

The Line Set: The Collected Volume

5 works compiled unchanged in substance, with one merged bibliography and one reading of the set

Daniel Ari Friedman

Active Inference Institute

daniel@activeinference.institute

ORCID: 0000-0001-6232-9096

DOI: 10.5281/zenodo.21754244

2026-07-29

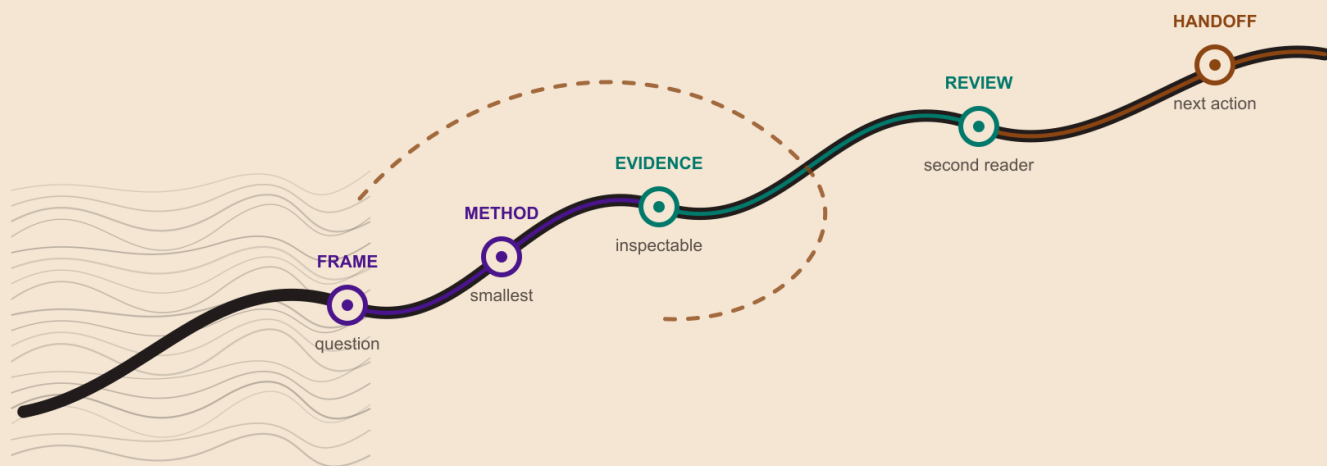
BLACK LINE · A POSITIVE METHOD

0.4 · LIVING METHOD

A line that can be followed

From intention to inspection to handoff

The work becomes reviewable where another reader can pick up the trail.



RED LINE remains a boundary, not a score

the reviewer follows the trail

noise → frame → method → evidence → review → handoff

BLACK LINE

Contents

1	About this compiled volume	6
1.1	The works in this volume	6
1.2	What the compilation changed, and what it did not	6
1.3	What is missing from this volume	6
2	The Line Set: Holding Instruments Apart	7
2.1	Abstract	8
2.2	Introduction: what keeps four instruments from becoming one	10
2.3	The set: four questions, four jobs, and one shared word	11
2.3.1	The one shared word	11
2.3.2	The colours, said plainly	11
2.4	Method: a declaration, a staged reader, and one seam	14
2.4.1	The declaration	14
2.4.2	The reader	14
2.4.3	The seam	14
2.4.4	Failing closed	16
2.4.5	The checks	16
2.4.6	Digests	16
2.5	The instrument stated formally	17
2.5.1	The declared objects	17
2.5.2	The exemption matcher	17
2.5.3	The reading	17
2.5.4	The self-application	17
2.5.5	What the statements do not carry	19
2.6	Scholarship: what is old about this problem, and what my check is not	21
2.6.1	Modules defined by what they hide	21
2.6.2	Prefixes, and what they are worth without an authority	21
2.6.3	Coordinating without merging	21
2.6.4	The hazard of the index	22
2.7	Extensibility: adding a colour is an edit to one file	23
2.7.1	The executed example	23
2.7.2	What the digest change means	24
2.7.3	What this does not establish	24
2.8	Worked readings	25
2.8.1	The live set	25
2.8.2	A second line adopts the word	25
2.8.3	A package nobody declared	25
2.8.4	Lines that will not read	27
2.8.5	No siblings at all	27
2.8.6	A weakened exemption	27
2.9	Limits and Epistemic Boundaries	29
2.9.1	The reader reads declarations	29
2.9.2	Disjoint vocabularies are not disjoint concepts	29
2.9.3	A digest is not tamper evidence	29
2.9.4	The reader cannot say whether a line is worth having	29
2.9.5	The reader cannot see a missing colour	29
2.9.6	Smaller boundaries worth naming	29
2.10	Conclusion	31
3	Personal Red Lines for Development	32
3.1	Abstract	34
3.2	Introduction	35
3.2.1	Four propositions	35
3.2.2	What this paper can establish	35
3.2.3	What the artifact is for	35
3.2.4	Beacon, evaluator, canary	35
3.2.5	A bounded comparison	37
3.3	Background — Turner as a bounded mechanism source	38

3.3.1	What is and is not transferred	38
3.3.2	Provenance labels for the adaptation	38
3.4	Global and Historical Scholarship: Situated Questions, Not a Universal Lineage	39
3.4.1	Before 1900: refusal, rule, knowledge, and power	39
3.4.2	Modern critical and institutional scholarship	39
3.4.3	Export control and the refusal of work	40
3.4.4	Selection limits and positionality	41
3.5	The four-line set: boundary, method, aspiration, absence	44
3.5.1	Note on the name	45
3.6	First-principles design: what the artifact can actually do	46
3.6.1	Deconstruction	46
3.6.2	Fundamental truths	46
3.6.3	Constraint analysis	46
3.6.4	Reconstruction: what the ordering has to be	47
3.6.5	Claim classes	47
3.7	Operating method: from first principles to a bounded release	48
3.7.1	The unit of analysis	48
3.7.2	The operating loop	48
3.7.3	Evidence states and stop points	48
3.7.4	Falsification and negative controls	50
3.7.5	Scope of inference	50
3.8	The Adaptation Thesis	51
3.8.1	Four adaptations	51
3.9	The two CANARY-severity boundaries	52
3.9.1	S1 — force and harm-capable systems	52
3.9.2	S2 — untargeted profiling	52
3.9.3	Canary status	52
3.10	Deployment tiers as oversight-retention grades	53
3.11	Written review and transparency	55
3.11.1	Finding record	55
3.11.2	Escalation is not permission	55
3.12	Durability and transparency: a hash-based canary	56
3.12.1	Deterministic registry hashing	56
3.12.2	The canary statement	56
3.12.3	Verification and escalation	56
3.12.4	The honest trust model	56
3.12.5	Defensive security context	56
3.13	Evidence, ambiguity, and evaluation	58
3.13.1	Exercised outcome coverage	60
3.13.2	Residual lexical limitation	61
3.14	The instrument stated formally	62
3.14.1	The domain objects	62
3.14.2	The decision rule	63
3.14.3	The report envelope	64
3.14.4	What binds each proposition	64
3.14.5	What the formalism does not establish	65
3.15	The red-line registry	68
3.15.1	s1-human-control-force — Human control over force and harm-capable systems	68
3.15.2	s2-untargeted-profiling — No untargeted profiling or mass surveillance	68
3.15.3	dual-use-ablation — Scoped release of dual-use models	70
3.15.4	cogsec-integrity — Cognitive security strengthens, never degrades, the epistemic commons	70
3.15.5	provenance-and-consent — Provenance and consent for data and identity	71
3.15.6	open-science-good-faith — Open-science claims are honest and reproducible	71
3.15.7	downstream-transfer — No knowing transfer to a violating end use	71
3.16	Registry composition as derived data	73
3.16.1	Severity and tier floors	73
3.16.2	Scope vocabulary and overlap points	73
3.16.3	Per-line structure	73
3.16.4	Evidence depth and the free-pass check	74

3.17	Limitations and negative space	77
3.17.1	What this document does not decide	77
3.17.2	Adversarial declarations	77
3.18	Conclusion	79
4	Black Line: Strong Work in Public	80
4.1	Abstract	81
4.2	Introduction	82
4.3	Relationship to the line set	83
4.3.1	Note on the name	83
4.4	Intellectual lineage: where the practices come from	85
4.4.1	Framing and falsification	85
4.4.2	Traceability, review, and the norms of science	85
4.4.3	Traceable prose and the smallest sufficient method	85
4.4.4	Reproducibility as the load-bearing modern practice	85
4.4.5	Reproducibility is not replication	85
4.4.6	Openness is infrastructure; review is a social act	86
4.4.7	Craft, technē, and legibility as a designed partial view	86
4.4.8	Verification is not validation	86
4.4.9	Handoffs are situated coordination objects	86
4.4.10	Humility is an operational requirement	87
4.4.11	What the lineage does and does not license	87
4.5	Method: positive wires and observable evidence	88
4.5.1	Six practice families: decision, procedure, record, world, authority	88
4.5.2	Three claim classes, three evidentiary burdens	88
4.6	Operating protocol: the smallest honest loop	90
4.7	The Black Line practices	91
4.7.1	Registry coverage and burden	91
4.8	Formal method: the evaluator and its invariants	97
4.8.1	Domain objects	97
4.8.2	Status codomains	97
4.8.3	The staged evaluator	97
4.8.4	Surfaces, projection, and the report envelope	98
4.8.5	Propositions	101
4.8.6	Structural invariants	103
4.8.7	Claim-to-test binding	104
4.9	Executed examples and boundaries	108
4.9.1	An executed incremental-declaration path	108
4.9.2	The decay sweep, executed	111
4.9.3	The refresh queue, executed	112
4.9.4	A batch, executed	112
4.9.5	Boundary cases	114
4.10	Limits and Epistemic Boundaries	115
4.10.1	Adversarial declarations	115
4.10.2	Legibility bias	115
4.10.3	A design claim, not an outcome claim	116
4.10.4	Bounded by the line set	116
4.11	Conclusion	117
5	Golden Line: Toward What Matters	118
5.1	Abstract	120
5.2	Introduction: the work needs a direction	121
5.3	Relationship to the line set	122
5.3.1	Note on the name	122
5.4	Method: aspiration as a directional record	123
5.4.1	The registry entry	123
5.4.2	The horizon entry	123
5.4.3	The staged evaluator	123
5.4.4	Conservative precedence	123
5.4.5	Evidence and artifact boundary	123

5.4.6	The descriptive analysis layer	123
5.5	Formalism: the evaluator and its invariants	125
5.5.1	Domain objects	125
5.5.2	The staged evaluator	125
5.5.3	Propositions about the evaluator	126
5.5.4	Structural invariants	130
5.5.5	The report envelope	131
5.5.6	Formalism-to-test bindings	131
5.6	Scholarship: an aspiration is a direction, not a score	135
5.6.1	A translation, not a synthesis	135
5.6.2	The shape of an end	136
5.6.3	Capability and flourishing	136
5.6.4	Attention, craft, and transfer	136
5.6.5	Commons, power, and return	137
5.6.6	Why it must not become a score	137
5.6.7	From theory to instrument	137
5.7	Evidence boundary and artifact chain	139
5.7.1	Hard constraints and soft choices	139
5.7.2	Statuses as bounded claims	139
5.7.3	Source to publication	139
5.8	The Golden Line aspirations	142
5.8.1	The signal vocabulary in aggregate	144
5.8.2	The reach of the nine horizons	144
5.9	Worked records	147
5.10	Reading a batch: the descriptive layer at work	150
5.10.1	A worked batch	150
5.10.2	What intake set aside	151
5.10.3	Vocabulary and reach as reading context	152
5.11	Limits and safeguards	153
5.11.1	Limits of the descriptive analysis layer	153
5.12	Conclusion	154
6	White Line: A Typed Ledger for the Edge of the Claim	155
6.1	Abstract	156
6.2	Introduction: the discipline of not filling the gap	157
6.3	Relationship to the line set	158
6.3.1	Note on the name	158
6.4	Scholarship: epistemic boundaries, classification, and refusal	159
6.4.1	Epistemic absence: knowing the edge of what one knows	159
6.4.2	Classification and information infrastructures	159
6.4.3	Apophatic and contemplative traditions: the discipline of not saying	160
6.4.4	Ethical restraint: absence as an obligation to people	160
6.4.5	The translation test: what the instrument takes, and what it refuses	162
6.5	Method: three kinds of absence, four ledger states	164
6.6	Review protocol: from state to responsible follow-up	166
6.7	Reproducibility boundary and release chain	167
6.8	The three White Line layers	168
6.9	Formal method: the evaluator and its invariants	171
6.9.1	Domain objects	171
6.9.2	The staged evaluator	171
6.9.3	Semantic guarantees	173
6.9.4	The witness layer	178
6.9.5	Structural invariants	179
6.9.6	Where each claim is checked	179
6.10	Worked ledger entries	182
6.11	Reading one report distributionally	183
6.11.1	A worked intake under careless and adversarial input	183
6.11.2	A zero row is untriggered, not unreachable	183
6.11.3	Where the states sit across the three kinds	185

6.11.4	What comes due next, and what cannot come due at all	185
6.11.5	The state as a safe projection: the witness layer	185
6.12	Limits and safeguards	190
6.13	Conclusion	191

1 About this compiled volume

This is a compiled volume. It adds no substantive instrument. The work that assembled it declares the set, reads whichever declared packages are installed, and checks that no two of them gave the same spelling to different things; it does not evaluate, rank, reconcile, or summarise the works it reproduces.

Each of the 5 works below is reproduced from its own source at the version and registry digest stated in the table, unchanged in substance. The compilation asserts nothing the individual works do not assert. Its one added claim is the reading recorded immediately below, and that reading says only that the declared vocabularies of the works that could be read did not overlap.

The reading, computed at assembly: status `set_legible`, read as of 2026-07-29, set digest `40db5e0e3e03...`, reading digest `e783001a0baa...`. It is not a finding that the instruments are correct, complete, good, or worth having, and it is not a finding about behaviour.

1.1 The works in this volume

Work	Version	Registry size	Registry digest	Volume sections
The Line Set: Holding Instruments Apart	0.1.0	4	40db5e0e3e03...	1-10
Personal Red Lines for Development	0.3.0	7	72835fd81d1f...	11-28
Black Line: Strong Work in Public	0.4.0	11	a02bff47a767...	29-39
Golden Line: Toward What Matters	0.4.0	9	3e0a7e38ecec...	40-51
White Line: A Typed Ledger for the Edge of the Claim	0.7.0	11	11047a641a8e...	52-64

Versions are read from each work's own `manuscript/config.yaml`, which the set contract treats as authoritative. Registry sizes and digests are read from the installed packages by this volume's own reader; a work that could not be read reports no size and no digest rather than a placeholder. A volume section number counts the source sections in reading order, front matter and part headings excluded.

1.2 What the compilation changed, and what it did not

Four mechanical transforms were applied and nothing else. Anchors were moved into a namespace per source work, so two works that both wrote `{#sec:abstract}` no longer collide, and every reference to a moved anchor was moved with it. Headings were shifted down one level so each work could be given a part heading carrying its own title. Figure embeds were retargeted at this volume's plate directory. The bibliographies were merged.

No claim, caveat, limit, or abstract was altered, and none was omitted. The transforms are invertible and the assembling project's test suite inverts them back to the source bytes for every section reproduced here.

The merged bibliography holds 141 distinct keys. 9 key(s) were declared by more than one work; for each, both declarations named the same work, the earlier work's entry was kept verbatim, and the later work's `note` gloss was dropped.

1.3 What is missing from this volume

Every declared work was found and reproduced. Nothing declared is absent from this volume.

The volume compiles under the assembling work's own LaTeX preamble. These directives from other works' preambles are not carried into it:

- `\usepackage{xcolor}` (from `black_line`)
- `\usepackage{etoolbox}` (from `black_line`)
- `\pagecolor{white}` (from `black_line`)
- `\AtBeginEnvironment{thebibliography}{\small}` (from `black_line`)

2 The Line Set: Holding Instruments Apart

A Declaration, a Reader, and a Non-Overlap Contract for a Growing Set of Small Instruments

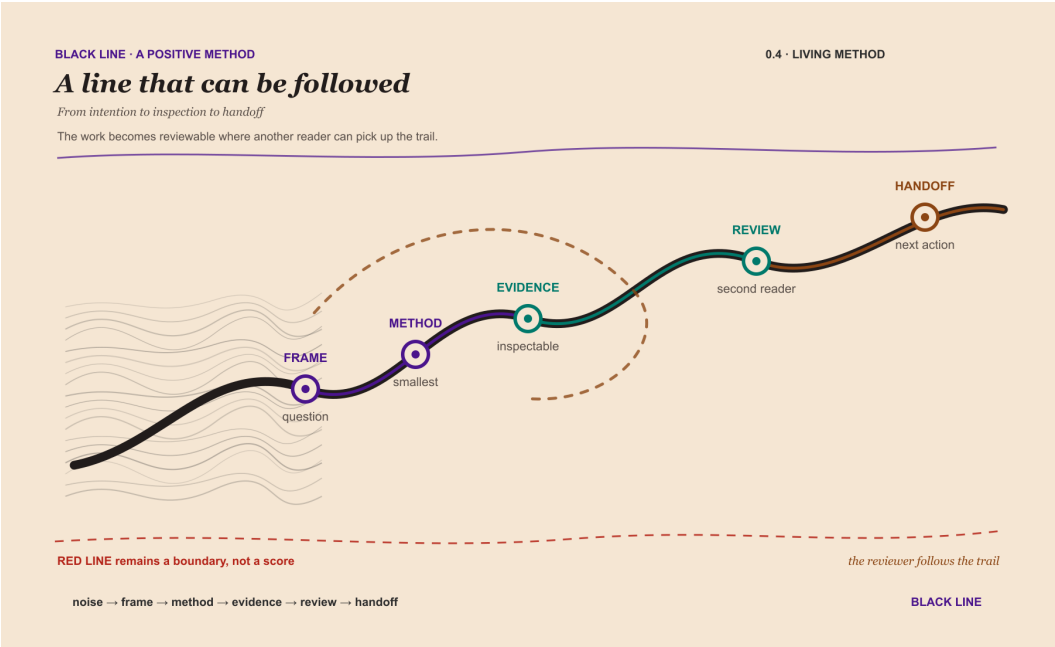


Figure 1: Cover art for The Line Set: Holding Instruments Apart

Reproduced unchanged in substance from its own source at version 0.1.0. It answers one question: How does a growing set of instruments stay separate?

2.1 Abstract

Four small instruments sit in my working tree, and each was built so the others would not have to answer its question. Red Line records what I refuse. Black Line records how I try to do strong work. Golden Line records what is worth reaching toward. White Line records what is absent, withheld, or unknowable. This fifth work adds no instrument of its own. It holds the declaration of what the set is, reads whichever sibling packages happen to be installed, and checks one narrow property: that no two lines have given the same spelling to different things.

The package is pure standard library and depends on nothing. Its declaration is four **LineEntry** records and one shared token. Its reader runs five recorded stages — resolve, bind, collide, declare, status — and returns one of four readings, in fixed precedence: **SET_COLLIDING**, then **SET_UNDECLARED**, then **SET_PARTIAL**, then **SET_LEGIBLE**. A live collision is therefore never hidden behind a missing install, and **SET_LEGIBLE** is reachable only when nothing above it matched. Each line's read outcome is one of four codes, and a line that cannot be read yields no version, no registry size, and no digest rather than a placeholder.

Seven structural checks run offline over the declaration alone; an eighth, which needs the siblings importable, applies this package's own collision check to a declaration that includes this package; it is the reason the four reading statuses are **SET_**-prefixed, and it is the property the project exists to hold. The objects, the five stages, the precedence rule, the exemption matcher's fail-closed conditions, and that self-application are each stated as a definition or proposition and each bound by a test that re-derives it from the code.

SET_LEGIBLE means one thing and I want it stated before anything else: the declared vocabularies of the lines that could be read did not overlap. It is not a finding that the instruments are correct, complete, good, sufficient, or worth having, and it is not a finding about behaviour. The reader reads names.

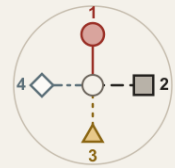
Read against the four installed siblings on 2026-07-29, the four package roots published nineteen enum classes declaring 89 members between them. The reader deduplicates within each line, since a line that spells one word in two of its own enums has still said one word: that gives 81 line-and-name pairs spanning 80 distinct names. Exactly one name is carried by more than one line. It is **OUTSIDE_SCOPE**, carried by `red_line` and `black_line`, and it is the one collision the set had already declared and disambiguated. What that establishes is that the declared vocabularies are disjoint. It does not establish that the instruments are conceptually non-overlapping: two lines can share no spelling at all and still be thinking about the same thing.

THE SET COMPASS

Which question sends you to which line

Each card is one declared line: the question it answers, the job it does, and the thing it must not become.

DRAWN FROM THE DECLARATION · 4 LINES · NO LINE RANKS ABOVE ANOTHER



working positions



red

package red_line · keeps red lines · emits classification

opus stage · rubedo

asks — What must I refuse?

does — Security boundary and explicit No document

must not become — A complete ethics system or an enforcement mechanism



black

package black_line · keeps practices · emits assessment status

opus stage · nigredo

asks — How do I do strong work?

does — Positive wire for concise, rigorous research and engineering

must not become — Permission to cross Red Line



golden

package golden_line · keeps aspirations · emits horizon status

opus stage · citrinitas

asks — What is worth reaching toward?

does — Aspirational thread and long-horizon direction

must not become — A compliance score or proof of virtue



white

package white_line · keeps absence records · emits absence state

opus stage · albedo

asks — What is absent or unknowable?

does — Epistemic gaps, ethical restraint, omission, silence, and negative space

must not become — Evidence that an absent thing is safe or true

BELOW THE RULE — NOT A LINE IN THE SET

the wrapper declares no refusal, method, aspiration, or absence



wrapper

colourless

package line_set · keeps line entries · emits set status

no opus stage

asks — How does a growing set of instruments stay separate?

does — Declare the set, read it, check non-overlap, admit new colours

must not become — A meta-evaluator, a fifth substantive instrument, a merge of the four, or a ranking of them

A card states what its line is for. It is not evidence that the line does it, and no card is a score.

Figure 2: Set compass: the declared lines, each with the question it answers, the job it does, and the thing it must not become. The plate is rendered from LINE_SET and re-renders correctly when a colour is appended; it is an orientation aid, not a ranking.

2.2 Introduction: what keeps four instruments from becoming one

For most of the time the four lines have existed, their separation was a sentence in a markdown file. Each project’s documentation said it did not duplicate the others’ registries, evaluators, figures, or conclusions. I believed it, and I had written it, and nothing anywhere checked it.

