020. ISBN 9780262044004. URL <https://mitpress.mit.edu/9780262044004/data-feminism/>. Intersectional account of data science, power, classification, and data ethics. Accessed 2026-07-18.
- Kristie Dotson. Tracking epistemic violence, tracking practices of silencing. *Hypatia*, 26(2):236–257, 2011. doi: 10.1111/j.1527-2001.2011.01177.x. URL <https://doi.org/10.1111/j.1527-2001.2011.01177.x>. Account of epistemic violence as a failure to meet speaker vulnerability under pernicious ignorance. Accessed 2026-07-18.
- W. E. B. Du Bois. *The Souls of Black Folk*. A. C. McClurg & Co., 1903. The “veil” names a perspective made structurally unseen rather than merely absent. Accessed 2026-07-18.
- Miranda Fricker. *Epistemic Injustice: Power and the Ethics of Knowing*. Oxford University Press, 2007. doi: 10.1093/acprof:oso/9780198237907.001.0001. URL <https://academic.oup.com/book/32817>. Distinguishes testimonial and hermeneutical injustice as wrongs done to people in their capacity as knowers. Accessed 2026-07-18.
- Édouard Glissant. *Poetics of Relation*. University of Michigan Press, 1997. French original 1990; develops the “right to opacity” of the other. Accessed 2026-07-18.
- Matthias Gross and Linsey McGoey, editors. *Routledge International Handbook of Ignorance Studies*. Routledge, London, 2015. ISBN 9780415718967. Edited survey of ignorance studies across its philosophical, sociological, and policy registers. Accessed 2026-07-28.
- Donna Haraway. Situated knowledges: The science question in feminism and the privilege of partial perspective. *Feminist Studies*, 14(3):575–599, 1988. URL <https://www.jstor.org/stable/3178066>. Account of partial, located knowledge and a non-transcendent conception of objectivity. Accessed 2026-07-18.
- Sandra Harding. Rethinking standpoint epistemology: What is “strong objectivity”? *The Centennial Review*, 36(3):437–470, 1992. Standpoint account of objectivity strengthened by examining the social locations and power relations shaping inquiry. Accessed 2026-07-18.
- John Keats. *The Letters of John Keats, 1814–1821*. Harvard University Press, 1958. Letter to George and Thomas Keats, 21 December 1817, introducing “Negative Capability.” Accessed 2026-07-18.
- Roderick J. A. Little. A test of missing completely at random for multivariate data with missing values. *Journal of the American Statistical Association*, 83(404):1198–1202, 1988. doi: 10.1080/01621459.1988.10478722. URL <https://doi.org/10.1080/01621459.1988.10478722>. A global test statistic for MCAR; rejecting one mechanism does not identify another. Accessed 2026-07-28.
- Roderick J. A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data*. Wiley, 3rd edition, 2019. doi: 10.1002/9781119482260. URL <https://onlinelibrary.wiley.com/doi/book/10.1002/9781119482260>. Modern treatment of inference under missing-data mechanisms and the assumptions required for analysis. Accessed 2026-07-18.
- Moses Maimonides. *The Guide of the Perplexed*. University of Chicago Press, 1963. Judeo-Arabic original completed c. 1190; cited in Shlomo Pines’s 1963 translation. Develops negative (apophatic) predication of the divine. Accessed 2026-07-18.
- Charles F. Manski. *Partial Identification of Probability Distributions*. Springer Series in Statistics. Springer, New York, 2003. ISBN 9780387004549. doi: 10.1007/b97478. URL <https://doi.org/10.1007/b97478>. Reports the bounds that data and stated assumptions support instead of a point estimate requiring more. Accessed 2026-07-28.
- Linsey McGoey. The logic of strategic ignorance. *The British Journal of Sociology*, 63(3):533–576, 2012. doi: 10.1111/j.1468-4446.2012.01424.x. URL <https://onlinelibrary.wiley.com/doi/10.1111/j.1468-4446.2012.01424.x>. Analysis of ignorance as a productive organizational resource rather than merely a deficit of knowledge. Accessed 2026-07-18.
- José Medina. *The Epistemology of Resistance: Gender and Racial Oppression, Epistemic Injustice, and Resistant Imaginations*. Oxford University Press, 2013. doi: 10.1093/acprof:oso/9780199929023.001.0001. URL <https://academic.oup.com/book/9202>. Contextualist account of resistance, complicity, and shared responsibility for epistemic conditions. Accessed 2026-07-18.
- Thomas Merton. *New Seeds of Contemplation*. New Directions, 1961. On contemplative silence and the refusal to possess the ineffable as a claim. Accessed 2026-07-18.

- Charles W. Mills. White ignorance. In Shannon Sullivan and Nancy Tuana, editors, *Race and Epistemologies of Ignorance*, pages 11–38. State University of New York Press, Albany, 2007. URL <https://sunypress.edu/Books/R/Race-and-Epistemologies-of-Ignorance>. Analysis of racialized ignorance as an active and socially organized epistemology. Accessed 2026-07-18.