Two things go wrong when a set of small instruments grows, and they go wrong quietly. The first is that two instruments give the same name to different things — `OUTSIDE_SCOPE` is the set’s only such collision, defined in [Definition 2](#). Someone reading a stored finding sees that word and has to already know which line produced it before it means anything; if they guess wrong, they read a refusal as a competence judgement. The second is that one instrument slowly takes on another’s job — an aspiration registry acquires a prohibition, a method checklist starts issuing permissions — until the set is one instrument wearing four names. The second failure is the more serious of the two. It is also the one I cannot check by machine.

So this work addresses the first, and it addresses it narrowly. It declares the set as data, reads whichever line packages are installed, collects the enum member names each of them publishes at its package root, and reports whether any name is carried by more than one line. A name carried by two lines is a collision. A collision the set has already written down, with a separate meaning recorded for each line, is exempted. Any other collision makes the reading `SET_COLLIDING`, and that is the whole of the contract.

I want to be exact about the size of that claim, because a wrapper is the easiest place in a project to overstate one. Disjoint vocabularies are a necessary condition for the separation I want, not a sufficient one. Two lines can share no spelling whatsoever and be duplicating each other’s judgement in every particular; the reader would call that set legible and be technically right and substantively useless. The check catches the failure that produces silent misreads downstream. It does not catch conceptual overlap, and nothing in this package should be cited as though it did.

There is a hazard specific to being the fifth work, and I have tried to design against it rather than promise around it. An instrument that sits above four others is well positioned to become a scoreboard for them — to start reporting which line is healthiest, which is best maintained, which one earns its place. The four lines are peers. Nothing here ranks them, aggregates them, merges them, or evaluates them, and the package’s own entry in `registry.py` records that as the thing it must not become, in the same field every line uses for the same purpose. The entry is deliberately colourless and carries no stage, because the wrapper answers no question about refusal, method, aspiration, or absence.

The design constraint that follows is the part I find most worth reporting. A package that checks other packages for shared tokens has to survive that check itself. If this one had named its reading statuses `LEGIBLE`, `PARTIAL`, `COLLIDING`, and `UNDECLARED`, it would have been one plausible sibling release away from colliding with the very set it reads. The statuses are `SET_`-prefixed instead, and a check appends the wrapper’s own entry to the declaration and runs the ordinary reader over the result. That check is the only one in the battery that needs the siblings importable, so it sits outside the offline battery and reports failure — not silence, and not a pass — when it has nothing to compare against.

The rest of the paper describes the declaration and the reader, states both as definitions and propositions that a test re-derives from the code, situates the separation question in work on modularity and on artifacts that coordinate without merging, shows an executed example of appending a fifth colour, walks through readings I actually took, and states what the instrument cannot see.

The paper proceeds as follows. [Section sec. 2.3](#) names the four instruments and the one shared word. [Section sec. 2.4](#) defines the declaration, the staged reader, and the seam. [Section sec. 2.5](#) states the instrument formally. [Section sec. 2.7](#) demonstrates that adding a colour is an edit to one file, and [Section sec. 2.9](#) closes with epistemic boundaries.

2.3 The set: four questions, four jobs, and one shared word

The declaration is small enough to print. Each line answers one question, does one job, and names the thing it must not become; the last field is the one that does the work, because a boundary is easier to hold when the drift it guards against has been written down in advance.

Line	Question	Job	Must not become
Red Line	What must I refuse?	Security boundary and explicit No document	A complete ethics system or an enforcement mechanism
Black Line	How do I do strong work?	Positive wire for concise, rigorous research and engineering	Permission to cross Red Line
Golden Line	What is worth reaching toward?	Aspirational thread and long-horizon direction	A compliance score or proof of virtue
White Line	What is absent or unknowable?	Epistemic gaps, ethical restraint, omission, silence, and negative space	Evidence that an absent thing is safe or true

The order is the order I use them in. Red stops a prohibited direction before any work begins. Black gives the work that survives that stop a method. Golden keeps the method pointed at something worth serving. White keeps every claim honest about what was never observed. Each line has its own package, registry, evaluator, figures, tests, and manuscript, and a cross-reference between them is an orientation link rather than a dependency. A separated copy of any one of them still has to explain itself.

2.3.1 The one shared word

Red Line and Black Line both spell a status `OUTSIDE_SCOPE`, and they mean different things by it. In Red Line it marks a complete, evidenced intake that implicates no red line — the instrument looked and found no prohibition. In Black Line it marks an attempt outside the discipline’s evaluation scope — the instrument did not look, because the attempt is not the kind of thing its practices assess. One is a finding of absence after inspection. The other is a declination to inspect.

Both lines needed a name for work their own question does not reach, and the two senses did not turn out to be the same sense. Rather than rename one of them, the set records the overlap: `SHARED_TOKENS` holds a single entry naming the token, the exactly two lines allowed to carry it, a distinct meaning for each, and a rationale. The exemption table is part of the declaration and travels inside the set digest, so a set whose exemptions changed is a different set even when its lines did not.

2.3.2 The colours, said plainly

The four colour names openly echo the classical stages of the alchemical *magnum opus*: nigredo the blackening, albedo the whitening, citrinitas the yellowing, rubedo the reddening. Anyone who knows the opus will hear it, and it is meant to be heard. I read it in exactly one register — the symbolic-psychological one, following Jung’s treatment of the stages as figures for individuation rather than as laboratory chemistry [Jung, 1953]. The echo is a naming resonance. It is not an empirical, mystical, or causal claim; the colour of a line has no operative power. It is also not a doctrinal endorsement of alchemy or of Jung’s psychology, which would be a much larger commitment than borrowing four words.

Three caveats keep the resonance from being read as more than it is, and the first two of them are checked in code rather than merely promised here.

The working order is not the opus order. The set runs refuse, method, aspire, account-for-absence: Red, Black, Golden, White. The opus runs nigredo, albedo, citrinitas, rubedo: Black, White, Golden, Red. The working order was chosen for how the instruments actually support the work, not for any dependency between them — each line stands alone and a cross-reference is an orientation link — and the two sequences deliberately do not line up. `check_orders_diverge` sorts the declaration both ways and fails if they ever coincide, which keeps the claim true by construction instead of by assertion.

Citrinitas is kept distinct on purpose. After the fifteenth century the yellowing stage was frequently folded into the reddening, leaving a three-stage nigredo–albedo–rubedo reading. The set does not perform that collapse. Golden Line stays its own instrument with its own question, and `OPUS_STAGE_ORDER` carries four stages rather than three, so a future edit that dropped citrinitas would have to do so visibly.

No borrowed telos and no ranking. The opus is a teleological arc toward gold and a completed stone. The set borrows the stage names as symbols and claims none of the direction. Calling Red Line rubedo does not make it final, perfected, or the culmination of the other three; calling Black Line nigredo implies no deficiency in it. The colours are peers. No line completes, supersedes, or outranks another, and this package has no aggregate, no score, and no ordering by merit to offer.

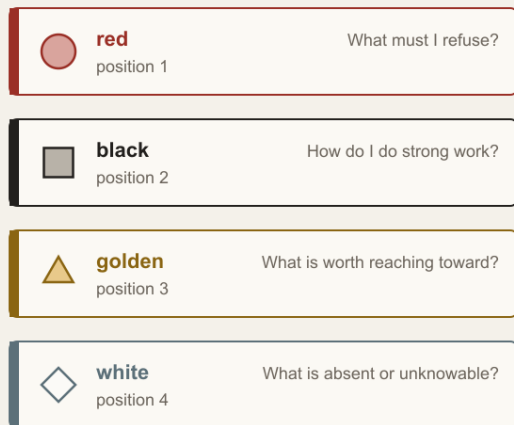
TWO ORDERS

The working order is not the opus order

Left: the sequence the instruments are used in. Right: the borrowed stage names, in their own sequence.

THE CROSSINGS ARE THE POINT · THE DIVERGENCE IS DELIBERATE

WORKING ORDER refuse → method → aspire → absence



OPUS ORDER nigredo → albedo → citrinitas → rubedo



LIVE CHECK · orders_diverge · holds

working order ['red_line', 'black_line', 'golden_line', 'white_line'] differs from opus order ['black_line', 'white_line', 'golden_line', 'red_line']

1. The working order is functional. The opus order is a borrowed set of stage names. The two deliberately do not line up.
2. Citrinitas is kept as a stage of its own. This set does not perform the later collapse of the yellowing into the reddening.
3. No telos is borrowed and nothing is ranked. A stage name is a label, not a claim that a line is final, complete, or above another.

Figure 3: Two orders: the working sequence (refuse, method, aspire, absence) drawn against the opus sequence (nigredo, albedo, citrinitas, rubedo), with connectors showing where they cross. The divergence is deliberate, and citrinitas is deliberately not collapsed into rubedo. Position in either column encodes sequence only, never rank.

The set's short internal orientation note, `docs/line-set.md`, lives in the author's private projects tree; it is unpublished and does not travel with this repository, so it is cited here by name rather than linked. Each instrument it maps is its own repository — **Red Line**, **Black Line**, **Golden Line**, and **White Line** — and those are the durable references. This paper is the version with a package under it.

2.4 Method: a declaration, a staged reader, and one seam

2.4.1 The declaration

A `LineEntry` has ten fields. Four are identity and prose — the line’s id, its colour, the question it answers, the job it does. One is the boundary: `must_not_become`, which every line fills in and which a structural check refuses to let go blank. Two are ordering: `opus_stage`, which may be absent, and `working_position`, a one-based integer that must land in a contiguous `1..N` across the whole declaration. The last three are addresses. `package_name` is what the reader asks the import system for, and it is kept separate from `id` so that a line’s identity in the declaration does not depend on how its code happens to be distributed. `registry_noun` and `verdict_noun` are prose labels for what a line keeps and what it emits; figures and manuscript text use them, and the reader never uses them to find anything.

A `SharedToken` has four: the token, the tuple of line ids allowed to carry it, a tuple of per-line meanings, and a rationale. The meanings tuple is the load-bearing one. A shared token with no meaning recorded for each line it names is a label, and a label cannot show that two uses of one spelling are two different things.

The two record types are stated as [Definition 1](#) and [Definition 2](#). The declaration ships four line entries and one shared token. The wrapper’s own entry lives beside `LINE_SET` rather than inside it, because it is not a fifth instrument; its `working_position` continues the sequence so that appending it still yields a contiguous run.

2.4.2 The reader

`read_set` takes the lines, the exemption table, an optional resolver, and an optional review date, and runs five stages, formalized as [Definition 4](#). Each stage records what it did into the returned derivation, so a reading can be re-read rather than trusted.

Resolve asks the resolver for each declared line’s package and records one of four codes. `RESOLVED` means the package imported. `NOT_INSTALLED` means the import system found nothing. `IMPORT_FAILED` means it found something that raised, and the exception text is kept. `NO_VOCABULARY` means the package imported and exported no enum at its root. None of these is an exception. A line that cannot be read produces no version, no registry size, and no digest, and those fields stay `None`; nothing in the package invents a value for a package it could not read.

Bind reads what a resolved package publishes about itself: a version string from `__version__` or `PROJECT_VERSION`, a registry size, a digest, and the member names of every `enum.Enum` exported at the package root. Only the root is read. A token defined in a submodule and never re-exported is invisible here, which is intentional — the contract is about the vocabulary a line publishes, not about every name that exists somewhere inside it.

Collide inverts the observations into a token-to-lines map and keeps every token carried by more than one line. Each such token is partitioned by asking the exemption matcher whether the set already declared it.

Declare looks for line packages that resolved but appear nowhere in the declaration. This stage depends on the resolver being willing to enumerate; when it is not, the derivation says the scan did not happen rather than reporting that nothing was found. That distinction matters more than it looks: an empty finding and an unasked question read identically in a summary, and only one of them is evidence.

Status applies the fixed precedence of [Proposition 2](#). An unexempted collision yields `SET_COLLIDING` and outranks everything, because it is the single finding that says the set’s contract stopped holding. An undeclared line yields `SET_UNDECLARED`, ranking above a partial read because a line nobody declared is a gap in the declaration rather than a gap in the installation. A line that could not be read yields `SET_PARTIAL`. Only when none of those applies is the reading `SET_LEGIBLE`.

2.4.3 The seam

One module imports siblings. `binding.py` resolves a package by name, reads a version, counts a registry, asks the package for its own digest, and collects enum member names. It never calls a sibling’s evaluator, never passes it data, and never forms an opinion about what a sibling concluded. Every other module in the package is pure computation over a declaration handed to it as an argument.

Two details in that module are deliberate. The digest is always the package’s own, computed by the package’s own function over the package’s own registry; a digest this wrapper derived on a sibling’s behalf would say nothing about whether the sibling agrees with it. And the registry-size rule is a convention read rather than a contract: the size is the length of the unique public, non-empty tuple in the package’s `registry` submodule whose members are all one dataclass type, with the strictly largest winning when several qualify and a tie reporting no size at all rather than a guess. A package that keeps its registry somewhere else simply reports nothing, which is the correct answer to a question the wrapper is not entitled to force.

The default resolver adds nothing to `sys.path` and finds exactly what an ordinary import would find. Looking in the sibling checkouts is an explicit opt-in, and a resolver that was asked to do so records precisely which directories it inserted.

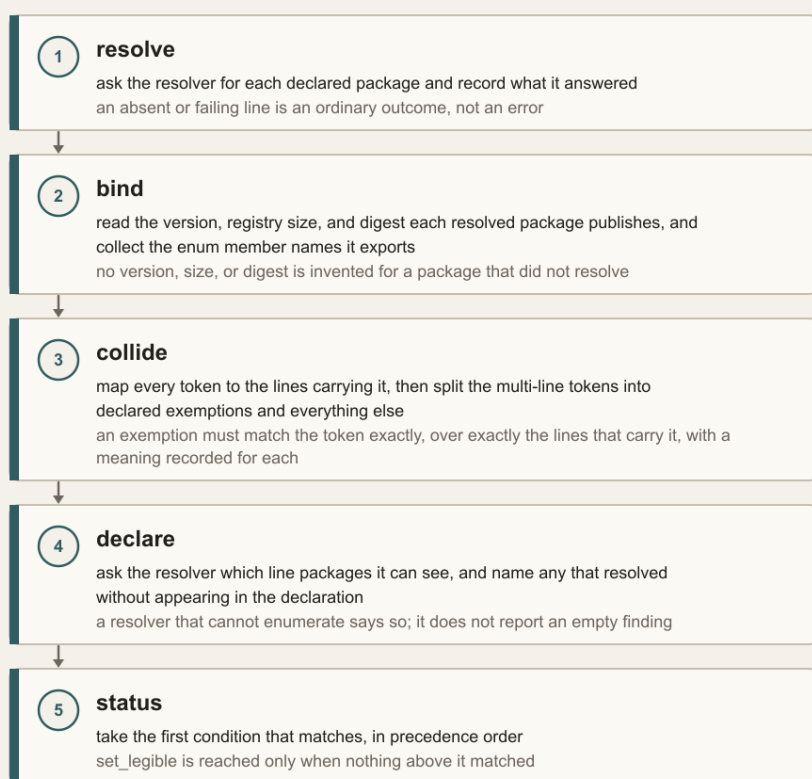
READING THE SET

Five stages, then the first condition that matches

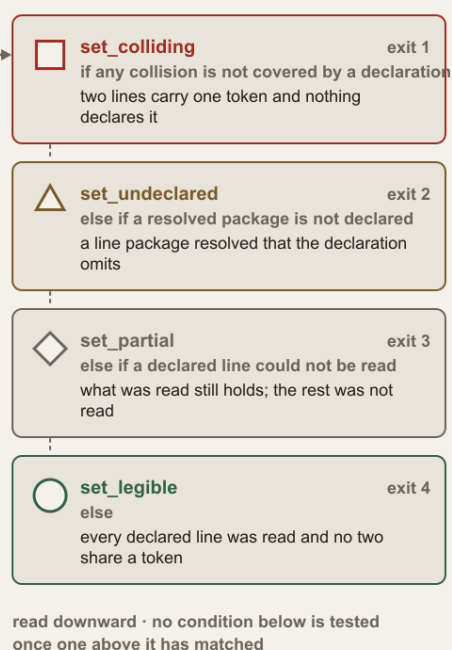
Each stage records what it did into the reading's derivation, so a reading can be re-read rather than trusted.

5 STAGES · 4 EXITS · THE FIRST MATCH WINS

THE STAGES, IN ORDER



THE EXIT LADDER



WHAT AN EXIT ESTABLISHES

An exit reports what the reader found in the declared names. It is not a finding about how any line behaves, and no exit grades a line.

A line that could not be read contributes no version, no registry size, and no digest; those stay empty rather than being filled in.

Figure 4: The reader's five stages, with the status precedence drawn as the exit ladder: a collision leaves at the first rung, an undeclared line at the second, an unread line at the third, and only a reading that reaches the bottom is legible.

2.4.4 Failing closed

The exemption matcher is the only thing standing between a declared exemption and the non-overlap guarantee, so it is written so that a too-generous match is impossible rather than unlikely. Its conditions are stated as [Definition 3](#) and its behaviour as [Proposition 1](#); the design decision behind them is that every condition is an equality and every failure returns no match. A matcher that accepted `OUTSIDE` for `OUTSIDE_SCOPE`, or that let a two-line exemption cover a third line which later adopted the word, would exempt collisions nobody ever declared, and it would do so silently.

2.4.5 The checks

Seven checks run offline over the declaration and import nothing: distinct ids, distinct colours, distinct opus stages drawn from the known stage names, contiguous working positions, working order differing from opus order, a non-blank `must_not_become` on every entry, and an exemption table the live matcher would honour. That last one calls the matcher the reader actually uses instead of re-deriving its rules, so the check cannot drift away from the thing it is checking.

Every one of the seven fails on an empty declaration, and each says why in its detail. A check that passed because it had nothing to look at is not evidence of anything, and the same principle applies to the exemption table: an empty table fails, because a table that is empty because none is needed and a table that was accidentally cleared look identical from inside the check.

The eighth check appends the wrapper’s own entry to the declaration, runs the ordinary reader over the result, and asks whether any of the wrapper’s own tokens turned up in any collision; it is stated as [Proposition 3](#). A declared exemption does not help here — the wrapper is not a line and has no standing to share a token with one, so an exempted collision involving it counts against it exactly like an unexempted one. The check needs the siblings importable, so it lives outside the offline battery, and when there is no sibling vocabulary to compare against it reports failure with the reason. Moving it into the offline battery would make an unattended run go green and would establish nothing.

2.4.6 Digests

Two canonical forms exist, both sorted before serialization so that no dictionary or set iteration order can reach them. The set digest covers the lines and the exemption table together. The reading digest covers a whole reading, including its date and derivation. Both are review handles for detecting that two people are looking at different things. Neither carries safety, warranty, or attestation semantics, and neither is tamper evidence.

2.5 The instrument stated formally

The method section says what the reader does, in the order it does it. This one restates the same machinery as objects and rules, so that a claim about the instrument can be checked against a statement rather than against a paragraph. Every statement below is written from the module it describes and is bound by a named test that re-derives it; where the code turned out narrower than a formal statement might be expected to be, the narrower thing is what is written down.

2.5.1 The declared objects

Definition 1 (Declared line). A declared line is a `LineEntry` record with ten fields: `id`, `color`, `question`, `job`, `must_not_become`, `opus_stage`, `working_position`, `package_name`, `registry_noun`, and `verdict_noun`. `opus_stage` may be absent, `working_position` is an integer, and every other field is text. A *declaration* is a tuple of such records paired with an exemption table. The reader, the structural checks, the serialization, and the figure builders each take that pair as an argument, and no module outside `registry.py` names an individual record.

Definition 2 (Declared exemption). A declared exemption is a `SharedToken` record with four fields: `token`, the spelling more than one line is allowed to carry; `lines`, the ids of the lines allowed to carry it; `meanings`, a sequence of line-and-meaning pairs; and `rationale`. The exemption table is a tuple of these records, and it is serialized together with the lines, so a set whose exemptions changed is a different set even when its lines did not.

The meanings field is the load-bearing one. A shared token with no meaning recorded per line is a label, and a label cannot show that two uses of one spelling are two different things — which is what [Definition 3](#) refuses to accept.

2.5.2 The exemption matcher

Definition 3 (Exemption match). Given an exemption table, a token, and the set of lines carrying that token, `exemption_for` returns a record when all of the following hold, and no match otherwise. The token is non-empty text. At least two lines carry it. Exactly one record in the table carries a `token` equal to it, character for character. That record's `lines` are text, non-blank, distinct, at least two, and equal *as a set* to the lines carrying the token. Its `meanings` supply a non-blank meaning for each line it names and name no line outside them. A multi-line token with a match is an **exempted** collision; one without a match is an **unexempted** collision.

Proposition 1 (Failing closed). Every condition in [Definition 3](#) is an equality, and every failure returns no match, which leaves the collision unexempted. Six weakenings are therefore refused: a query for a proper prefix of a declared token, a query for the token in another case, a query naming one line more than the record does, a record naming one line more than carries the token, a record omitting the meaning for a line it names, and two records declaring one token. What the refusals cost is asymmetric, and that asymmetry is the reason for the design: a missed exemption costs a visible `SET_COLLIDING` reading that a person then fixes by writing the declaration properly, while a generous match costs a guarantee that stopped holding without anyone being told.

[Proposition 1](#) names six weakenings, and the plate below is those six run through the matcher the reader actually calls, together with the declaration as written. The seventh row is the positive control: without it, six refusals would be equally consistent with a matcher that refuses everything.

2.5.3 The reading

Definition 4 (A reading). `read_set` takes a declaration of [Definition 1](#) records, an optional resolver, and a review date, and returns a reading that records five stages in order. **Resolve** asks the resolver for each line's `package_name` and records one of `RESOLVED`, `NOT_INSTALLED`, `IMPORT_FAILED`, or `NO_VOCABULARY`. **Bind** reads, from each package that resolved, a version string, a registry size, a digest that package computed itself, and the member names of every enum exported at its root; a line that did not resolve carries no version, no registry size, and no digest. **Collide** inverts the observations into a token-to-lines map, keeps every token carried by more than one line, and partitions those by [Definition 3](#) over the [Definition 2](#) table. **Declare** names packages that resolved and appear in no entry, or records that the resolver could not be asked. **Status** applies [Proposition 2](#). The self-application check ([Proposition 3](#)) appends the wrapper's own entry and takes an ordinary reading through the same five stages.

Proposition 2 (Status precedence). A reading's status is the first of `SET_COLLIDING`, `SET_UNDECLARED`, `SET_PARTIAL`, `SET_LEGIBLE` whose condition holds: an unexempted collision, then a resolved package that is not declared, then a declared line whose read code is not `RESOLVED`, then nothing else. `SET_LEGIBLE` is reachable only when no condition above it matched, so a live collision cannot be hidden behind a missing install and a partial installation cannot present itself as a clean reading.

2.5.4 The self-application

Proposition 3 (Self-disjointness). `check_self_disjointness` appends the wrapper's own entry to the declaration, takes an ordinary reading of the result by [Definition 4](#), and holds only when four things are true together: the wrapper's own vocabulary was read; at least one other line supplied a vocabulary; no collision in that reading names the wrapper; and no other declared

THE EXEMPTION GATE

What a declared exemption has to survive

Each row weakens the declaration or the query one way and reports what the matcher the reader uses returned for it.

7 PROBES · 1 HONoured · 6 REFUSED · 0 DIVERGED FROM SPECIFICATION

☒ declared the exemption as declared, asked for exactly the lines it names table: 1 declaration · OUTSIDE_SCOPE over red_line, black_line · 2 meanings	exempted as specified asked: OUTSIDE_SCOPE over red_line, black_line
☐ truncated token a proper prefix of the token; no substring test table: 1 declaration · OUTSIDE_SCOPE over red_line, black_line · 2 meanings	left unexempted as specified asked: OUTSIDE_SCOP over red_line, black_line
☐ case-folded token the token in another case; no case folding table: 1 declaration · OUTSIDE_SCOPE over red_line, black_line · 2 meanings	left unexempted as specified asked: outside_scope over red_line, black_line
☐ a third line carries it one more line carries the token than the exemption names; no subset test table: 1 declaration · OUTSIDE_SCOPE over red_line, black_line · 2 meanings	left unexempted as specified asked: OUTSIDE_SCOPE over red_line, black_line, golden_line
☐ declaration names a line that does not carry it the exemption names one line more than carries the token; no superset test table: 1 declaration · OUTSIDE_SCOPE over red_line, black_line, golden_line · 2 meanings	left unexempted as specified asked: OUTSIDE_SCOPE over red_line, black_line
☐ a named line has no meaning a line is named with no meaning recorded for it table: 1 declaration · OUTSIDE_SCOPE over red_line, black_line · 1 meaning	left unexempted as specified asked: OUTSIDE_SCOPE over red_line, black_line
☐ two declarations of one token the same token declared twice; an ambiguous exemption exempts nothing table: 2 declarations · OUTSIDE_SCOPE over red_line, black_line · 2 meanings	left unexempted as specified asked: OUTSIDE_SCOPE over red_line, black_line
every probe agreed with its specification 1 probe honoured and 6 refused, all decided by the matcher the reader calls rather than by a second copy of its rules.	

READING RULE

A refusal leaves the collision standing, which is the cheap direction: a missed exemption costs a visible colliding reading that a person then fixes by writing the declaration properly, while a generous match costs a guarantee that stopped holding without anyone being told. What this plate shows is detection over the weakenings named on it, and nothing about weakenings nobody wrote down.

Figure 5: The exemption gate. Each row weakens the declaration or the query in one way and reports what `exemption_for` returned for it; the outcome column is a return value, not a claim written into the plate. The first row is the declaration as shipped, asked for exactly the lines it names, and it must be honoured. A refused row leaves its collision standing.

line went unread. An exempted collision naming the wrapper counts against it exactly as an unexempted one does, because the wrapper is not a line and has no standing to share a token with one.

The middle two conditions are what keep the property from being established by an empty comparison. They are also why the check cannot live in the offline battery: it needs the sibling packages importable, so on a machine without them it reports failure with the reason rather than a pass.

Proposition 4 (Vocabulary disjointness). The reading (see [Definition 4](#)) partitions every token into the lines that carry it. Each vocabulary term maps to exactly one registry entry: no term is ambiguous across the line set. After exempted collisions are removed by [Definition 3](#), any token appearing in more than one vocabulary is an unexempted collision, and the reading’s status is `SET_COLLIDING`. The set’s contract is that every terminal spelling names exactly one registry entry; non-overlap of vocabulary is a necessary condition for separation and nowhere near a sufficient one — two lines can be about substantially the same thing in different words and read `SET_LEGIBLE`, and two lines can be entirely distinct in substance and collide because they both liked a word.

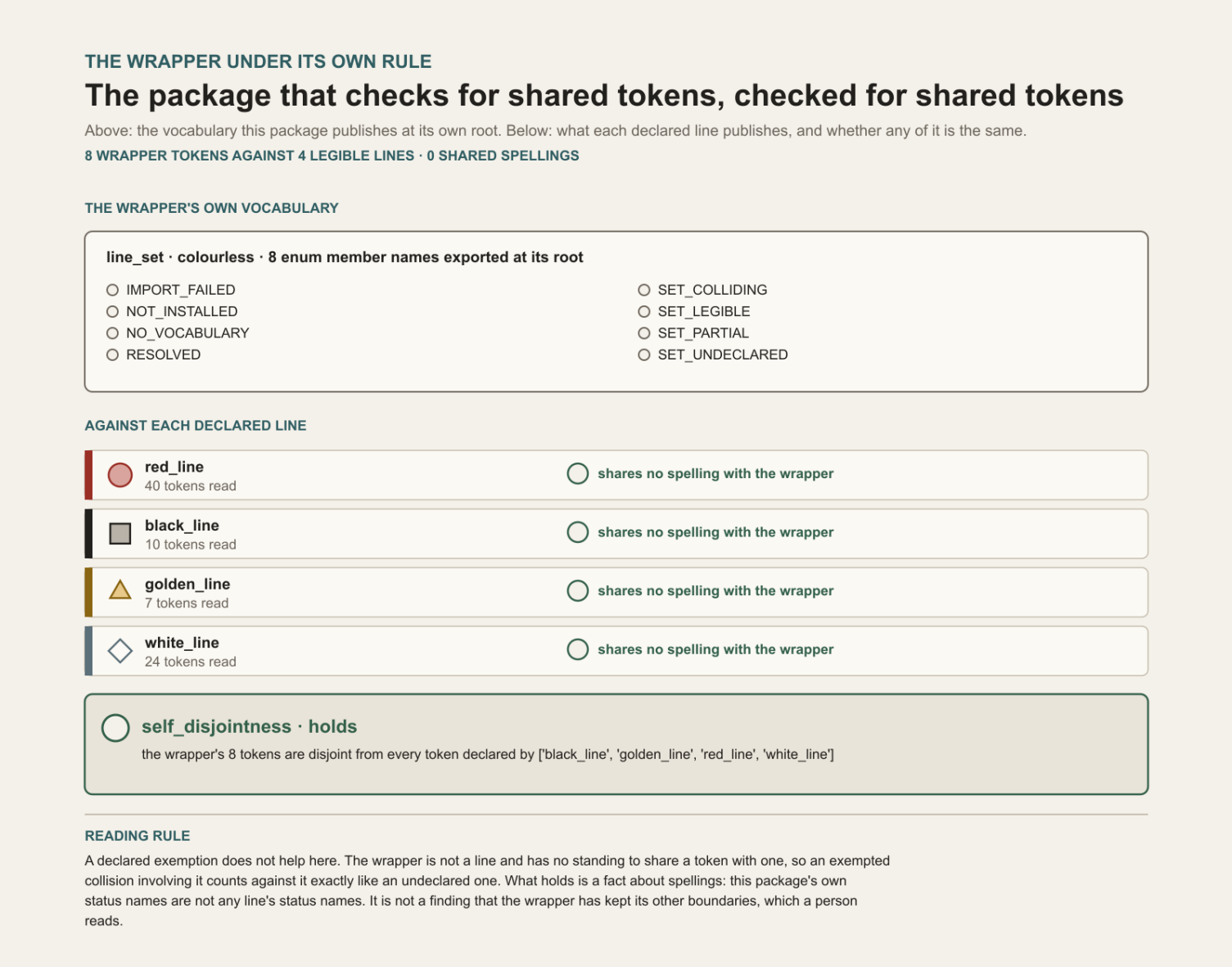


Figure 6: The wrapper under its own rule. Above, the enum member names this package publishes at its own root; below, one row per declared line with the vocabulary it published and whether any spelling is shared. The verdict panel prints the check’s own result and detail over this reading, so the plate and the check cannot disagree.

2.5.5 What the statements do not carry

They are statements about this package, and each is exactly as strong as the function it describes. [Proposition 2](#) says which status a reading takes, not that the status is deserved. [Proposition 1](#) says the matcher refuses six named weakenings, not that no seventh

would be accepted; the probes demonstrate detection over the inputs written down, which is the most a demonstration can do. **Proposition 3** is about spellings in namespaces and says nothing about whether this package has stayed out of the siblings' business — that boundary is prose, and a person checks it.

2.6 Scholarship: what is old about this problem, and what my check is not

Keeping parts of a system from becoming each other is not a new problem, and I did not solve it. The literature below gave me the shape of the failure and a list of things to be afraid of. None of it validates the declaration, the reader, or the collision check, and none of the cited work studies this package or anything like it. I am reporting where the design came from, not borrowing authority for it.

2.6.1 Modules defined by what they hide

Parnas argued that a system should be decomposed by design decisions rather than by processing steps, and that a module's value lies in what it keeps from the rest of the system — the decision it hides, so that changing it does not propagate [Parnas, 1972]. That is the argument the line set is organized around. Red Line's classification logic, Black Line's practice registry, Golden Line's horizon rules, and White Line's staleness contract are all decisions each line owns and none of the others should have to know. The published vocabulary is the interface; the evaluator is not. The distance between that argument and what this package checks is worth stating precisely, because it is large. Parnas is concerned with which decisions are hidden and whether a change stays local. My reader collects the enum member names a package exports at its root and asks whether any two lines spell something the same. Those names are a proxy for the published interface, and a coarse one: a line could hide nothing, expose its entire internal state through functions rather than enums, and still read as perfectly disjoint here. Information hiding is a property of what a module keeps back. A name collision is a property of what two modules happen to say out loud. The second is checkable in a hundred lines and the first is not, which is why I checked the second and am saying so rather than letting the citation imply otherwise.

Thirteen years later Parnas, Clements, and Weiss reported what it took to keep a real decomposition legible, and their answer was a second artifact: a *module guide*, a document naming each module and the secret it holds, maintained deliberately because the decomposition is not recoverable from the code that implements it [Parnas et al., 1985]. `LINE_SET` is a module guide of the smallest possible kind, one record per line (Definition 1). Their paper is also the warning I took most seriously, because a guide is a second statement of the structure and a second statement can disagree with the first. Their remedy is review by people who know the system. Mine is narrower and mechanical: the entry that names a package is the entry the reader imports, so a guide describing a package that is not there produces `SET_PARTIAL` rather than a sentence nobody reread. The interesting half still has no mechanism. Nothing checks that a line's declared job is its actual job, and nothing checks that the secret each line holds is still hidden.

Dijkstra's essay names the discipline the set is trying to practise: study one aspect at a time, in the knowledge that the other aspects remain true and will have to be faced [Dijkstra, 1982]. He is careful that separating concerns is not the same as neglecting the ones set aside. The set inherits that caution structurally — Red Line's refusal keeps applying while I am working inside Black Line, and a `TOWARD` reading in Golden Line cannot authorize anything Red Line refuses. What this package adds is small: it checks that the vocabularies stayed apart. It does not check that the concerns did.

Conway observed that a system's structure tends to reproduce the communication structure of the organization that built it [Conway, 1968]. The line set inverts the premise rather than illustrating it. There is no organization here; there is one person and four repositories, which means no team boundary, no handoff, and no separate maintainer doing the separating for me. Whatever separation exists has to be written into a declaration and checked mechanically, precisely because the social structure that would ordinarily produce it is absent. I take Conway as a diagnosis of why a single-author set needs an explicit contract, not as evidence that this one works.

2.6.2 Prefixes, and what they are worth without an authority

The oldest mechanical answer to a name collision is to qualify the name. XML namespaces bind a short prefix to a URI so that names minted by different authorities cannot collide even when they spell the same word, and the URI is what does the work: it is owned, so two authors cannot mint the same one by accident [Bray et al., 1999]. The `SET_` prefix on this package's reading statuses is that move performed by hand, at the smallest scale I could get away with, and Proposition 3 is what it buys.

Where the transfer stops is exactly where the authority would have been. There is no URI, no registry, and nothing preventing a sibling from adopting the same prefix tomorrow. A namespace makes collisions impossible by construction; a convention makes them *visible*, one review at a time, and only on a machine where every line is installed. That is the weaker guarantee, and it is the one available to a set of packages nobody governs.

2.6.3 Coordinating without merging

Star and Griesemer described boundary objects: artifacts “plastic enough to adapt to local needs” in several communities while remaining “robust enough to maintain a common identity across sites,” which lets groups cooperate without agreeing [Star and Griesemer, 1989]. The shared token (Definition 2) has that shape. `OUTSIDE_SCOPE` carries one spelling and two local meanings, and the exemption table records both rather than forcing either line to give the word up.

Star’s companion paper of the same year is the more useful one here, because it is the one with a typology: repositories, ideal types, coincident boundaries, and standardised forms [Star, 1989]. The exemption table is a standardised form — a fixed shape, filled in the same way each time, whose function is to carry a local meaning across a boundary without negotiating it away. Reading it that way is what stopped me treating the shared token as a defect awaiting a rename.

Star later objected to how the concept had been taken up, in particular to its application to any object that happens to sit between two groups, detached from the infrastructural and standardizing work the original study was about [Star, 2010]. Her objection applies to me, so I will not make the claim. Two Python packages in one person’s working tree are not two communities of practice; there is no negotiation between amateur collectors and professional zoologists here, no institutional ecology, and no cooperation across divergent viewpoints, because the divergent viewpoints are mine. What I took from the 1989 papers is a design move: a shared term can be kept, with its local meanings written down, instead of being resolved by renaming. That is a resemblance I found useful. It is not a case study, and the concept is not doing evidentiary work in this package.

2.6.4 The hazard of the index

The specific danger of building the fifth work is that an instrument which reads the other four becomes the place people look, and then the other four start being written to read well in it.

Bowker and Star show how classification systems become infrastructure: once embedded, they stop being visible as choices, and the categories begin to organize the work rather than describe it, with the things that fit badly quietly pushed into residual bins [Bowker and Star, 1999]. A declaration listing four lines with a colour, a stage, and a position is a classification system, however small. Strathern’s account of audit makes the same point about instruments of assessment specifically — a mechanism introduced to describe a practice can end up reorganizing that practice around the mechanism’s own categories [Strathern, 1997a]. Neither argument is about a hundred-line Python package, and I am not claiming a package this size carries that weight. The mechanism is what I am watching for, at the scale I actually have.

The design responses are modest and I would rather name them than promise around them. There is no score, no aggregate, and no ordering by merit; the reader emits a status about the set and never a judgement about a line. The wrapper’s own entry is colourless and carries no stage, so it cannot be read as a fifth position in the sequence. The entry declares what the wrapper must not become in the same field every line uses for the same purpose, which puts the wrapper under the same discipline it applies to the others. And the wrapper’s own vocabulary is run through the wrapper’s own collision check, so the package cannot exempt itself from the one rule it enforces.

None of that prevents the drift. Bowker and Star’s point is that infrastructure becomes invisible, and a package that has become invisible is exactly the one nobody re-reads. What these design choices buy is that the drift, if it happens, has to happen in the open: someone would have to add a ranking field, or an aggregate, or an exemption for the wrapper, and each of those is a visible edit to a small file. Visibility is a weaker guarantee than prevention. It is the one I can actually implement.

2.7 Extensibility: adding a colour is an edit to one file

The set has grown before and will grow again, so the cost of adding a line is a design property rather than a convenience. The target is that a new colour costs one appended record in `registry.py` (Definition 1) and nothing else: no branch in the reader, no case in a check, no new figure code.

Parnas treats extension and contraction as one problem and puts the cost of both in the *uses* relation rather than in the size of the edit: what makes a subset removable, or an addition cheap, is that nothing outside it names it [Parnas, 1979]. That is the property claimed here, and it is claimed narrowly. He is designing families of programs whose minimal subsets are chosen in advance and whose uses relation is documented and enforced; I have one package, and the only uses relation I have measured is which modules mention a line id as a whole word. Those are different standards of evidence for the same-shaped claim, and the second is the one this section supports.

The claim holds because no module outside `registry.py` names an individual line. Every function that does work takes `lines` and `shared` as arguments and computes over whatever it was handed. Searching the package for the four line ids as whole words finds them only in `registry.py`, four and four and two and two times; `reader.py`, `invariants.py`, `binding.py`, `serialization.py`, `models.py`, and `__init__.py` contain none of them. One mention of `OUTSIDE_SCOPE` survives in `reader.py`, inside the exemption matcher's docstring, where it is the worked example of a substring the matcher must refuse. It is prose, not a branch.

2.7.1 The executed example

Here is a fifth colour, appended at runtime and put through the whole battery. Nothing in the package was edited to run it.

```
from line_set import (LINE_SET, SHARED_TOKENS, LineEntry, all_invariants,
                      read_set, registry_digest, sibling_path_resolver)

fifth = LineEntry(
    id="green_line",
    color="green",
    question="What does keeping this alive cost?",
    job="Standing upkeep, dependencies, and the bill for continued existence",
    must_not_become="A reason to drop work that is merely expensive",
    opus_stage=None,
    working_position=5,
    package_name="green_line",
    registry_noun="upkeep records",
    verdict_noun="upkeep status",
)
extended = LINE_SET + (fifth,)

print("lines:", len(LINE_SET), "->", len(extended))
print("digest:", registry_digest(LINE_SET, SHARED_TOKENS)[:12],
      "->", registry_digest(extended, SHARED_TOKENS)[:12])
for r in all_invariants(extended, SHARED_TOKENS):
    print(f" {'PASS' if r.passed else 'FAIL'} {r.name}")
reading = read_set(extended, SHARED_TOKENS,
                   resolver=sibling_path_resolver(), as_of="2026-07-29")
print("status:", reading.status.value)
print("counts:", reading.counts())
print("reason:", reading.derivation[-1].detail)
```

The measured output:

```
lines: 4 -> 5
digest: 40db5e0e3e03 -> 7ca600e34332
PASS distinct_line_ids
PASS distinct_colours
PASS distinct_opus_stages
PASS contiguous_working_positions
PASS orders_diverge
PASS must_not_become_declared
PASS shared_tokens_disambiguated
status: set_partial
```

```
counts: {'resolved': 4, 'not_installed': 1, 'import_failed': 0, 'no_vocabulary': 0}
reason: set_partial: 1 declared line(s) could not be read: ['green_line']
```

All seven structural checks pass on the five-entry declaration. Position contiguity now expects 1..5 and gets it. Colour distinctness has a fifth colour to consider. The orders still diverge, because the fifth entry declares no opus stage and is therefore ignored by that comparison — a line may join the set without being assigned a stage, and a set could in principle grow past four while the four borrowed stage names stay exactly four.

The wrapper’s own collision check ([Proposition 3](#)) applies to any extended declaration.

The reading is `SET_PARTIAL`, and that is the correct answer rather than a shortfall. I declared a line whose package does not exist on this machine, and the reader said so, named it, and left its version, registry size, and digest unset. It did not invent a version, and it did not quietly drop an entry it could not resolve. A declaration that runs ahead of an installation is an ordinary state — it is what the first commit of a new line looks like — and the reader is built to report it rather than to fail.

2.7.2 What the digest change means

The set digest moved from 40db5e0e3e03 to 7ca600e34332. That is the intended behaviour: the digest covers the lines and the exemption table together, so a set with a fifth line is a different set. It is also worth noticing that the digest binds to the *content* of the entry and not merely to its presence. Rewording the fifth line’s question produces a different digest, which is what makes the value useful for spotting that two people are reading different declarations.

The value is a comparison handle and nothing more. It tells you that a declaration you hold differs from a declaration someone else holds. It does not tell you which one is right, it does not record who changed what, and it is not tamper evidence.

2.7.3 What this does not establish

What the run above shows is that the declaration is extensible without touching the reader or the checks: the battery passed on five entries, the reading was taken, and no module outside `registry.py` was edited to make either happen. Two nearby claims are not shown and should not be read in.

Adding a colour is cheap in code. It is not cheap in judgement: the hard part of a fifth line is deciding whether the set actually has a fifth question, and nothing in this package can help with that. A line whose question overlaps an existing line’s would pass every check here, because the checks read names.

The same requirement is placed on the figure builders — they draw from the declaration rather than from a fixed list of four, and a test appends a colour and re-renders to hold them to it. A passing render is a weaker result than it sounds. It says the plate was produced, not that it is still readable at five entries, or at nine.

2.8 Worked readings

Every reading below was executed against the package as it stands. Where a reading needed a package that does not exist, I built one in a temporary directory and pointed a resolver at it; there is no mocking framework anywhere in this project, and a resolver is a plain callable passed as an argument.

2.8.1 The live set

```
from line_set import read_set, reading_digest, sibling_path_resolver

reading = read_set(resolver=sibling_path_resolver(), as_of="2026-07-29")

status          : set_legible
set_digest      : 40db5e0e3e038707
reading_digest  : e783001a0baa1c16
counts         : {'resolved': 4, 'not_installed': 0, 'import_failed': 0, 'no_vocabulary': 0}
  red_line      resolved v0.3.0 registry= 7 digest=72835fd81d1f tokens=40
  black_line    resolved v0.4.0 registry=11 digest=a02bff47a767 tokens=10
  golden_line   resolved v0.4.0 registry= 9 digest=3e0a7e38ecec tokens= 7
  white_line    resolved v0.7.0 registry=11 digest=11047a641a8e tokens=24
  exempted:     OUTSIDE_SCOPE over ['black_line', 'red_line']
  unexempted:   []
[resolve] asked the resolver for 4 declared packages
[bind] read version, registry size, digest, and enum member names from each package that resolved
[collide] 0 unexempted and 1 exempted cross-line token collisions
[declare] scanned 4 candidate packages and found 0 that resolved but were not declared
[status] set_legible: all 4 declared line(s) were read and their vocabularies do not overlap
```

The four packages export 19 enum classes between them — 7 in `red_line`, 3 in `black_line`, 2 in `golden_line`, 7 in `white_line` — and those classes declare 89 members in total. Eight of the 89 are a line repeating a word it already uses in another of its own enums, which is not a cross-line event and which the reader collapses: `red_line` spells `OUTSIDE_SCOPE` in two of its enums, `black_line` repeats three names across its three, and `white_line` repeats four names across its seven — its witness-facet alphabets reuse states its ledger already declares. After that collapse the reading holds 81 line-and-name pairs — 40 for `red_line`, 10 for `black_line`, 7 for `golden_line`, 24 for `white_line` — spanning 80 distinct names.

The arithmetic is the whole finding: 81 minus 80 is one, the one is `OUTSIDE_SCOPE`, and the set had already declared it (governed by [Proposition 2](#)).