- Geert Molenberghs, Caroline Beunckens, Cristina Sotito, and Michael G. Kenward. Every missingness not at random model has a missingness at random counterpart with equal fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(2):371–388, 2008. doi: 10.1111/j.1467-9868.2007.00640.x. URL <https://doi.org/10.1111/j.1467-9868.2007.00640.x>. Shows that observed data alone cannot adjudicate between MNAR and MAR models, so the mechanism is an assumption. Accessed 2026-07-28.
- Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(1):119–158, 2004. URL <https://digitalcommons.law.uw.edu/wlr/vol79/iss1/10/>. Privacy benchmark requiring information flows to fit the norms of their specific context. Accessed 2026-07-18.
- Nāgārjuna. *The Fundamental Wisdom of the Middle Way: Nāgārjuna’s Mūlamadhyamakakārikā*. Oxford University Press, 1995. Sanskrit original composed c. 2nd century CE; cited in Jay L. Garfield’s 1995 translation and commentary. Emptiness must not be reified into a hidden substance. Accessed 2026-07-18.
- Naomi Oreskes and Erik M. Conway. *Merchants of Doubt: How a Handful of Scientists Obscured the Truth on Issues from Tobacco Smoke to Global Warming*. Bloomsbury Press, New York, 2010. ISBN 9781596916104. Historical account of organized campaigns that manufactured the appearance of unsettled science. Accessed 2026-07-28.
- Plato. *Five Dialogues: Euthyphro, Apology, Crito, Phaedo, Phaedrus*. Hackett Publishing Company, 2nd edition, 2002. The Apology dramatizes Socrates’s disavowal of knowledge he does not have. Accessed 2026-07-18.
- Robert N. Proctor and Londa Schiebinger. *Agnotology: The Making and Unmaking of Ignorance*. Stanford University Press, 2008. Finds agnotology, the study of how ignorance is produced and structured. Accessed 2026-07-18.
- Pseudo-Dionysius the Areopagite. *Pseudo-Dionysius: The Complete Works*. Classics of Western Spirituality. Paulist Press, 1987. Contains The Mystical Theology, a founding text of apophatic (via negativa) discourse. Accessed 2026-07-18.
- Steve Rayner. Uncomfortable knowledge: the social construction of ignorance in science and environmental policy discourses. *Economy and Society*, 41(1):107–125, 2012. doi: 10.1080/03085147.2011.637335. URL <https://doi.org/10.1080/03085147.2011.637335>. Names denial, dismissal, diversion, and displacement as institutional strategies for excluding knowledge that will not fit. Accessed 2026-07-28.
- Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976. doi: 10.1093/biomet/63.3.581. URL <https://doi.org/10.1093/biomet/63.3.581>. Foundational framework for inference under missing-data mechanisms. Accessed 2026-07-18.
- Michael A. Sells. *Mystical Languages of Unsayings*. University of Chicago Press, Chicago, 1994. ISBN 9780226747866. URL <https://press.uchicago.edu/ucp/books/book/chicago/M/bo3617573.html>. Reads apophysis as a performative regress of saying and unsaying rather than as a description of an ineffable object. University of Chicago Press, 1994. ISBN 9780226747866. URL may have moved; book available in libraries under this ISBN. Accessed 2026-07-28.
- Audra Simpson. *Mohawk Interruptus: Political Life Across the Borders of Settler States*. Duke University Press, 2014. doi: 10.1215/9780822376781. URL <https://www.dukeupress.edu/mohawk-interruptus>. Ethnographic and political account of refusal, sovereignty, and the limits of incorporation by dominant institutions. Accessed 2026-07-18.
- Shannon Sullivan and Nancy Tuana, editors. *Race and Epistemologies of Ignorance*. State University of New York Press, Albany, 2007. ISBN 9780791471012. URL <https://sunypress.edu/Books/R/Race-and-Epistemologies-of-Ignorance>. Edited collection on how ignorance is produced and sustained in knowledge practices. Accessed 2026-07-18.
- Nassim Nicholas Taleb. *The Black Swan: The Impact of the Highly Improbable*. Random House, 2007. On consequential events outside the space of anticipated possibilities. Accessed 2026-07-18.
- Eve Tuck. Suspending damage: A letter to communities. *Harvard Educational Review*, 79(3):409–428, 2009. doi: 10.17763/haer.79.3.n0016675661t3n15. URL <https://doi.org/10.17763/HAER.79.3.N0016675661T3N15>. Critique of damage-centered research and its tendency to reproduce one-dimensional accounts of communities. Accessed 2026-07-18.
- John W. Tukey. *Exploratory Data Analysis*. Addison-Wesley, Reading, MA, 1977. ISBN 978-0-201-07616-5. Descriptive display and arrangement of data as a discipline in its own right, prior to and distinct from confirmatory inference.
- Denys Turner. *The Darkness of God: Negativity in Christian Mysticism*. Cambridge University Press, Cambridge, 1995. ISBN 9780521453172. doi: 10.1017/CBO9780511583131. Argues that the medieval negative tradition is a discipline of language rather than a report of extraordinary inner experience. Accessed 2026-07-28.

Ludwig Wittgenstein. *Tractatus Logico-Philosophicus*. Kegan Paul, Trench, Trubner & Co., 1922. German original 1921; proposition 7 closes the work on what cannot be said. Accessed 2026-07-18.