The version and registry numbers above are dated. They are what the installed siblings reported on the review date, not constants of the set, and a sibling release changes them without changing anything this package claims. The digests are there so a later reader can tell whether they are looking at the same four packages I was.

2.8.2 A second line adopts the word

Suppose a fifth line joins the set and reuses `OUTSIDE_SCOPE`. Written as a real package in a temporary directory and declared as a fifth entry:

```
status: set_colliding
  unexempted: OUTSIDE_SCOPE over ('black_line', 'red_line', 'teal_line')
[status] set_colliding: 1 cross-line token collision(s) are not covered by a declared exemption
```

The existing exemption did not stretch to cover it. It names exactly two lines, three lines now carry the token, and [Definition 3](#) requires set equality rather than a subset test, so the exemption stops applying to a collision it no longer describes. This is the behaviour I most wanted to see and least wanted to assume: an exemption written for two parties does not silently license a third. Making the reading legible again means writing a new declaration that says what the word means in the third line, or renaming.

2.8.3 A package nobody declared

The same temporary package, present and importable but absent from the declaration:

```
status: set_undeclared
  undeclared_lines: ('teal_line',)
[declare] scanned 5 candidate packages and found 1 that resolved but were not declared
```

THE VOCABULARY MATRIX

Which line spells which token

Tokens are the enum member names each package exports from its root. A name defined but not re-exported is not here.

LIVE READING · 80 TOKENS ACROSS 4 LEGIBLE LINES

 red 40 tokens	 black 10 tokens	 golden 7 tokens	 white 24 tokens
---	---	---	---

TOKENS CARRIED BY MORE THAN ONE LINE · 0 undeclared · 1 declared

 OUTSIDE_SCOPE EXEMPT — declared, with one meaning per line	 red · carries	 black · carries	 golden · does not	 white · does not
--	--	--	--	---

THE FULL GRID · filled cell = that line exports that token

ABSOLUTE					MULTIPLE_PROHIBITED_DIMENSIONS				
AFFECTED_PARTIES					NAMED				
AIR_GAPPED					NEEDS_EVIDENCE				
ALIGNED					NEEDS_REWORK				
ALL					NON_COMPLIANT				
ALL_LINES_NARROWED					NON_ITERABLE_INPUT				
AMBIGUOUS_INTAKE					NON_OBSERVATION				
ANY					NOT_OBSERVED				
BELOW_TIER_FLOOR					NOT_RECORDED				
CANARY					OPEN				
CAPABILITY_SCOPE					 OUTSIDE_SCOPE				
COMMUNICATION					PURPOSE				
COMPLIANT					RECORD_OR_EXPLICITLY_SCOPE				
CONFLICTING_STATES					REOPEN_OR_BOUND				
CONNECTED					REQUIRES_MODIFICATION				
CONTEMPLATIVE					RESERVED_STATE				
CONTRADICTED					RETAIN_NAMED_SCOPE				
CURRENT					SELF_ASSERTED				
DATA_PROVENANCE					STALE				
DEPLOYMENT					STALE_EVIDENCE				
DOWNSTREAM_TRANSFER					STEWARDSHIP				
DRIFTING					STRONG				
DUPLICATE					TOWARD				
DUPLICATE_STATE					TRACEABILITY				
EMPTY_SCOPE					UNDATED				
END_USE					UNDATED_OBJECT				
EPISTEMIC					UNDECLARED_UNKNOWN				
ETHICAL					UNDISCLOSED				
FRAMING					UNEXEMPTED_LINE				
HONOR_AND_CONFIRM_BOUNDARY					UNKNOWN_ID				
HOSTED					UNKNOWN_RECORD				
HUMAN_CONTROL					UNKNOWN_SCOPE				
INQUIRY					UNRESOLVED				
INSUFFICIENT_INFORMATION					UNRESOLVED_EVIDENCE				
INTAKE_BLOCKED					UNTYPED_STATE				
INVALID_SCOPE					UNVERIFIED				
LEGAL_BASIS					VERIFICATION				
MALFORMED					VERIFIED				
METHOD					VERIFIED_EXEMPTION				
MISSING_EVIDENCE					WITHHELD				

READING RULE

A grid with no undeclared collision says the declared names do not overlap. It does not say the lines do not overlap: two lines can spell nothing alike and still be asking the same question. A declared exemption records that one spelling carries two different senses; it is not permission to merge them.

Cells are drawn from this build's reading; a line that was not read contributes no column content and is marked in the key above.

Figure 7: Vocabulary matrix: exported enum member names down the side, declared lines across the top. A cell is marked where a line carries that token. The one token carried by two lines is marked as a collision, and the exempted cell is distinguished from an unexempted one by both its shape and its label, so the plate reads in greyscale.

Discovery here is convention-based: the resolver looks for importable top-level packages whose names end in `_line`, plus anything already in `sys.modules`. That finds a package following the naming convention and cannot find one that does not. An empty `undeclared_lines` is a report of what the convention turned up, never a proof that nothing else exists. When a resolver offers no enumeration at all, the derivation says the scan did not happen instead of reporting zero.

2.8.4 Lines that will not read

Two more temporary packages, one that raises on import and one that imports cleanly while exporting no enum:

```
status: set_partial
  red_line      import_failed  version=None  :: import raised RuntimeError: this line does not import
  black_line    no_vocabulary  version=9.9.9 :: the package root exports no enum
  golden_line   not_installed   version=None  :: the import system found no such package
  white_line    not_installed   version=None  :: the import system found no such package
```

All four read codes appear at once. The one worth pointing at is `black_line`: it imported, so its real version string was read and kept, but it published no vocabulary, so there was nothing to compare and no registry to size. The reader neither discarded the version it did have nor filled in the two it did not.

2.8.5 No siblings at all

Run from a directory where none of the four packages is importable:

```
registry_sound()          : True
offline checks passing    : 7 / 7
read_set()                : set_partial {'resolved': 0, 'not_installed': 4, ...}
every version/size/digest None: True
check_self_disjointness() : FAIL - no line vocabulary was available to compare
                           against, so the comparison set was empty;
                           unread lines: ['black_line', 'golden_line',
                           'red_line', 'white_line']
```

The seven offline checks pass, because they are computation over the declaration and need nothing installed. The reading is honestly partial. And self-disjointness fails rather than passing, which is the point of keeping it out of the offline battery: an overlap check that found no overlap because it had nothing to look at is not a result, and reporting it as a pass would be the most convenient lie this package could tell.

2.8.6 A weakened exemption

The last reading is the one that checks the check. Replacing the shipped exemption with one that names both lines but records a meaning for only the first:

```
exemption_for(...) -> None
status              : set_colliding
  unexempted: OUTSIDE_SCOPE over ('black_line', 'red_line')
```

The declaration still names the token and still names both lines. It is refused anyway, because an exemption without a meaning for each line it names is a label, and a label does not show that two uses of one spelling are two different things. The same refusal covers every other weakening in [Proposition 1](#), and no match leaves the collision standing.

THE INSTALLATION SURFACE

What this machine could actually read

One row per declared line: whether it resolved, what version it publishes, how large its registry is, and the prefix of its digest.

LIVE READING · 4 DECLARED LINES · 0 LINES NOT READ

	red red_line digest read from registry_hash()	 resolved imported, and its exported enum members were read	VERSION 0.3.0	REGISTRY SIZE 7	TOKENS 40	DIGEST 72835fd81d1f
	black black_line digest read from registry_digest()	 resolved imported, and its exported enum members were read	VERSION 0.4.0	REGISTRY SIZE 11	TOKENS 10	DIGEST a02bff47a767
	golden golden_line digest read from registry_digest()	 resolved imported, and its exported enum members were read	VERSION 0.4.0	REGISTRY SIZE 9	TOKENS 7	DIGEST 3e0a7e38ecec
	white white_line digest read from registry_digest()	 resolved imported, and its exported enum members were read	VERSION 0.7.0	REGISTRY SIZE 11	TOKENS 24	DIGEST 11047a641a8e
	set_legible every declared line was read and no two share a token resolved 4 · not_installed 0 · import_failed 0 · no_vocabulary 0 declaration digest 40db5e0e3e03 · undeclared lines: none reported scanned 4 candidate packages and found 0 that resolved but were not declared					

A row marked not read is a fact about this machine's installation, not about the line. Nothing on that row is filled in from memory, a default, or a previous build.

Figure 8: Installation surface: one row per declared line, showing whether it resolved, the version it reported, its registry size, and the first characters of its digest. Unresolved lines render as visibly empty rather than as zero, because a line that could not be read has no size, not a size of none.

2.9 Limits and Epistemic Boundaries

2.9.1 The reader reads declarations

Everything this package reports is derived from what the sibling packages say about themselves at their package roots: a version string, a registry tuple, a digest function, and the member names of the enums they export. Nothing is executed. No evaluator is called, no data is passed to a line, and no finding a line has ever produced is examined. A line could return the same verdict for every input it is given and read as perfectly healthy here, because the reader never asks it a question.

The consequence is that `SET_LEGIBLE`, in [Proposition 2](#), is a statement about names in four namespaces on one machine on one date. It is not a statement that the instruments work, that they are internally consistent, that their tests pass, or that their registries are current.

2.9.2 Disjoint vocabularies are not disjoint concepts

This is the limit I most want a reader to carry away, because it is the one the instrument’s own name invites people to forget. The check finds tokens two lines spell identically. It cannot find two lines that answer the same question in different words.

Nothing would stop me from writing a fifth line whose registry duplicates Golden Line’s aspirations under new names, whose evaluator reimplements Golden Line’s decision rule, and whose statuses are spelled so as to collide with nothing. The reading would be `SET_LEGIBLE`. It would be correct and it would be worthless, because the separation I actually care about is conceptual and the property I can check is lexical. The lexical check is worth having — it catches the failure that makes stored findings ambiguous downstream — and it is a necessary condition for the separation, not evidence of it. Whether two lines are thinking about the same thing remains a judgement someone has to make by reading them.

2.9.3 A digest is not tamper evidence

The set digest and the reading digest exist so that two people, or one person at two times, can tell whether they are looking at the same declaration or the same reading. A mismatch is a prompt to go and find out what changed. That is the entire semantics.

A digest carries no safety, warranty, attestation, or authorship claim. It detects disagreement with a digest someone already holds; it does not detect tampering, because anyone who can change the declaration can recompute the digest, and there is no signature, no key, and no independent witness anywhere in this package. Each sibling’s digest is likewise the sibling’s own value, computed by its own function over its own registry, and this package never derives one on a sibling’s behalf — a number the wrapper computed would say nothing about whether the sibling agrees with it.

2.9.4 The reader cannot say whether a line is worth having

There is no score here, no aggregate, no ranking, and nothing that could be turned into one without an obvious edit. A reading says whether the declared vocabularies overlapped and which lines could be read. It has no view on whether Red Line’s question is well posed, whether White Line earns its place, whether four is the right number, or whether any of these instruments has ever improved a piece of work.

That silence is deliberate rather than an omission awaiting a future release. The lines are peers, the colours are labels rather than a ladder, and an instrument sitting above four others is exactly the wrong place to start assigning merit. The wrapper’s own declaration records that as the thing it must not become.

2.9.5 The reader cannot see a missing colour

Nothing in this package can tell whether the set is incomplete. It reads what was declared and looks for installed packages whose names follow a convention; a question nobody has asked yet leaves no trace in either. The set could be missing an instrument for what work costs to maintain, or for who else is affected, or for something I have not thought of, and every check here would keep passing.

Discovery is also weaker than it may appear from a passing run. It finds importable top-level packages whose names end in `_line`, plus what is already imported in the process. A line package named otherwise is invisible to it. An empty `undeclared_lines` is a report of what that convention turned up on that machine, and never a proof that nothing else exists.

2.9.6 Smaller boundaries worth naming

Only the package root is read for enums. A token defined in a submodule and never re-exported does not participate in the collision scan. That is a deliberate scoping of the contract to the vocabulary a line publishes, and it does mean two lines could collide privately without this reader noticing.

Registry size comes from a convention: the unique public, non-empty tuple in the package's **registry** submodule whose members are all one dataclass type, largest winning, ties reporting nothing. A package that organizes itself differently reports no size, which is an absence of information rather than a finding about the package.

Self-disjointness needs the siblings importable and fails when they are not — the second and third conditions of **Proposition 3** are what make it fail rather than pass. That failure is honest and it is also a real coverage limit: on a machine with no siblings, the property this project exists to hold is unestablished, not established.

Each structural check is backed by a test that plants its defined bad input and proves the check rejects it. What that establishes is detection of the planted defect. It is not evidence that no other defect exists in the declaration, and it says nothing at all about the four packages the declaration describes.

Finally, the colours. The alchemical resonance is a naming resonance held in Jung's symbolic register and nothing further [**Jung, 1953**]. No line is a stage of anyone's transformation, the working order does not re-enact the opus, and the borrowed names carry none of the opus's direction. A structural check pins the divergence of the two orders; nothing pins how a reader will hear the names, and the honest safeguard there is to keep saying what the echo is not.

2.10 Conclusion

The four lines were separate because I said they were. They are still separate for the same reason, and now one narrow part of that claim is checkable: on the review date, the four packages between them put 81 line-and-name pairs in front of the reader, spanning 80 distinct names, and the single name carried by two lines was the one the set had already written down and disambiguated.

That is a small result and I have tried not to dress it up. A necessary condition held. Whether the instruments overlap is the question I actually care about, no reader of names can answer it, and it is still a matter of reading the four manuscripts and deciding (see [the scholarship section](#)).

What the project earns beyond the reading is the shape of its own constraint. A package that checks other packages for shared tokens has to survive its own rule, or the rule is something it imposes rather than something it holds. Prefixing the reading statuses and then running the wrapper's entry through the wrapper's own collision check closes that gap cheaply, and it forced one real design decision: the check needs the siblings installed, so it cannot live in the offline battery, so an unattended run reports it as failing rather than passing. I would rather have a check that says *unestablished* on a bare machine than one that says *fine* because it had nothing to compare. Writing that condition down as [Proposition 3](#), with a test that re-derives it, is what keeps the sentence and the function from drifting apart.

The fifth work adds no instrument. Red Line still refuses, Black Line still builds, Golden Line still reaches, and White Line still marks the edge; this one holds the declaration of what they are, reads whichever of them is present, and says whether any two started using the same word for different things. Adding a sixth colour costs an appended record and no other edit (see [the extensibility section](#)), which is the property I wanted most, because a set that is expensive to extend stops being extended and starts being stretched.

The work is built for DAF's public research index at [docxology](#) — machine-readable, cross-linked, and verification-logged — with its eventual public home at [docxology/line_set](#). The set's short orientation note, [docs/line-set.md](#), remains an unpublished working note in the author's private projects tree and does not travel with this repository, so it is cited by name rather than linked; this paper is the version with a package under it and a reading attached.

3 Personal Red Lines for Development

An Evidence-Gated Personal Security Boundary and Explicit No Document for Dual-Use Development

Reproduced unchanged in substance from its own source at version 0.3.0. It answers one question: What must I refuse?

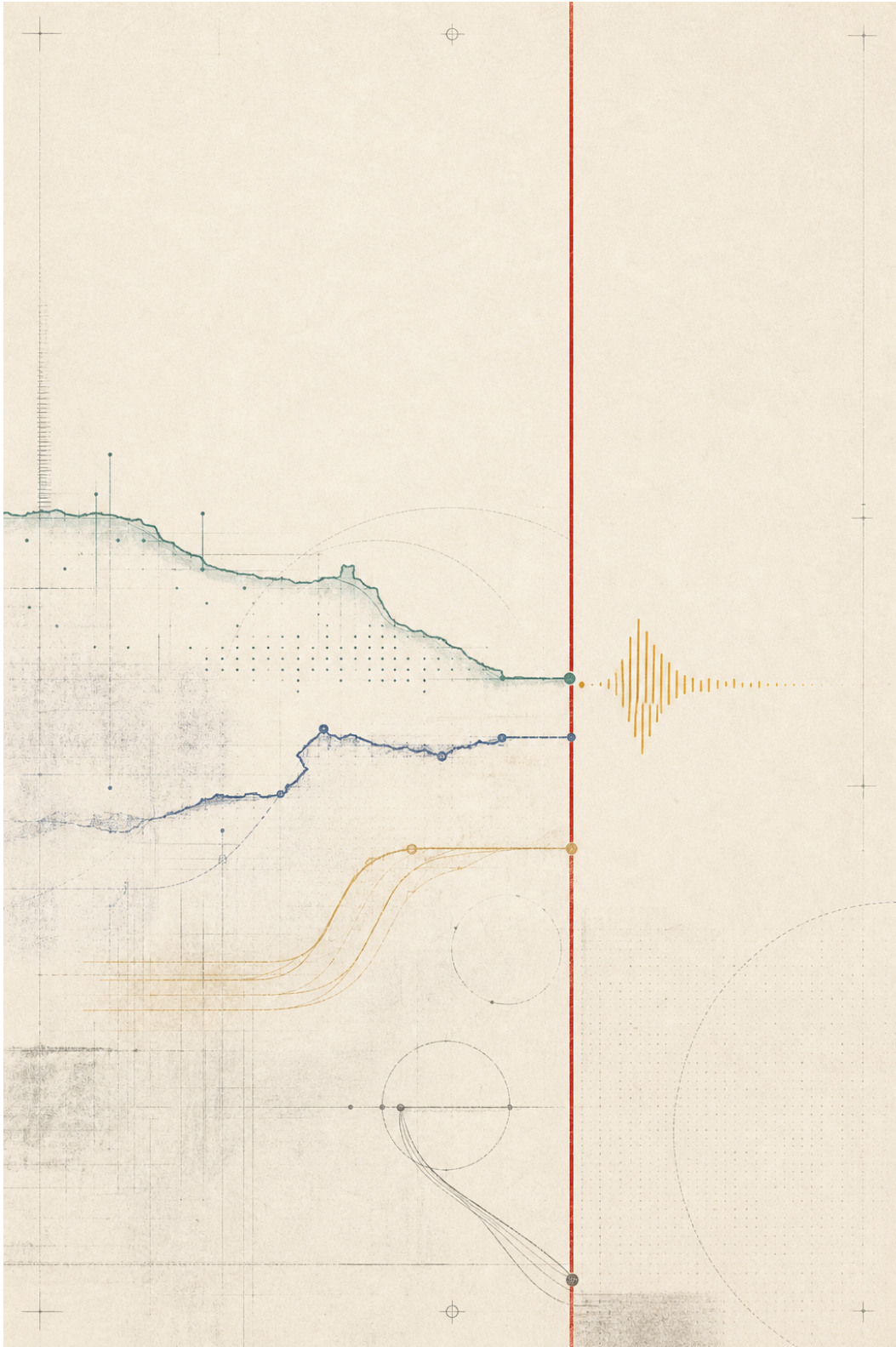


Figure 9: Cover art for Personal Red Lines for Development

3.1 Abstract

Red Line is a personal security boundary and explicit **No document** for dual-use development. Its function is deliberately narrow: help one practitioner refuse work before commitment, make a near-boundary review inspectable, and make material weakening visible over time. It records seven first-person refusals as a versioned, machine-readable registry and subjects a proposed engagement to a strict intake gate. Purpose, end use, affected parties, provenance, legal basis, human control, deployment, downstream transfer, and capability scope must each have a reviewable evidence record. Missing, self-asserted, unverified, stale, or contradicted context cannot produce **COMPLIANT**.

The instrument classifies every intake into one of five categories (defined formally in [Definition 11](#)): insufficient information blocks inspection, outside scope finds no line implicated, and the remaining three — compliant, requires modification, non-compliant — carry an ordered strictness. Typed exemptions replace narrative carve-out clauses; explicit aliases replace heuristic stemming in scope normalization, where a light stemmer survives only in the advisory description/scope mismatch hint and never reaches a classification; and a named authorization records escalation without releasing a blocking finding. A hash-based canary detects drift only against a prior copy outside the writer’s control. The decisive limitations follow: local evidence is not independent truth verification, lexical matching is not semantic understanding, and code is not enforcement.

Those mechanics are also stated formally in [sec. 3.14](#), which restates the domain objects, the staged evaluator, and its decision rule as numbered definitions and propositions written from the shipped code, each bound to a named test that re-derives it, so a sentence that has drifted from the procedure it describes fails rather than persuades. Two propositions are exercised at scale: degrading any one of the nine intake dimensions withdraws a compliant result in all forty-five executed evaluations, and an ALL-mode exemption stays unreachable by a single declared token across fifty-eight more.

The scholarship is situated rather than universalizing. It joins pre-1900 primary texts and historical scholarship with contemporary work from multiple regions and disciplines, recording what question each source carries, what claim it does not establish, and where transfer into this personal instrument stops. Two literatures bound the project from either side: dual-use export control, which is what this question looks like when institutions answer it, and the refusal literature, whose standing a practitioner declining paid work cannot borrow. Turner’s framework is the mechanism source ([sec. 3.3](#)), not the title, subtitle, or authority of the project. The figures are deterministic and generated from the registry, evaluator, canary, and source ledger; captions separate implementation facts from interpretation and visual rhetoric, and the claim register and decision protocol carry the evidence chain and the operator sequence.

Red Line remains narrow. Black Line addresses positive operating discipline; Golden Line addresses aspiration; White Line addresses absence, restraint, and unknowability. They cross-reference one another without importing their substantive methods into this refusal boundary.

3.2 Introduction

Some work becomes harder to refuse once it has acquired a client, a deadline, or a sunk-cost narrative. Red Line places the refusal earlier. It is the author’s personal security boundary and explicit **No document**: a dated, revisable statement of work he will not accept, build, tune, release, or knowingly transfer.

This is not a complete ethics system, a legal opinion, or an enforcement service. Black Line concerns constructive practice; Golden Line concerns direction; White Line concerns absence, omission, and epistemic restraint. Red Line is narrower because a boundary must be usable at the moment of decision. It answers three practical questions: what is refused, what context must be established before a near-boundary action can be considered, and what the instrument cannot establish.

3.2.1 Four propositions

- 1. **Red Line is a security boundary.** Its seven first-person lines identify work that this author will not pursue or knowingly enable.
- 2. **Red Line is an explicit No document.** It is a personal commitment, not a universal moral authority and not a statement made by an AI system on the author’s behalf.
- 3. **Red Line is an evidence-gated auditability aid.** It makes a local review inspectable and reproducible; it does not prove safety, legality, or truth.
- 4. **Red Line escalates when context cannot be established.** Missing or unverified evidence is INSUFFICIENT_INFORMATION, not permission.

3.2.2 What this paper can establish

Evidence layer	It can establish here	It cannot establish
Personal commitment	what this author currently refuses	universal moral authority or another person’s consent
Local implementation	what the code and tests return for declared inputs	semantic truth, honest input, or operational safety
Scholarship transfer	which questions a source prompts	source endorsement, consensus, or public authorization
Render and release	that a specific validation run connected source to artifact	external certification or publication readiness without clean and independent witness gates

3.2.3 What the artifact is for

At the decision moment the instrument is a triage card rather than an essay: read the beacon, name the concrete capability and deployment tier, resolve the nine context dimensions with evidence instead of assertion, run the local evaluator, preserve the finding with its stable reason codes, and either narrow the work or stop. Maintaining a prior canary outside the author’s write boundary is the one step that runs on a different clock. sec. 3.7 states the sequence in full; the operator version is in docs/decision-protocol.md, and day to day the instrument is run through the daf-red-line skill in the author’s private daf-skills toolchain, so this paper is the theory and the skill is the practice. The point is relevance to a real decision under time pressure. A polished boundary that does not change what the practitioner records, asks, or refuses is only decoration. fig. 10

The design puts refusal before optimization and evidence before policy matching. It also separates four kinds of statement: a source may support a descriptive claim; an interpretation may transfer a question with an explicit stopping point; an implementation claim must be supported by code, tests, generated artifacts, validation, or inspected output; and a release claim must be tied to a validated source-to-render chain for a particular artifact. None of these surfaces can silently upgrade another.

3.2.4 Beacon, evaluator, canary

The registry is a **beacon**: collaborators can read the boundary before a request becomes a project. The evaluator is a narrow policy instrument, not a semantic judge. It requires a typed ActionContext, verified evidence for each required dimension, normalized scope, and a structured exemption before it can return COMPLIANT. A complete action that touches no registered line is OUTSIDE_SCOPE, which is deliberately not collapsed into compliance.

The canary is a dated attestation over canonical registry content. It can expose drift when a prior copy is held beyond the author’s write boundary; it cannot prevent a rewrite, prove authorship, enforce a finding, or independently establish that a source record is true. A review authorization records escalation but never unblocks a non-compliant or evidence-insufficient result.

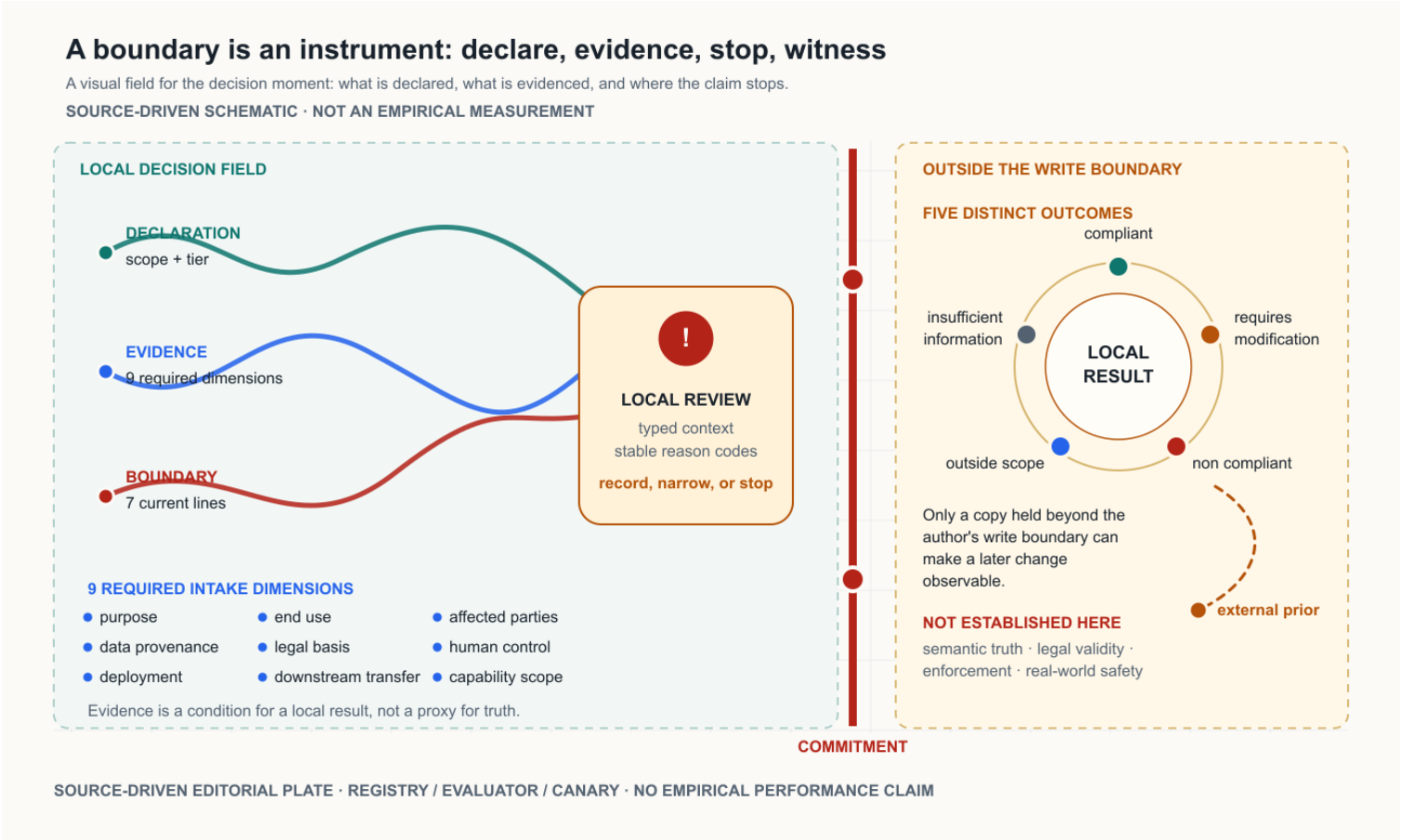


Figure 10: A boundary is an instrument: declare, evidence, stop, witness. This editorial plate turns the live registry, nine-field intake, five evaluator classifications, and external-prior condition into a single decision-time field. Its paper texture, traces, and red mark are visual rhetoric, not empirical data; the caption and alt text preserve the non-claims if the image is unavailable.

3.2.5 A bounded comparison

Alex Turner’s *A Red Line and Oversight Framework for Government AI Contracts* [Turner, 2026] is the mechanism source for this project’s architecture (sec. 3.3). Its organization-to-government setting, standing Review Body, legal context, and specific scope are not imported as authority. The adaptation keeps only a carefully marked pattern — precommitment, retained oversight, review classifications, and durability — and changes the subject to one practitioner’s own work.

The manuscript therefore keeps two questions separate: what the sources make visible, and what this author commits to. The scholarship widens the questions asked of the registry—who bears the cost of a false positive, what legibility can conceal, whose knowledge is being organized—but it does not convert a plural reading list into universal legitimacy. The claim register makes this stopping rule auditable.

The paper proceeds as follows. sec. 3.3 records the mechanism source. Section sec. 3.14 states the instrument as formal definitions and propositions. The red-line registry is composed in Section sec. 3.15, and Section sec. 3.17 closes with the instrument’s limitations and negative space.

3.3 Background — Turner as a bounded mechanism source

Alex Turner’s *A Red Line and Oversight Framework for Government AI Contracts* is the principal comparative source for this project’s architecture [Turner, 2026]. Turner writes at organization-to-government scale. The framework combines bright-line standards, retained-oversight tiers, a standing Review Body, review classifications, durability provisions, and transparency. This section records the source before describing the personal adaptation.

Turner’s formulation of appropriate human control is more demanding than a checkbox saying “human in the loop.” It asks whether an identifiable person is accountable, whether that person can exercise independent judgment, whether the system’s operational design permits meaningful evaluation, and whether legal and operational transparency are available [Turner, 2026]. It also recognizes that speed, volume, and complexity can make nominal human review ineffective. Red Line carries these as questions for the intake; it does not pretend that a text field or token can establish them.

Turner’s surveillance standard similarly distinguishes individualized, particularized analysis from inference based only on a category, population, or bulk dataset. The existence of data about a person is not, by itself, permission to generate an individualized assessment. Those details matter here because they show why Red Line’s `human_control`, `affected_parties`, `purpose`, and `data_provenance` fields must be evidenced rather than asserted.

Turner also describes scope-specific carve-outs, oversight thresholds, and durability procedures. Ambiguity resolves toward coverage in the source framework [Turner, 2026]. Red Line preserves that conservative direction but implements a stronger local first gate: unresolved context becomes `INSUFFICIENT_INFORMATION` before the registry can return a policy result.

3.3.1 What is and is not transferred

Transferred as a question: can a bright-line commitment be made inspectable, reviewed before action, and made harder to weaken silently? Not transferred as authority: Turner’s legal setting, government contracting relationship, seven-member institution, leadership powers, or claims about external enforcement. Red Line is a personal, self-authored instrument with an explicit external-witness limitation.

The implementation mapping appears in the next section. The important boundary is that an analogy to a governance mechanism is not evidence that the personal instrument has the institution’s powers.

The source framework also contains organizational, contractual, legal, and operational machinery that this project does not possess: a company, government counterparties, a standing Review Body, neutral-auditor access, safety-stack controls, and service suspension. Those omissions are not hidden implementation debt. They are the reason the local artifact claims auditability rather than institutional enforcement.

3.3.2 Provenance labels for the adaptation

The manuscript uses three labels so that resemblance does not become false attribution:

Element	Label	Exact boundary
Human control for harm-capable action	source-derived question; adapted intake	Turner supplies the accountable-human problem; Red Line adds fields and evidence, not Turner’s authority.
Individualized and proportionate scrutiny	source-derived distinction; adapted refusal	Turner supplies particularization; Red Line applies it to S2, not to a complete proportionality theory.
Bright lines, tiers, review, and durability	source-derived mechanism; adapted architecture	The local registry, evidence gate, five classifications, self-review, and canary are this project’s implementation.
Personal No document and strict evidence gate	independently authored	Personal design decisions, not Turner quotations, legal standards, or universal conclusions.

3.4 Global and Historical Scholarship: Situated Questions, Not a Universal Lineage

The reading base is an audit surface, not an authority bundle. The selection protocol prefers primary texts, scholarly editions, publisher pages, and official institutional sources; includes pre-1900 materials when their question remains useful; and records the boundary on every transfer. For every one of its 45 sources, the ledger in `docs/research-method.md` records place, period, the question carried into the audit, the non-transfer boundary, and a verified publisher or primary-text URL. Those 45 sources sit in 44 table rows: Kukutai and Taylor is listed in both tables, and the Cugoano row cites the primary text alongside its *Stanford Encyclopedia* entry.

Genre, language or translation, and a stable locator are recorded only for the 22 sources in the deepened second table; the machine-readable ledger at `data/source_claims.json` marks the remaining 23 `not recorded` rather than filling the field with boilerplate, and `validate_source_claims` rejects any locator derivable from a source’s own citation key.

3.4.1 Before 1900: refusal, rule, knowledge, and power

Sunzi’s *Art of War* makes information, deception, timing, and judgment operational questions [Sunzi, 1910]. Kautilya’s *Arthaśāstra*, in Patrick Olivelle’s scholarly translation, joins law, administration, intelligence, and the practical limits of rule [Kautilya, 2013]. Aristotle’s *Politics* asks how constitutions distribute authority, while al-Fārābī’s political philosophy links knowledge, political order, and human flourishing [Aristotle, -350, Mahdi, 2000]. Ibn Khaldūn’s account of ‘asabiyya and justice adds a historically situated analysis of cohesion and power [Darling, 2007].

These texts are not a single tradition. Machiavelli supplies an uncomfortable test about incentives and power [Machiavelli, 1513]; Wollstonecraft asks whether a discourse of reason can survive the denial of equal intellectual standing [Wollstonecraft, 1792]. Ottobah Cugoano’s late-eighteenth-century abolitionist work adds a Black Atlantic voice on liberty, consent, responsibility, and the self-deception that allows beneficiaries of coercion to call it legitimate [Cugoano, 1787, Dahl, 2025]. Cugoano is not used as a generic representative of “African ethics”; he is read in his own Atlantic, religious, abolitionist, and philosophical context.

The transfer from these sources is a set of questions: who acts, under what authority, with what knowledge, toward whom, with what power to withdraw or repair, and who bears the cost when a claim is wrong? None supplies a modern rights framework, a universal AI policy, or an endorsement of this registry.

The Latin American record adds a necessary asymmetry. Bartolomé de las Casas’s account is a colonial-era Spanish cleric’s polemic about violence in the Americas, not Indigenous testimony and not a representative voice for the region [Las Casas, 1552]. Its narrow use here is diagnostic: an institution can describe its own authority as administration, improvement, or salvation while affected people bear coercive costs. The source therefore sharpens the intake questions about affected parties and power; it does not supply a universal theory of consent or authorize the author’s categories.

3.4.2 Modern critical and institutional scholarship

Elinor Ostrom shifts attention from abstract market/state choices toward rules-in-use, monitoring, graduated responses, and situated knowledge [Ostrom, 1990a]. James C. Scott shows how administrative legibility can simplify a system while erasing local knowledge [Scott, 1998]. Linda Tuhiwai Smith makes research power and extraction part of method [Smith, 1999]. Helen Nissenbaum’s contextual integrity rejects a context-free account of information flow [Nissenbaum, 2004b]. Together they require Red Line to treat provenance, affected parties, downstream use, and public legibility as context-dependent rather than as checkboxes that automatically confer legitimacy.

Jobin, Ienca, and Vayena show convergence and divergence among AI principles; Selbst and colleagues warn that abstraction can sever technical categories from their sociotechnical context [Jobin et al., 2019, Selbst et al., 2019]. Birhane, and Mohamed, Png, and Isaac, put coloniality, dependence, local knowledge, and critical technical practice into the AI governance conversation [Birhane, 2020, Mohamed et al., 2020]. These sources constrain the project’s temptation to call a readable registry universal or emancipatory.

The surveillance and classification literature makes the same warning more concrete. Browne traces surveillance of Blackness through historical and contemporary practices [Browne, 2015]; Noble shows how apparently neutral search infrastructures can reproduce racialized hierarchy [Noble, 2018]; and Eubanks documents how automated administrative systems can concentrate burdens on people with little power to contest them [Eubanks, 2018]. Fricker supplies a distinct epistemic question—who is treated as a credible knower and who lacks interpretive resources [Fricker, 2007b]—while Couldry and Mejias frame data extraction as a relation of power rather than a mere technical flow [Couldry and Mejias, 2019]. Gray and Suri add hidden human labor to the system boundary [Gray and Suri, 2019]. Together these works justify keeping affected parties, provenance, downstream transfer, and human control as separate intake dimensions. They do not prove that any particular action is unlawful or unsafe.

Four further works make the bridge from scholarship to operating intelligence explicit. Haraway’s account of situated knowledges requires the author to name the standpoint and partial perspective behind a claim rather than perform a view from nowhere [Haraway, 1988b]. Jasanoff’s “technologies of humility” organize uncertainty around framing, vulnerability, distribution, and

learning [Jasanoff, 2003]. Costanza-Chock’s design-justice practice asks who is centered, burdened, excluded, or able to contest a design [Costanza-Chock, 2020]. D’Ignazio and Klein add a power-aware account of data, classification, and invisible labor; data do not speak for themselves [D’Ignazio and Klein, 2020b].

The transfer is operational, not ornamental. These questions attach to the intake as follows: standpoint and missing perspective sharpen provenance and affected parties; framing and vulnerability sharpen purpose, end use, and unknowns; participation and contestability sharpen human control and downstream transfer; and power, classification, and labor sharpen capability scope. None of the four sources can turn a self-assertion into verified evidence or make a local result a public authorization. Their value is that they make the operator ask a better question before the evaluator returns a bounded answer.

Kukutai and Taylor add a collective-authority test that individual consent does not settle: when data concern an Indigenous people or community, who has standing to govern collection, interpretation, and downstream use? [Kukutai and Taylor, 2016] This is a question for `affected_parties`, `data_provenance`, `legal_basis`, and `downstream_transfer`, not a new universal permission rule. The source is especially relevant because it prevents the intake from treating data sovereignty as a property of an individual record alone; it does not replace the authority of the affected community or establish that a particular data use is legitimate.

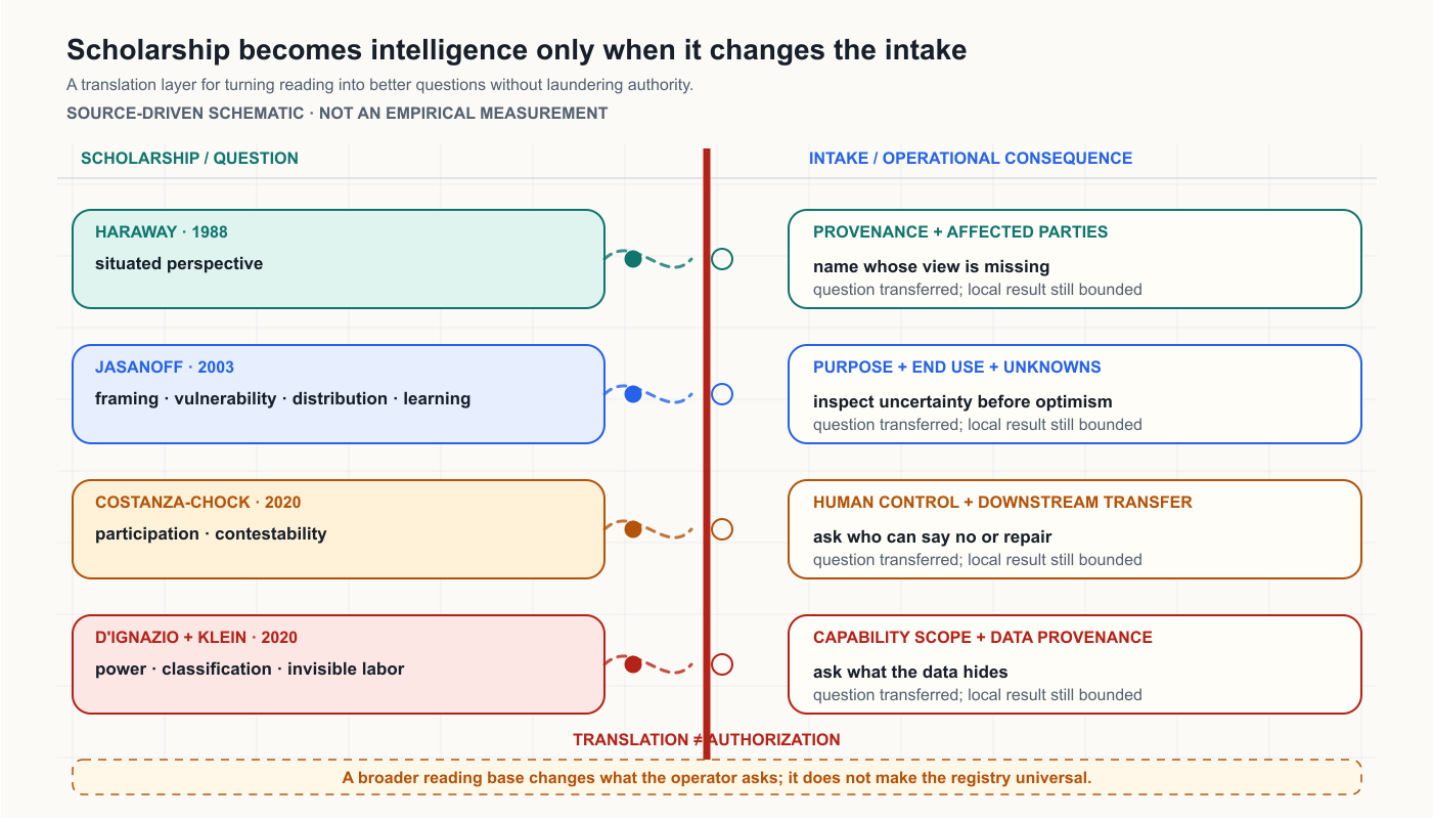


Figure 11: Scholarship becomes intelligence only when it changes the intake. This bridge maps four added works to concrete questions about perspective, uncertainty, contestability, classification, and hidden labor. The red gap is deliberate: sources widen the audit surface but do not authorize an action, verify an intake field, or replace affected-party testimony.

3.4.3 Export control and the refusal of work

Two literatures sit closer to this artifact than any cited so far, and neither of them flatters it. The first governs dual-use capability from above; the second describes refusing work from below. Red Line is neither, and saying so precisely is more useful than claiming kinship with both.

Export control is the mature institutional form of the question this registry asks. The Wassenaar Arrangement maintains a list of dual-use goods and technologies that participating states implement in national law [The Wassenaar Arrangement on Export Controls for Conventional Arms and Dual-Use Goods and Technologies, 2025]. Its structure is instructive twice over. It keys control to declared *capability categories* rather than to an assessment of the person applying for a licence, which is the same lexical bet Red Line makes and the same one sec. 3.17 concedes is escapable by description. And its 2013 addition of “intrusion software” is the reference case for what goes wrong when a capability definition is drawn slightly too wide: a category meant to constrain commercial spyware also described the exchange of defensive research, and the definition had to be revisited. A personal registry is smaller than a multilateral regime in every respect that matters, but it inherits that failure mode exactly, which is why the scope vocabulary in sec. 3.15 is enumerable and printed in full rather than described.

Harris’s comparative study of nuclear, biological, and cyber governance sets out the layering these regimes depend on: treaty obligations, export controls, institutional review, and the judgment of the researcher, each carrying part of the load and none of them sufficient alone [Harris, 2016]. The Fink report made the last of those layers explicit for the life sciences, naming seven categories of experiment that should trigger review and arguing that the scientific community’s own screening is a governance layer rather than an informal habit [National Research Council, 2004]. That is the strongest available warrant for an instrument like this one, and it is also a bounded one: the report proposed researcher judgment *alongside* institutional review boards, never instead of them, and Red Line has no board. What transfers is the proposition that a practitioner’s pre-commitment is a recognized layer. What does not transfer is any suggestion that the layer is adequate by itself.

The refusal literature supplies the vocabulary the registry was missing for its own act. Zong and Matias give refusal a structure — autonomy, timing, power, and cost — written deliberately from the standpoint of people refusing an institution’s data collection rather than the institution seeking compliance [Zong and Matias, 2024]. Their four facets are the sharpest available test of this instrument: it is individual rather than collective on autonomy, proactive rather than reactive on timing, weak on power because it binds only its author, and it redistributes cost onto that author alone. The transfer stops firmly at standing. Refusal from below is refusal by those with the least power in a relation; a practitioner declining paid work is not that, and borrowing the term’s moral weight would be exactly the authority-washing this section exists to prevent. The same boundary applies with more force to Simpson, whose account of refusal as a positive political stance — a sovereignty claim with its own content, not the absence of consent — belongs to Kahnawà:ke and to settler-colonial history [Simpson, 2014b]. The concept that a “no” can be generative rather than merely negative is what this project takes; the standing is not available to be taken.

Between the two literatures sits the one documented case of collective refusal in this industry. Crofts and van Rijswijk trace Google’s Project Maven episode: thousands of workers refused military AI work, the contract lapsed, principles were published — and the corporate form absorbed the objection intact [Crofts and van Rijswijk, 2020]. The episode is evidence that refusal is possible and legible, not a base rate. For a single practitioner the lesson is narrower still: refusal held collectively had leverage refusal held alone does not, which is the honest reason this artifact claims auditability rather than effect.

3.4.4 Selection limits and positionality

This is a curated conceptual audit, not a systematic review, a representative survey of world traditions, or a substitute for consultation with people affected by a proposed system. The selection is constrained by sources available in accessible editions and translations, the author’s questions, and the need to keep each transfer legible. Geographic variety is therefore a corrective to a narrow lineage, not a coverage statistic. A source from a region is not a proxy for everyone in that region; a translated text is not identical to its language of composition; and placing sources in one figure does not make them equivalent.

The consequence is methodological restraint: the reading base can expand the intake questions—about power, classification, labor, consent, and contestability—but it cannot supply affected-party testimony, jurisdiction-specific legal review, or independent verification of an evidence record. Those remain separate obligations.

For dual-use security, Brundage and colleagues map malicious-use concerns across digital, physical, and political domains and argue for prevention and mitigation rather than a single prediction [Brundage et al., 2018]. That supports a capability-and-transfer question in Red Line, not a claim that a lexical registry forecasts threat likelihood. Three engineering references sit beside it and are taken up again in sec. 3.12, where they describe what this project does and does not do: the NIST Secure Software Development Framework, MITRE ATT&CK, and SLSA v1.2 [Souppaya et al., 2022, MITRE, 2026, SLSA Community, 2026]. Here they enter the reading base for one reason only — they name the questions to ask about provenance, adversary behavior, and build attestation. Neither a standard nor a threat catalog is evidence that Red Line is secure, legally compliant, or resistant to an advanced persistent threat.

UNESCO, the OECD, and the African Union provide policy comparison points; NIST provides operational vocabularies for risk management and continuous verification [UNESCO, 2021, OECD, 2019, African Union Commission, 2022, Tabassi, 2023, Rose et al., 2020]. They are not certifications of Red Line and cannot substitute for the project’s own evidence gate.

The method is assembled from many situated questions, not one universal lineage

Situated questions; placement does not imply influence or equivalence.

SOURCE-DRIVEN SCHEMATIC · NOT AN EMPIRICAL MEASUREMENT

BEFORE 1900

early and early-modern questions

c. 5th BCE
China
information, deception, judgment
Sunzi

c. 3rd BCE–3rd CE
India
statecraft, intelligence
Kauṭilya / Arthaśāstra

4th BCE
Greece
law, authority, constitutions
Aristotle

10th c.
Central Asia / Islamic world
order, knowledge, flourishing
al-Fārābī

14th c.
North Africa
cohesion, justice, power
Ibn Khaldūn

1513
Italy
institutions, incentives, realism
Machiavelli

1552
Spanish colonial Americas
violence hidden by authority
Las Casas

1787
Gold Coast / Black Atlantic
liberty, consent, responsibility
Ottobah Cugoano

1792
England
capacity, education, standing
Wollstonecraft

1900–2000

institutional and methodological critiques

1990
United States / global commons
self-governance and rules in use
Ostrom

1998
United States / Southeast Asia
legibility, simplification, state power
James C. Scott

1999
Aotearoa New Zealand / Māori scholarship
research power and decolonization
Linda Tuhiwai Smith

2000–PRESENT

AI governance and implementation

2004
United States
contextual integrity of information flows
Nissenbaum

2016
Aotearoa / Indigenous data governance
collective data authority
Kukutai · Taylor

2019–20
Global / Africa / decolonial AI
plural ethics, coloniality, sociotechnical limits
Jobin · Birhane · Mohamed et al.

2021–23
Global institutions
policy action, rights, operational risk
UNESCO · OECD · NIST

Reading rule

borrow questions; do not claim inheritance, consensus, or endorsement

Figure 12: The method is assembled from many situated questions, not one universal lineage. The cards group pre-1900 primary texts, historical scholarship, critical methodology, collective data-governance work, and contemporary standards by broad period and region. Direct labels state the question carried into Red Line; placement does not imply direct influence, equivalence, or endorsement. The figure deliberately includes an interpretive boundary: a source can widen the audit surface without authorizing the personal refusal registry. It is a provenance and humility device, not a measurement of global coverage or scholarly quality.

Every source transfer carries a question and a stopping point

Every row records a source question, claim class, and explicit interpretation limit.

SOURCE-DRIVEN SCHEMATIC · NOT AN EMPIRICAL MEASUREMENT

SITUATED SOURCE	QUESTION CARRIED	CLAIM CLASS	STOPPING POINT
Sunzi · China	judgment under deception	DESCRIPTIVE	not a modern ethics code
Kauṭilya · India	administration and intelligence	DESCRIPTIVE	not a universal model of rule
al-Fārābī · Islamic world	knowledge and political order	TRANSFER	not a rights-equivalent framework
Ibn Khaldūn · North Africa	cohesion, justice, power	TRANSFER	not a timeless sociology
Cugoano · Black Atlantic	liberty, consent, responsibility	TRANSFER	not generic continental representation
Wollstonecraft · England	equal intellectual standing	TRANSFER	not a complete justice theory
Smith · Aotearoa / Māori	research power and extraction	TRANSFER	not decorative diversity
Kukutai · Taylor · Indigenous data	collective authority and governance	TRANSFER	not a consent shortcut
Birhane · Africa	dependency and coloniality	DESCRIPTIVE	not one effect for every import
UNESCO / AU / NIST	policy and operational vocabulary	IMPLEMENTATION	not Red Line approval

Reading rule: transfer a question, never a universal endorsement; claim class and stopping point travel together.

Figure 13: Every source transfer carries a question and a stopping point. This deterministic matrix makes the claim discipline visible: descriptive, transfer, and implementation claims are separated, and each source row names the boundary beyond which it is not used. Collective data authority appears alongside individual consent and contextual privacy; the visual does not score traditions, demonstrate consensus, or make the reading base universal. It is designed for the PDF and narrow HTML layout with direct labels, repeated color-and-text encodings, and a caption that remains meaningful if the image is unavailable.

3.5 The four-line set: boundary, method, aspiration, absence

Red Line is the first work in a four-line set, but it is not an omnibus theory. It is the security boundary and explicit **No document**: a dated record of work the author refuses, the conditions under which a proposed action is reviewed, and the limits of a self-authored canary.

The companion works answer different questions. Black Line describes the positive operating discipline for doing strong, concise, intelligent work and research. Golden Line describes a higher aspirational thread: directions that can orient a life or project without becoming a compliance verdict. White Line describes what is absent or unknowable, including epistemic gaps, ethical restraint, withheld material, and the negative space around a claim.

The separation is substantive. Red Line does not turn good method into permission; Black Line does not turn aspiration into prohibition; Golden Line does not turn uncertainty into failure; White Line does not turn silence into evidence. Each work has its own registry, vocabulary, executable interface, sources, figures, tests, and limitations, and each is its own repository — `docxology/black_line`, `docxology/golden_line`, and `docxology/white_line`. The relationship is recorded in the companion `line_set` work (https://github.com/docxology/line_set), which declares the set and checks that no two instruments gave the same spelling to different things.

The distinction is summarized in fig. 14. The diagram is an orientation aid, not a hierarchy: the colors name different jobs, not different degrees of moral worth.

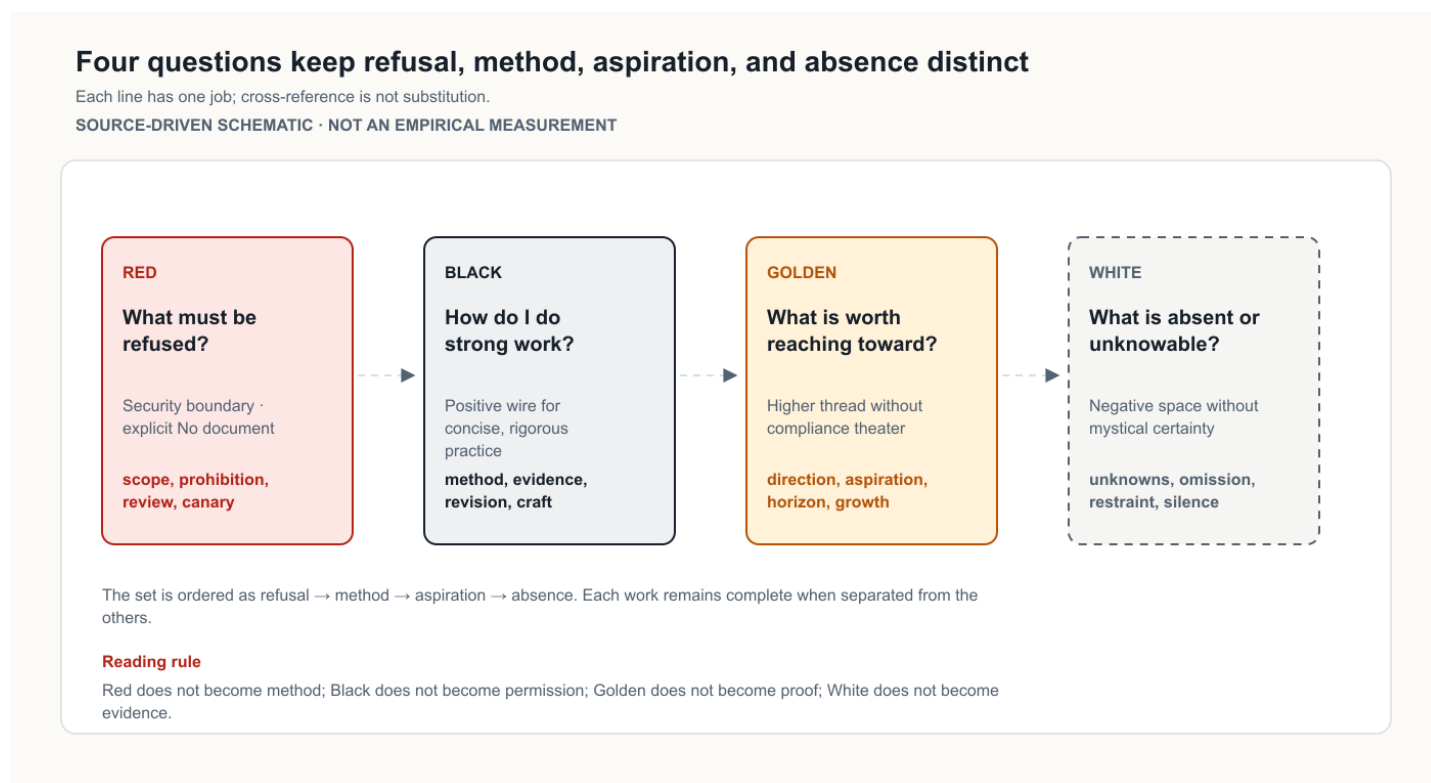


Figure 14: Four questions keep refusal, method, aspiration, and absence distinct. Orientation schematic for the four-line set: Red Line is the refusal boundary and explicit No document; Black Line is the positive wire for strong, concise, intelligent work; Golden Line is an aspirational direction rather than a pass/fail rule; and White Line records what is absent, withheld, uncertain, or left outside a claim. The colors are semantic accents, not severity scores. The four works are standalone instruments with different registries and evaluators, and the figure does not imply that one line can substitute for another.

The current project therefore remains narrow by design. Its questions are:

1. What must this practitioner refuse?
2. How can that refusal be made inspectable before work begins?
3. What can a registry, evaluator, self-review record, and external prior copy demonstrate, and what can they not demonstrate?

Black, Golden, and White now exist as sibling instruments in their own repositories, each with its own release number; the `line_set` companion records their ordering and non-overlap contract, and this paper pins none of their version numbers. Red Line remains the reference boundary rather than a container for their adjacent concerns. A cross-reference does not create a combined evaluator: a positive method cannot authorize a refused action, an aspiration cannot resolve missing evidence, and a statement of absence

cannot prove safety.

One vocabulary item is deliberately shared across the set, and a reader holding only this paper needs to know it. Black Line also publishes a status spelled `OUTSIDE_SCOPE`, and it does not mean there what it means here: in Red Line it marks a complete, evidenced intake that implicates no current registry line, while in Black Line it marks an attempt outside that discipline’s evaluation scope. One is a finding of absence after inspection; the other is a declination to inspect. The overlap is declared rather than accidental, and neither sense may be read into the other.

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 — including the shared token just named. It adds no substantive instrument, and Red Line does not import, depend on, or defer to it.

3.5.1 Note on the name

This paper’s alchemical stage is **Rubedo**, the reddening — the culminating phase of the classical *magnum opus* — and the color openly echoes that opus. The set’s four colors (black, white, gold, red) name the stages of the alchemical work in plain sight. That echo is read here only in Carl Jung’s symbolic-psychological register, where the *opus* is a map of individuation; it is not an empirical, mystical, or causal claim about matter, minds, or this instrument. “Green is not proof” applies to a fluent symbolic parallel as much as to a passed test. The opus reads rubedo as a culmination, but the name is borrowed as symbol, not as rank: Red Line is not the completion of the set nor superior to the other three instruments — the four are peers, and Red Line is merely the one that answers what to refuse. Note too that the set’s working order — refuse, method, aspire, absence — is functional, chosen for how the instruments are used, and does not reenact the classical stage sequence (in which reddening comes last, not first). The full framing, with its single Jung citation and the honest esoteric caveats, is documented in the `line_set` companion work (https://github.com/docxology/line_set) and is not restated here.

3.6 First-principles design: what the artifact can actually do

Before asking whether Red Line resembles another governance framework, ask what problem the artifact must solve and what information it can actually possess. One practitioner, deciding alone, before the work starts. That is the whole situation the artifact is built for, and it fixes what the artifact can know: whatever the practitioner has written down and whatever he can check locally. Everything the abstract promises follows from that constraint rather than from an ambition — and so does everything it refuses to promise, which is universal morality, verification of the world, and stopping an action by software alone.

3.6.1 Deconstruction

The project is made of six constituent parts:

Part	Irreducible job	Evidence in this repository
Personal commitment	State what this author will not accept, build, tune, release, or knowingly transfer	Seven dated first-person RedLine records
Intake	Describe an action before its description can be used as a policy shortcut	ProposedAction and nine-field ActionContext
Evidence gate	Refuse a green result when required context is missing, stale, contradicted, or merely asserted	Typed EvidenceRecord values and INSUFFICIENT_INFORMATION
Local decision procedure	Apply declared scope, typed exemptions, and oversight floors consistently	evaluate_action and its five classifications (Definition 11)
Review and change record	Preserve what was decided and expose registry drift	Frozen ReviewFinding, transparency aggregation, and CanaryStatement
Publication surface	Let a reader compare source, implementation, and rendered artifact	Beacon prose, source ledger, deterministic figures, PDF, and HTML

The parts are complementary but not interchangeable. A hash cannot replace a review. A review cannot replace evidence. Evidence cannot become truth merely because it is labeled VERIFIED. A public document cannot become universal authority merely because it cites sources from many places.

3.6.2 Fundamental truths

The design begins from facts that survive removal of familiar labels:

1. A personal refusal has authority over the author’s own participation, not over other people or institutions.
2. A local program receives declarations and pointers to records; it does not, by inspecting their labels, know whether the underlying world is true.
3. A lexical scope matcher can apply explicit vocabulary consistently, but it cannot infer the actual capability or purpose of arbitrary work.
4. A cryptographic digest can show that current content differs from a prior digest. It does not prove authorship, authenticity, semantic adequacy, or non-forgeability.
5. A frozen finding records a result; without an external authority or admission control it cannot physically prevent execution.
6. If the conditions for a review cannot be established, treating the action as compliant would manufacture certainty from absence. This is the classical fail-safe-defaults posture: base a decision on demonstrated permission rather than on the mere absence of objection.
7. A publication claim is only as strong as the chain connecting its source, code, tests, generated assets, and rendered output.

These are not claims that the instrument is safe. They are the reasons its strongest honest output is often a stop, a limitation, or a request for an independent witness.

3.6.3 Constraint analysis

The project also contains choices that should not be mistaken for laws of nature. The following classification keeps the mechanism revisable:

Design element	Classification	Why it is kept or challenged
First-person, dated provenance	Hard requirement of the stated function	Without an accountable author and date, the artifact cannot be a personal commitment.
Fail closed on unresolved required context	Hard epistemic consequence	Missing information cannot logically support a positive review result.
Nine required intake dimensions	Deliberate policy choice	The fields are a broad audit surface, not a complete ontology; new evidence needs may require revision.
Explicit aliases and an ASCII boundary	Deliberate policy choice	They block trivial masquerading without pretending to solve semantic interpretation.
Seven current lines	Deliberate scope choice	<code>REGISTRY_IS_EXHAUSTIVE = False</code> ; absence is not endorsement.
180-day freshness window	Maintenance policy	It is a useful review cadence, not a universal truth about evidence expiry.
Turner as the organizing comparison	Removable interpretive choice	The project must stand without importing Turner’s authority, institution, or legal claims.
External prior canary copy	Hard condition for external change detection	A same-author file can be rewritten together with the registry and therefore cannot witness itself.
PDF and HTML validation	Hard requirement of a publication claim	A source that does not reach the reader intact has not been successfully published.

3.6.4 Reconstruction: what the ordering has to be

Those truths do not yet give a procedure, but they fix its order. Refusal comes before optimization, because a boundary consulted after a project has a client is a boundary that argues with sunk cost. Evidence comes before policy matching, because a description that can select its own exemption token is not a description. A result comes before authorization, because an escalation that can be recorded ahead of a finding is an override wearing a different word. And a local attestation comes before, without ever substituting for, an independent witness — a file the author can rewrite cannot testify about the author.

The eight-step form the practitioner actually runs is in sec. 3.7; it is one procedure stated once, not a summary and an expansion.

3.6.5 Claim classes

The manuscript uses four claim classes so that a polished document does not make unlike evidence look interchangeable:

Claim class	What can support it	What cannot upgrade it
Descriptive	A cited source, edition, and stable locator	A source’s existence does not establish the author’s interpretation of it
Transfer / interpretive	An explicit question, rationale, and stopping point	A broad reading list does not create universal legitimacy
Implementation	Source code, tests, generated figures, validation, and inspected output	Green code does not verify the world, the source records, or operational safety
Release	A validated source-to-render chain, artifact manifest, and inspected PDF/HTML	A successful build does not provide external certification or real-world outcome evidence

The adaptation is therefore a bounded construction rather than a conclusion that the cited traditions, standards, or threat catalogs endorse Red Line. The artifact is strongest when it says exactly what it did, what it refused to infer, and where another reviewer must take over.

The resulting operating sequence is made explicit in sec. 3.7, where each claim carries an evidence state and stopping point through the local decision and publication gates.

3.7 Operating method: from first principles to a bounded release

The first-principles reconstruction (Section sec. 3.6) fixes the order; this section gives the procedure.

The first-principles section identifies the artifact’s irreducible job: help one practitioner decline or narrow high-risk development work before commitment, record the local reason, and disclose what the record cannot establish. This section turns that reconstruction into a repeatable method. It is a protocol for producing inspectable evidence, not a claim that the protocol can discover semantic truth or guarantee a safe outcome.

3.7.1 The unit of analysis

Keep three objects separate:

Object	Question	Permitted evidence
Observation	What was declared, recorded, or rendered?	a typed field, dated record, source locator, test output, or artifact hash
Interpretation	What question or limit is being transferred?	an explicit rationale and stopping point attached to a source or claim
Action decision	What may this author do within this local registry?	the evidence-gated evaluator result for a declared scope, tier, and review date

A source can support an interpretation without authorizing an action. A test can support an implementation claim without verifying the truth of an intake field. A rendered page can show that a statement reached an artifact without showing that the statement is correct in the world. This separation is the central anti-confusion rule.

3.7.2 The operating loop

1. **Deconstruct the function.** State the concrete decision or release problem before importing a label such as governance, security, or safety. Identify who can refuse, who may be affected, and what the instrument can actually observe.
2. **Challenge the boundary.** List hidden assumptions about authority, scope, evidence, freshness, semantic interpretation, and external witnessing. Mark each as a hard constraint, a local policy choice, or an open question.
3. **Declare the claim.** Assign a claim class (**descriptive**, **transfer**, **implementation**, or **release**), a verification mode, a supporting surface, and a stopping point. The machine-readable register in `data/claim_register.json` is the publication-contract.
4. **Reconstruct the smallest honest rule.** Represent an action with its scope, deployment tier, nine context dimensions, evidence records, and explicit unknowns. Do not let free-text purpose or an exemption token substitute for typed context.
5. **Evaluate fail-closed.** Normalize the scope, check missingness, unresolved statuses, freshness, and ambiguity before policy matching. Then apply typed exemptions and tier floors. The evaluator’s five classifications (**Definition 11**) remain distinct and ordered, from the intake-blocking `INSUFFICIENT_INFORMATION` through `OUTSIDE_SCOPE`.
6. **Freeze the record.** Preserve the classification, implicated lines, human-readable reasons, stable reason codes, normalized scope, evidence-stop dimensions, review date, and any authorization. An authorization may document escalation or remediation; it cannot turn a block into permission.
7. **Verify the publication chain.** Rebuild source-driven figures, validate bindings, run tests and coverage, render PDF and HTML, and inspect the combined artifacts. A source claim is not a release claim until its required surface exists and is checked.
8. **Revise or stop.** If a result, source, figure, or rendered artifact fails its criterion, record the defect and either amend it with a dated rationale or stop. Never replace an unresolved state with a favorable default.

3.7.3 Evidence states and stop points

The protocol uses evidence as a condition for a local decision, not as a proxy for truth. `MISSING` means no usable value or supporting record exists; `SELF_ASSERTED` means the requester supplied an assertion without an independent reviewable basis; `UNVERIFIED` means a pointer exists but has not been checked; `CONTRADICTED` means the available record conflicts with the declaration; and `STALE` means the record is outside the configured review window or future-dated. None can narrow a line. `VERIFIED` means only that a reviewable record was checked for this decision and remains current; it can narrow a line only through its typed exemption and declared scope.

This vocabulary prevents an important category error. A verified legal-basis record is evidence that a reviewer checked the supplied record; it is not a legal opinion generated by Red Line. A verified human-control record is not proof that people will exercise that control under pressure. The evaluator therefore refuses to manufacture `COMPLIANT` from an information gap.

An evidence-gated improvement loop keeps claim, action, and artifact boundaries aligned

A bounded workflow for turning uncertainty into inspectable revision.

SOURCE-DRIVEN SCHEMATIC · NOT AN EMPIRICAL MEASUREMENT

LOCAL, REPRODUCIBLE ARTIFACT



A green local result is a bounded record, not a safety certificate; a failed gate is a reason to stop or revise, not a reason to hide uncertainty.

OUTSIDE THE ARTIFACT

The loop can record a local decision and validate a generated release.

It cannot establish:

- semantic truth
- legal validity
- enforcement
- independent witnessing
- real-world safety

Figure 15: An evidence-gated improvement loop keeps claim, action, and artifact boundaries aligned. The eight-step schematic moves from deconstruction and challenge through claim declaration, reconstruction, fail-closed evaluation, frozen records, source-to-render verification, and dated revision. The outside panel names semantic truth, legal validity, enforcement, independent witnessing, and real-world safety as claims this local loop cannot establish. It is a reproducible method map, not a measurement of improvement or safety.

The frozen `ReviewFinding` carries stable reason-code values alongside the display prose. This is a small but important audit distinction: prose can be clarified without forcing downstream consumers to parse wording, while the codes make a block such as `STALE_EVIDENCE` or `UNEXEMPTED_LINE` available for aggregation and regression checks.

3.7.4 Falsification and negative controls

The method is credible only if it can fail visibly. The release checks should include negative controls that deliberately attempt to cross the boundary:

- remove a required claim class, stopping point, or verification mode;
- add a duplicate or unknown claim identifier to the prose register;
- describe a stale, self-asserted, or contradicted evidence record as current;
- make the figure registry and visualization brief disagree;
- alter a source-driven figure and test whether byte determinism detects the difference; and
- render without a required artifact and verify that the release manifest refuses readiness.

These controls test the instrument’s refusal behavior. They do not estimate false-positive or false-negative rates in the world, because this project has no labeled corpus of real engagements and no independent operational observer. The appropriate conclusion is narrower: the local gates either detect the planted defect or they do not.

3.7.5 Scope of inference

The method can support a statement such as: “given this registry, this typed intake, this review date, and these recorded evidence statuses, the evaluator returned this classification.” It cannot support: “the action is safe,” “the action is lawful,” “the source endorses the framework,” or “the rendered artifact has been independently certified.” Those stronger conclusions require different authorities and evidence. The project is most coherent when it keeps that stopping point attached to every claim, figure, and release record.

3.8 The Adaptation Thesis

Turner’s framework demonstrates a pattern: make a narrow refusal explicit, grade release by retained oversight, issue a review finding, and preserve a durable record [Turner, 2026]. Red Line adapts that pattern to one practitioner. The transfer is methodological, not institutional. The first-principles reconstruction in sec. 3.6 is the controlling logic; the Turner comparison is useful only where it clarifies a mechanism.

The personal version changes the decision boundary in an important way. A proposed action is not allowed to reach policy matching on the strength of a description or a self-selected exemption token. It must carry an **ActionContext** and evidence records for purpose, end use, affected parties, data provenance, legal basis, human control, deployment, downstream transfer, and capability scope. That is an implementation choice made here in response to the risk of scope laundering and false certainty; it is not attributed to Turner. The evidence gate is a refusal to manufacture a green result, not a claim that the local verifier has discovered the truth of the records.

3.8.1 Four adaptations

1. The standing Review Body becomes a written self-review record plus an external-witness hook. The absence of independent enforcement is explicit.
2. Deployment tiers become grades of how much the author can observe, update, suspend, or withdraw a work product after release.
3. The registry retains narrative carve-outs but implements narrowing clauses as typed exemptions requiring verified evidence.
4. Durability becomes deterministic hashing and canary verification, while the successor rationale makes registry changes visible rather than forbidden. The hash is a change signal, not a signature or an enforcement mechanism.

The result is not a private morality engine. It is a reproducible local review instrument whose strongest claim is: “given this registry, this intake, this review date, and these recorded evidence statuses, this is the documented result.” A reader should be able to identify which parts are source-derived, which are the author’s commitments, and which are implementation facts.

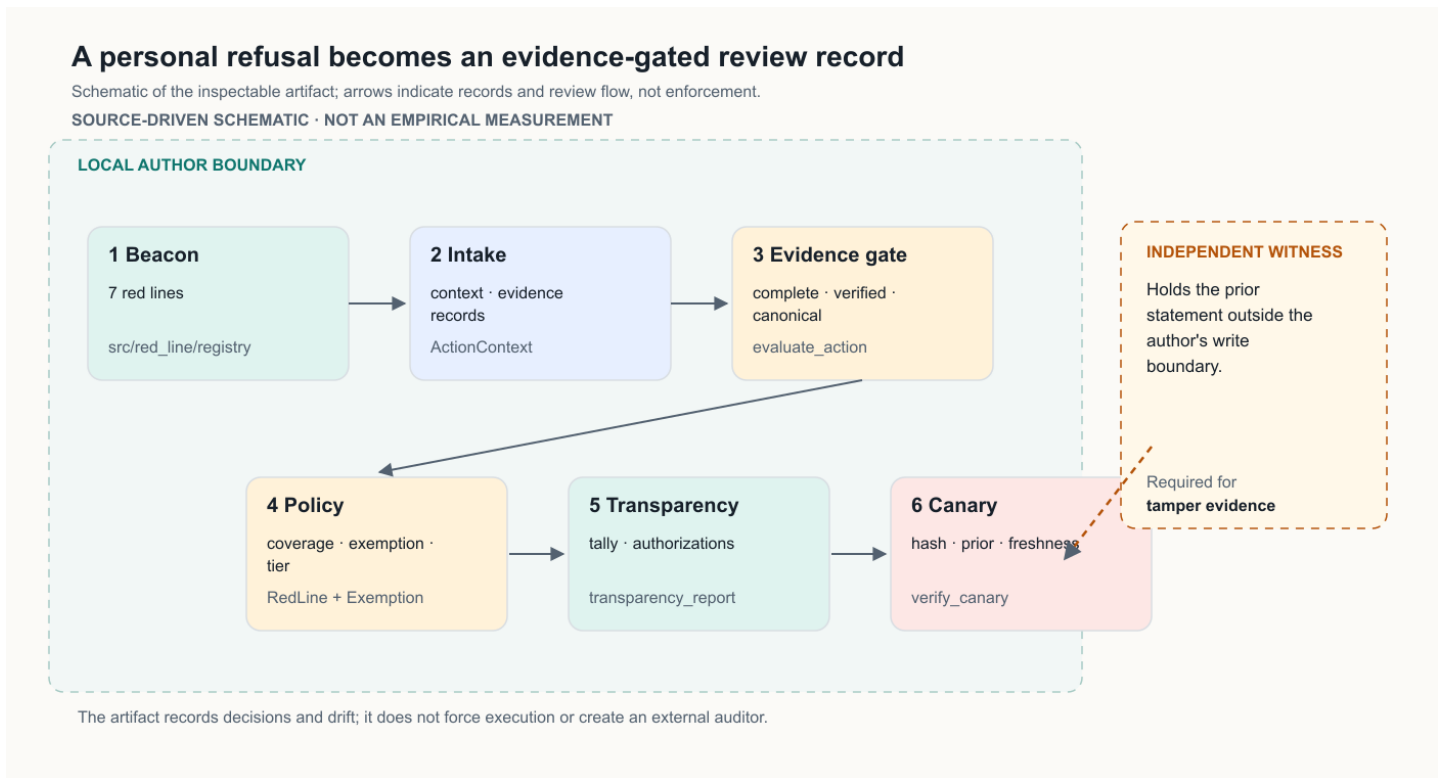


Figure 16: The framework turns a private commitment into an evidence-gated review event. The deterministic architecture schematic draws six stages inside the local author boundary: the seven-line registry beacon, the action-declaration intake, the evidence gate, the policy match over lines and typed exemptions, the transparency tally, and the canary check. The external-witness boundary remains outside the local author boundary. Arrows describe records and accountability flow, not enforcement; a green result is not a safety certification.

3.9 The two CANARY-severity boundaries

Turner’s framework gives special attention to human control over harm-capable systems and to individualized scrutiny that is not generated from bulk data or group membership [Turner, 2026]. Red Line carries these as its two CANARY-severity records, **s1-human-control-force** and **s2-untargeted-profiling**, while making clear that the personal registry is not Turner’s institution or legal standard.

3.9.1 S1 — force and harm-capable systems

The first line refuses building, tuning, or knowingly supplying a component to a system that selects and engages targets for force without appropriate human control over each engagement. In this revision, “human control” is not proved by the token **human_in_loop**. A compliant typed exemption requires verified context about the accountable human, the end use, the purpose, and the operational ability to review the action. The local evaluator can require those records; it cannot inspect a deployment and decide that the judgment was genuinely independent or operationally meaningful.

The typed exemptions cover defensive alerting and adjacent support such as logistics, translation, maintenance, research, and accountable intelligence analysis. The exemption trigger is only a declaration; the required evidence must still be verified. Multiple prohibited dimensions resolve to **REQUIRES_MODIFICATION**, and an unsatisfied exemption never softens the line. The source framework’s more specific requirements around proportionality, operational tempo, and legal transparency remain questions for human review, not claims discharged by this lexical interface [Turner, 2026].

3.9.2 S2 — untargeted profiling

The second line refuses converting bulk data into individualized intelligence on people who are not already subjects of a specific, lawful, individualized process. Turner’s source formulation makes particularization important: data about a person does not itself authorize an individualized assessment, and an assessment should not be initiated solely from a demographic, national-origin, religious, or political category [Turner, 2026].

Red Line therefore requires evidence for affected parties, provenance, legal basis, and purpose even where an action declares **aggregate_research**, **identified_subject**, or **opt_in_analytics**. These tokens select a possible exemption path; they do not create consent or legality. The local line also does not implement the source framework’s full proportionality, territory, independent-basis, or downstream institutional controls. That omission is a scope boundary, not an implied approval.

3.9.3 Canary status

Both records remain CANARY-grade. Here “CANARY” is a change-signal grade, not a ranking of moral importance. Structural invariants require that both ids remain present, retain CANARY severity, and never allow an **AIR_GAPPED** ceiling. Their narrative carve-outs remain readable in the beacon, while their executable exemptions and evidence requirements are included in the canonical registry digest.

3.10 Deployment tiers as oversight-retention grades

Turner relates deployment eligibility to the oversight retained over a system after delivery [Turner, 2026]. Red Line reinterprets that mechanism for a single practitioner:

- HOSTED retains author-operated observation and withdrawal;
- CONNECTED retains a maintained update or suspension path;
- AIR_GAPPED means beyond recall, such as an unrestricted release or handed-off model.

The `oversight_rank` values are 2, 1, and 0 respectively. Each registry line declares a `max_tier`; despite that field name, it is used as a minimum retained- oversight floor: the least-oversight environment in which a fully evidenced typed exemption may operate. A tier floor is never a permission by itself: scope, context, and exemption evidence still govern. Definition 1 and Definition 13 state the rank and the floor test exactly as the code applies them.

The evaluator is monotonic in danger. An unexempted implicated line is `NON_COMPLIANT` at every tier; reducing retained oversight adds an aggravating reason but cannot turn a hard block into a softer outcome. A verified exemption below its line’s floor becomes `REQUIRES_MODIFICATION`. Proposition 8 states this as a property and names the test that exercises it. It is a consistency property of the local decision procedure, not empirical evidence that a hosted system is safe or that an air-gapped system is dangerous in every circumstance.

“Air-gapped” is shorthand for beyond the author’s practical ability to observe, update, suspend, or withdraw the work product. It is not a security certification and does not mean that a released artifact is literally disconnected in every deployment.



Figure 17: Less retained oversight narrows the release envelope. The registry-derived matrix marks each line’s oversight floor and labels the tiers directly; a filled cell means only that the tier floor is met. It does not mean an action is compliant, safe, or legally permitted. The two CANARY records remain visibly bounded away from AIR_GAPPED release by an executable invariant. Shape, text, and cell state repeat the meaning so the figure remains interpretable in grayscale and on a narrow page.

The monotonicity property itself is exercised, not merely asserted. The analysis module `red_line.analysis.monotonicity` sweeps all 36 line/keyword slots (34 distinct tokens) across the seven current lines through the real `evaluate_action` at each of the three deployment tiers — 108 executed evaluations with a fully evidenced fixture intake — and records the verdict lattice the evaluator actually returned, with zero inversions. The slot count exceeds the vocabulary because `handoff` and `provenance` are each declared by two lines and are therefore swept once per line.

Dropping retained oversight never softens a verdict

Each chip is one executed evaluate_action run at review date 2026-07-15; strictness may only grow as retained oversight drops.

SOURCE-DRIVEN SCHEMATIC · NOT AN EMPIRICAL MEASUREMENT

LINE · SCOPE KEYWORD most oversight → least oversight		HOSTED full observation + withdrawal	CONNECTED maintained update / suspend	AIR-GAPPED beyond recall	MONOTONE
cogsec-integrity	cogsec	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	deception	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	disinformation	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	influence_ops	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	manipulation	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	propaganda	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
downstream-transfer	downstream	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	handoff	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	integration	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	resale	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	sublicense	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
dual-use-ablation	handoff	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	model_release	COMPLIANT	COMPLIANT	NON COMPLIANT	✓
	weights	COMPLIANT	COMPLIANT	NON COMPLIANT	✓
open-science-good-faith	benchmark_claim	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	metric	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	provenance	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	publication	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	reproducibility	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
provenance-and-consent	consent	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	data_acquisition	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	pii	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	provenance	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
s1-human-control-force	scraping	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	autonomous_weapon	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	force	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	kinetic	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	lethality	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
s2-untargeted-profiling	targeting	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	weapons	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	biometric_id	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	bulk_data	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	dragnet	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	profiling	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	surveillance	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓
	tracking	NON COMPLIANT	NON COMPLIANT	NON COMPLIANT	✓

36 line/keyword slots (34 distinct tokens) × 3 tiers = 108 executed evaluate_action runs · inversions: 0

Positive control lives in the regression suite: the identical sweep flags the replicated pre-fix defect, so this green is falsifiable.

Monotone strictness is a consistency property of the local decision procedure run on fixture evidence. It is not a safety score, an accreditation, or evidence that any real engagement was reviewed.

Figure 18: Dropping retained oversight never softens a verdict. Executed verdict-strictness lattice computed by `red_line.analysis.monotonicity.run_monotonicity_sweep`: all 36 line/keyword slots (34 distinct tokens) across the seven current lines run through the real `evaluate_action` with a fully evidenced fixture intake at each of the three deployment tiers — 108 executed evaluations — and every chip is the classification actually returned. Read left to right, a row moves from most to least retained oversight and strictness never decreases; the sweep records zero inversions, and its regression test carries a positive control that detects the replicated pre-fix defect. Monotonicity is a consistency property of the local decision procedure exercised with fixture evidence, not a safety measurement or a review of any real engagement.

3.11 Written review and transparency

Turner uses a standing Review Body and a fixed vocabulary of findings [Turner, 2026]. Red Line cannot recreate that institutional separation. It keeps the durable local outputs: a written finding for each reviewed action and a tally that makes classifications and escalations visible.

3.11.1 Finding record

`review_engagement` returns a frozen `ReviewFinding` containing the engagement, classification, implicated line ids, finding text, review date, declared scope, tier, and ambiguity flag. The evaluator is run as of that review date, so a finding cannot silently describe one date while evaluating freshness at another. The finding preserves `INSUFFICIENT_INFORMATION` as a blocking result; it does not turn incomplete intake into a policy judgment.

The result vocabulary is:

Result	Meaning
<code>INSUFFICIENT_INFORMATION</code>	required context or evidence is unresolved; stop
<code>NON_COMPLIANT</code>	a line is implicated without a verified narrowing condition
<code>REQUIRES_MODIFICATION</code>	a verified exemption still has a dimension or tier problem
<code>COMPLIANT</code>	complete intake and verified exemption satisfy the local registry
<code>OUTSIDE_SCOPE</code>	complete intake, but no current registry line applies

Outside scope is not a compliance certification. It says only that this registry did not match the documented action. A transparency report aggregates the findings supplied to it; it is not an automatic publication channel or a third-party audit.

3.11.2 Escalation is not permission

The former boolean override was too easy to misread as authorization. The new `ReviewAuthorization` records `authorized_by`, `authority`, `rationale`, and `recorded_on`. It is an escalation or remediation record. A finding remains blocking when its classification is `NON_COMPLIANT`, `REQUIRES_MODIFICATION`, or `INSUFFICIENT_INFORMATION`; the transparency report counts the named authorization without changing that fact.

This is a deliberate difference from institutional governance. The author can make an exception visible, but cannot make the personal “No” disappear by calling it an override.

3.12 Durability and transparency: a hash-based canary

Turner’s framework protects the *durability* of a red-line commitment through procedure: a material modification requires advance notice with a rationale, and an impairment of the Review Body’s capacity must be disclosed [Turner, 2026]. A single practitioner has no review body and no counterparty to notify. The personal analog is therefore cryptographic rather than procedural — it makes weakening the standard visible and auditable, not procedurally gated.

3.12.1 Deterministic registry hashing

`registry_hash` reduces the red-line registry to a single SHA-256 digest over its canonicalized content. Canonicalization sorts the lines by `id`, serializes each fixed payload including narrative carve-outs and typed exemption `ids`, trigger scopes, and required evidence kinds, and dumps the whole as JSON with stable separators. No timestamps or environment inputs enter the payload, so the hash is a pure function of registry *content*: the current seven lines yield the pinned digest `72835fd8...f5aad7`, and any change to a line’s policy semantics changes it. `line_digest` applies the same atom to a single line. This is a cryptographic checksum, not a digital signature: it does not authenticate who issued a statement or make a rewrite impossible.

3.12.2 The canary statement

A `CanaryStatement` is a dated attestation. It binds the registry digest, the set of line `ids` present at issue time, and a tuple of per-line (`id`, `severity`, `digest`) triples, all captured on a stated `issued_on` date. Binding the issue-time severity alongside each digest — not merely the aggregate hash — lets a later check tell *which* line changed and how severe it was when the author last vouched for it. `issue_canary` also enforces a successor guard: when a prior canary is supplied and the hash has drifted, a silent re-issue is refused unless the author supplies a rationale, which is folded into a self-describing successor statement naming the superseded digest and the removed or added `ids`.

3.12.3 Verification and escalation

`verify_canary` compares a prior statement to the live registry. It reports `drift` (any hash change), explicit `removed_ids` and `modified_ids`, and `stale` independently, so a caller sees *how* a canary failed. `intact` requires an unchanged hash, a fresh attestation, complete line-`id` agreement, and internally consistent populated per-line metadata. Freshness is always evaluated against a 180-day window and fails closed: a missing, future-dated, or unparseable date reads as stale rather than as a silent pass. When a removed or modified line was CANARY-grade at issue time, the detail escalates to `CANARY-GRADE LINE ALTERED`, surfacing it above any lower-severity change.

3.12.4 The honest trust model

This is the *pattern* of a warrant canary, not the legal instrument: a warrant canary’s force rests on a legal asymmetry that does not apply to a personal commitment. The statement is forgeable by anyone with write access to the registry, and the author is also its only enforcer. Its evidential force rests entirely on an external prior copy — a git-committed or otherwise independently held statement, checked by someone other than the author. The registry is pre-publication and there is no external verifier. The instrument makes weakening the standard tamper-evident under that condition; it does not prevent it, and that limitation is disclosed rather than papered over.

An external verifier should: obtain a prior statement from outside the author’s write boundary; recompute the current registry hash and per-line metadata; run the freshness and drift check; inspect any successor rationale; and report the result without treating a regenerated current statement as independent evidence. The full command-level procedure is in `docs/VERIFY.md`.

The trust boundary is therefore more important than the hash itself; fig. 19 shows what the mechanism can and cannot signal.

3.12.5 Defensive security context

The nation-state lens is applied here as a defensive threat model for a static private artifact, not as a claim that the repository is an APT-resistant service. The relevant crown jewels are the registry, evidence and review records, prior canary, private scholarship, release boundary, and rendered outputs. A patient adversary could alter a privileged checkout, poison a dependency or renderer, rewrite a same-author canary fixture, or make a source appear more authoritative than it is. The threat model records these paths and their residual limits in `docs/security-threat-model.md`.

NIST’s Secure Software Development Framework, MITRE ATT&CK, and SLSA provide useful vocabularies for provenance, adversary behavior, and future artifact attestation [Souppaya et al., 2022, MITRE, 2026, SLSA Community, 2026]. They are implementation context, not evidence of legal compliance, secure operation, or resistance to a nation-state actor. This project currently demonstrates dependency-light deterministic domain logic, locked development dependencies, local canary and render validation, and explicit external-witness limitations; it does not claim signed provenance, a generated SBOM, hermetic builds, or runtime telemetry.

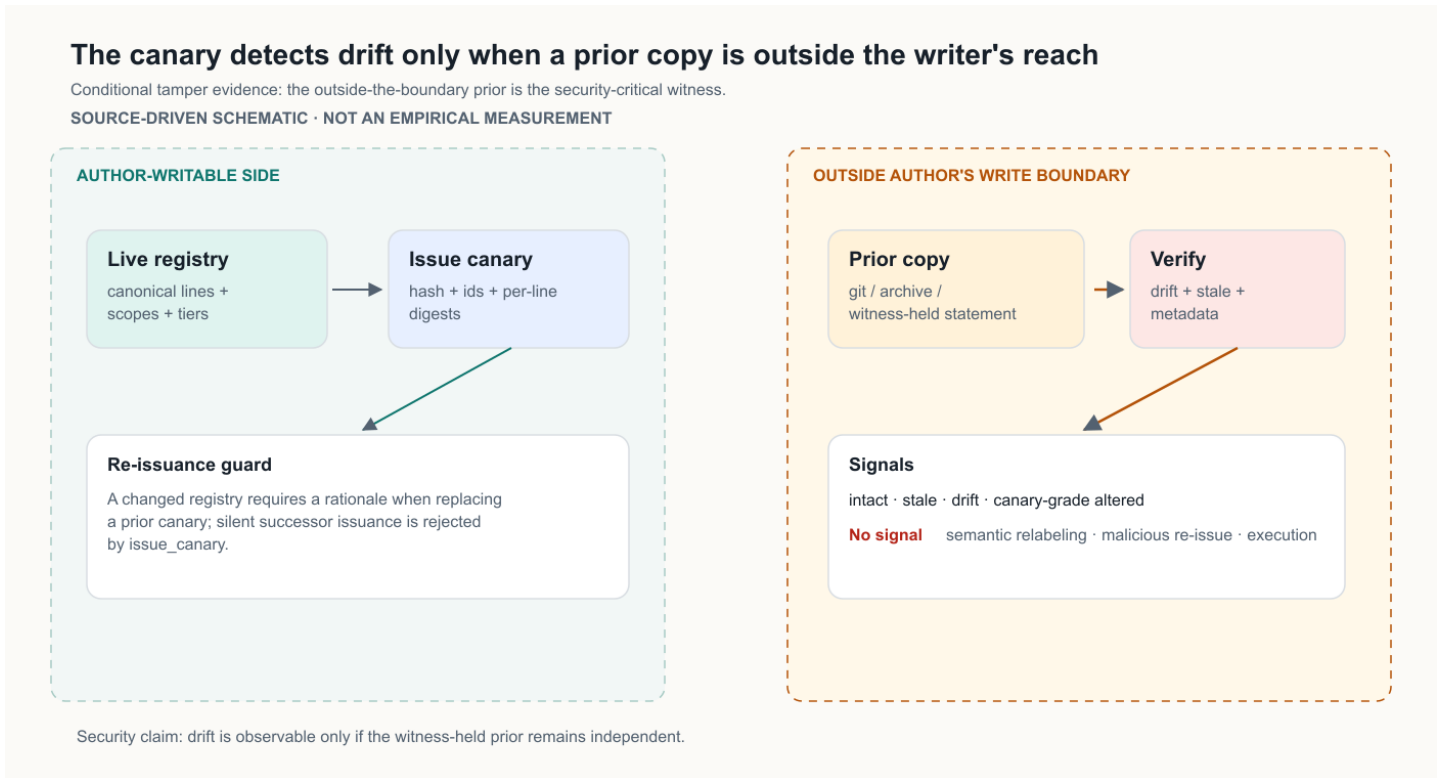


Figure 19: The canary detects drift only when a prior copy is outside the writer’s reach. Trust-boundary schematic for the canary mechanism: issuance binds the registry to an aggregate digest and per-line metadata; verification compares those values with the live registry and checks freshness. The dashed boundary is the critical condition: the prior statement must be held by someone or some system that cannot be rewritten by the same author who edits the registry. A canary cannot detect semantic violations hidden by misleading labels, cannot stop a malicious re-issuance, and cannot enforce an action finding. It is tamper evidence conditional on an independent witness, not prevention or legal protection.

3.13 Evidence, ambiguity, and evaluation

The evaluator is a staged method, not a semantic safety classifier. The first stage asks whether the action can be reviewed at all. `ActionContext` requires nine dimensions: purpose, end use, affected parties, data provenance, legal basis, human control, deployment, downstream transfer, and capability scope. Each must have a `VERIFIED` evidence record, including an explicit supported not-applicable value. Self-assertion, unresolved contradiction, unknown scope, empty scope, or explicit ambiguity returns `INSUFFICIENT_INFORMATION`. The gate is conjunctive over all nine and it says which one it stopped on; [Proposition 3](#) and [fig. 23](#) carry the executed sweep behind that claim.

The evaluator first attempts canonical normalization of the declared scope; malformed or non-ASCII tokens stop the intake rather than becoming new vocabulary. It then checks the evidence gate before matching registry coverage, applying typed exemptions, checking verified evidence for the exemption, and enforcing the deployment tier. Normalization is input hygiene, not a policy verdict. The description is useful for human reading and mismatch hints; it cannot satisfy the scope or evidence gate.

The distinction matters: `OUTSIDE_SCOPE` means the complete intake implicated no current line, while `COMPLIANT` means at least one line was implicated and a verified exemption plus tier requirements satisfied it. A missing legal basis is neither: it is an information stop. The precedence is therefore: intake defect first; then an unexempted line; then a verified but modified or under-tier use; then a fully narrowed implicated line; and only then outside scope. [Proposition 2](#) and [Proposition 4](#) state the same rule formally, each bound to a test that re-derives the branch order from the evaluator source rather than restating it.

This ordering is monotone in severity: the intake gate short-circuits before any policy matching, and the reduction over implicated lines always resolves to the single most severe applicable classification, so a less severe outcome never overrides a more severe one. [fig. 20](#) reads that precedence top to bottom, from the `INSUFFICIENT_INFORMATION` short circuit down through the hard block, the modification requirement, the narrowed compliant result, and the distinct `OUTSIDE_SCOPE` terminal.

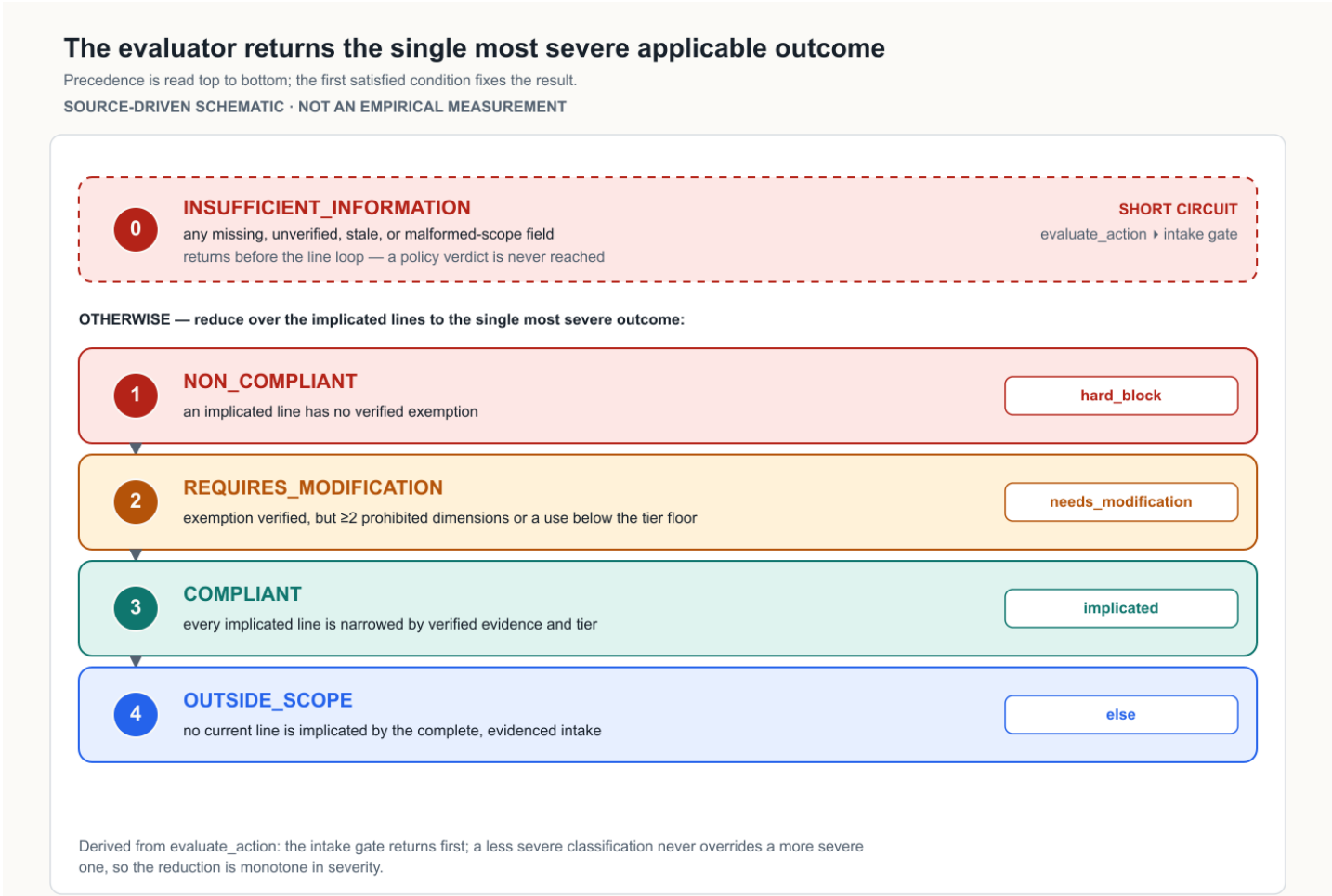


Figure 20: The evaluator returns the single most severe applicable outcome. Outcome-precedence ladder generated from the `evaluate_action` control flow: the intake gate returns `INSUFFICIENT_INFORMATION` before policy matching, after which the reduction over implicated lines resolves to one classification ordered by severity — a hard block dominates a modification requirement, which dominates a compliant result, which dominates `OUTSIDE_SCOPE`. A less severe outcome never overrides a more severe one. The ladder is a control-flow schematic, not empirical evidence that any action was safe, lawful, or correctly described.

Normalization and evidence gates precede every policy verdict

Normalization is input hygiene; verified evidence precedes every policy verdict.

SOURCE-DRIVEN SCHEMATIC · NOT AN EMPIRICAL MEASUREMENT

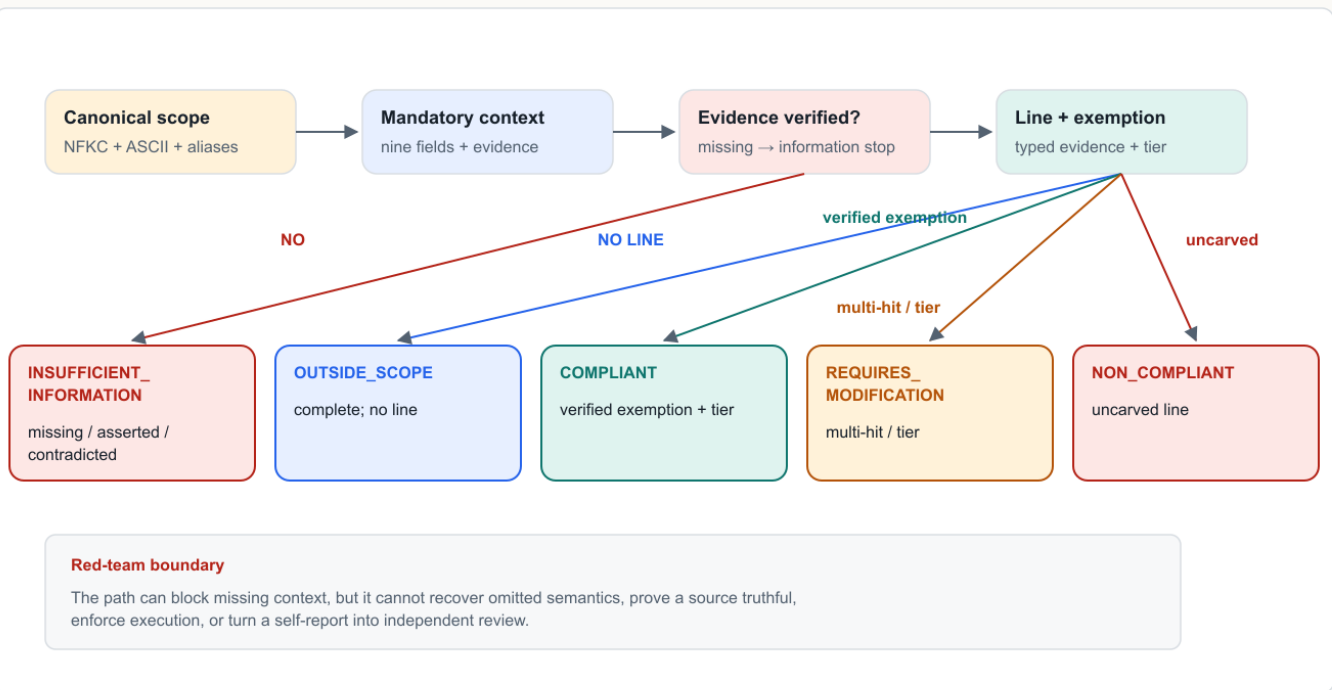


Figure 21: Normalization and evidence gates precede policy verdicts. This deterministic method map shows the reading path from canonical input normalization through mandatory context, required evidence, line coverage, typed exemption, and tier checks. The **INSUFFICIENT_INFORMATION** branch is deliberately upstream of compliance; **OUTSIDE_SCOPE** is a distinct terminal result. The figure is generated from the evaluator contract, not from observed safety data, and the caption's distinction is essential when the visual is read without the surrounding prose.

3.13.1 Exercised outcome coverage

The two figures above are schematics of intended control flow. The five-outcome claim is separately executed rather than drawn: the harness `red_line.analysis.outcome_coverage.run_outcome_coverage` runs a deterministic battery of five `ProposedAction` fixtures — one designed for each classification — through the real `evaluate_action` against the live registry at a fixed review date (2026-07-15). In the current registry all five of the five classifications are reached and every case lands on its intended outcome, with the evaluator’s stable reason codes recorded per case. The same harness honestly reports partial coverage when it should: run against an empty registry, the three implication-dependent outcomes (`COMPLIANT`, `REQUIRES_MODIFICATION`, `NON_COMPLIANT`) become unreachable, and a review date a year later downgrades every fixture’s evidence to stale, collapsing even the compliant case to an information stop. fig. 22 renders the executed report.

The reason codes returned by the executed battery also confirm that each case reaches the intended terminal by the intended route. The unevidenced case stops with `missing_evidence` and `intake_blocked` — the short circuit, not a policy verdict. The out-of-scope case returns the single code `outside_scope`. The compliant case carries both `verified_exemption` and `all_lines_narrowed`, meaning a line was implicated and then narrowed rather than never matched. The modification case reports `multiple_prohibited_dimensions` — a verified exemption undercut by extra prohibited scope on the same line — and the blocked case reports `unexempted_line`. A future refactor that preserved the five terminals but altered the paths to them would surface here as a changed code set.

Reachability is a structural property of the evaluator’s control flow. The battery’s evidence records are harness fixtures, so a complete report is a regression pin on the evaluator — it is not evidence that any real engagement was reviewed, safe, lawful, or correctly described.

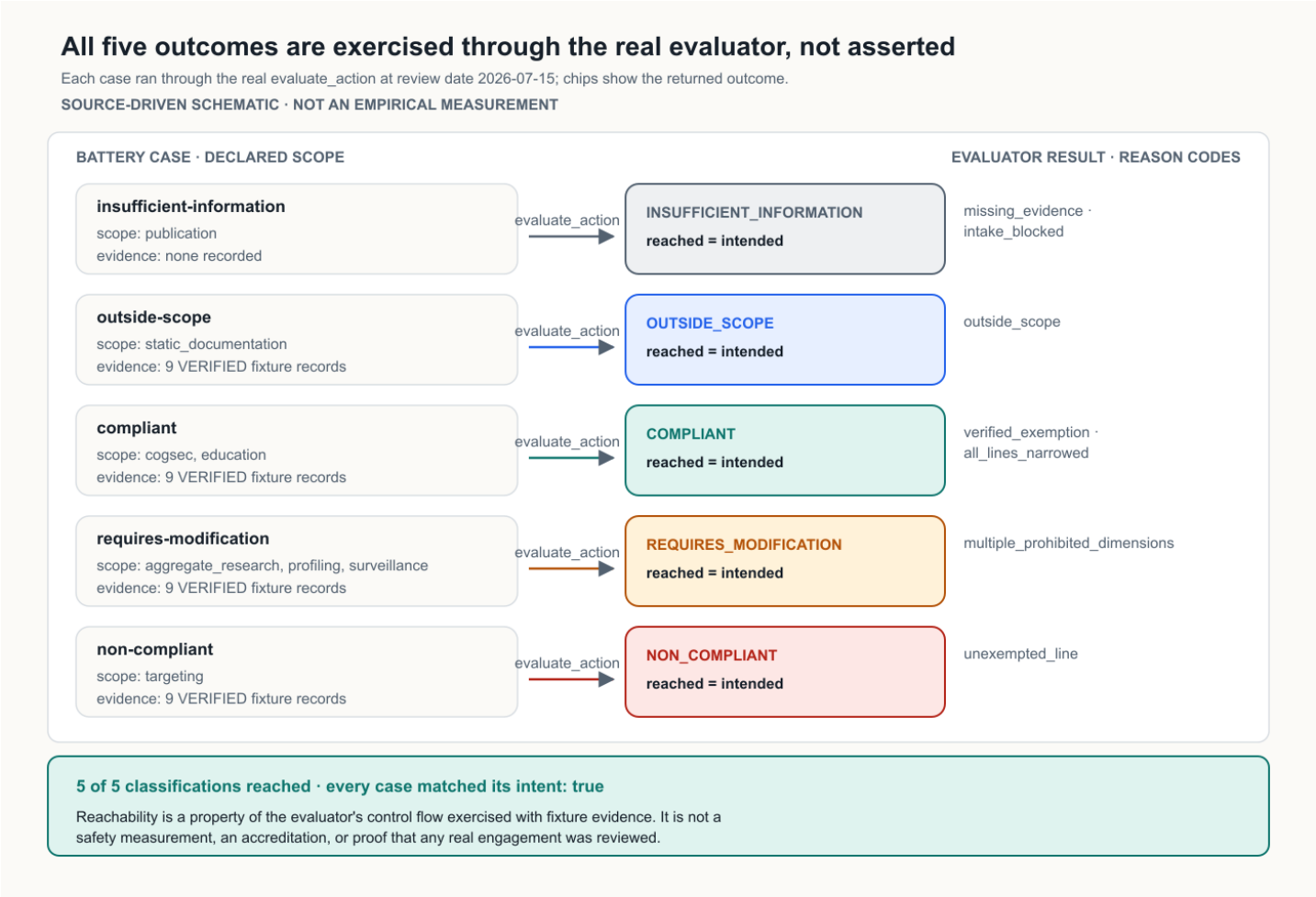


Figure 22: All five outcomes are exercised through the real evaluator, not asserted. Outcome-coverage plate computed by `red_line.analysis.outcome_coverage.run_outcome_coverage`: each battery case runs through the real `evaluate_action` at the fixed review date, and the chip on the right is the classification actually returned with its stable reason codes. All five classifications are reached and every case matches its intended outcome, upgrading the five-outcome claim from a control-flow assertion to an executed, regression-pinned property. The fixture evidence is not real-world verification, and the plate is not a safety measurement.

3.13.2 Residual lexical limitation

Explicit aliases close trivial singular/plural and punctuation near-misses, but they do not infer meaning. A complete-looking action can still misdescribe its capability or evidence. That is why the manuscript calls the output a local auditability result and requires independent witness/review before making a stronger public assurance claim.

The result is also bounded by the current registry vocabulary. A newly named capability, an unfamiliar deployment pattern, or a misleading but complete evidence packet can escape the intended semantic category. This is why the instrument treats `OUTSIDE_SCOPE` as a bounded registry statement and why the release claim remains auditability rather than safety certification.

That vocabulary is small enough to print in full, and fig. 26 does. Thirty-four words decide what this evaluator can see; two of them belong to more than one line. A reader who wants to know what `OUTSIDE_SCOPE` is bounded by can read the whole boundary off one grid rather than take the caveat on trust.

3.14 The instrument stated formally

The preceding sections describe the evaluator in prose. Prose about a decision procedure drifts from the procedure, and the reader has no way to see the drift. This section restates the same objects and the same decision rule as numbered definitions and propositions taken from the shipped code, and binds each proposition to a named test that fails if the two disagree. Nothing here is new policy. It is the existing instrument written so that a mismatch is a red test rather than an unnoticed sentence.

Two limits carry through every result below. Each is a statement about a local program’s behaviour on declared inputs, never about the world those inputs describe; and each is scoped to the registry version pinned by the digest in sec. 3.15, because these are properties of a decision procedure applied to particular content.

3.14.1 The domain objects

Definition 1 (Deployment tier and oversight rank). The deployment tiers are $T = \{\text{hosted}, \text{connected}, \text{air_gapped}\}$, ordered by a retained-oversight rank $\rho : T \rightarrow \{0, 1, 2\}$ with $\rho(\text{air_gapped}) = 0$, $\rho(\text{connected}) = 1$, and $\rho(\text{hosted}) = 2$. A higher rank means more oversight the author still holds after release: the ability to observe the work product, update it, suspend it, or withdraw it.

Definition 2 (Severity grade). The severity grades are $\{\text{canary}, \text{absolute}, \text{strong}\}$. The grade records how a change to the line itself is treated, not how bad a breach would be: a **canary** line’s removal or demotion is itself the reportable event.

Definition 3 (Evidence status). The evidence statuses are $\{\text{VERIFIED}, \text{SELF_ASSERTED}, \text{UNVERIFIED}, \text{CONTRADICTED}\}$. Only **VERIFIED** can support a required dimension. The evaluator does not decide whether a source is true; it records whether a reviewable artifact was checked.

Definition 4 (Intake dimension). The intake dimensions are the nine-element set K :

$$K = \{ \text{purpose}, \text{end_use}, \text{affected_parties}, \text{data_provenance}, \text{legal_basis}, \\ \text{human_control}, \text{deployment}, \text{downstream_transfer}, \text{capability_scope} \}$$

Their enum order is the intake order, and it is the column order of every derived matrix in this paper.

Definition 5 (Evidence record and staleness). An evidence record is a tuple (k, r, s, σ, d) of a kind $k \in K$, a reference r , a summary s , a status σ , and an ISO date d . Given a review date a and a window w (default $w = 180$ days), the record is *stale* when $d > a$ or $a - d > w$. A future-dated record is therefore stale in the same way an expired one is.

Proposition 1 (The staleness boundary is exclusive). Staleness uses the strict comparison $a - d > w$ of [Definition 5](#), so a verified record dated exactly w days before the review date is fresh, and a record dated $w + 1$ days before is stale. The boundary is the same strict-exclusive semantics as `black_line`’s currentness boundary and `golden_line`’s temporal boundary: the window’s last fresh day is age w , and the first stale day is $w + 1$. The threshold w is a review-cadence choice, not a decay law, and a future-dated record is stale regardless of the window.

Definition 6 (Action context). An action context C assigns a string to each of the nine dimensions of [Definition 4](#), and carries a finite tuple of evidence records in the sense of [Definition 5](#) and a finite tuple of declared unknowns. C *supports* a dimension k at review date a when it holds a record of kind k whose status is **VERIFIED** and which is not stale. A dimension whose assigned string is empty or one of **unknown**, **unspecified**, **tbd**, **unclear** is unsupported whatever its records say; **not_applicable** is a permitted value, but it is still only supported when a verified record backs it.

Definition 7 (Scope normalization). A declared token is canonicalized by NFKC normalization, case folding, an ASCII check that raises on anything else, replacement of every non-alphanumeric character by an underscore, collapse of repeated underscores, stripping of leading and trailing underscores, and a final lookup in a reviewed alias table. There is no stemming: a new synonym must be added to the table by hand. $\mathcal{N}(S)$ denotes the canonical token set of a declared scope S .

Definition 8 (Typed exemption). An exemption is a tuple $e = (\text{id}, \text{description}, T_e, R_e, m_e)$ where T_e is a non-empty trigger scope, $R_e \subseteq K$ is the required evidence, and $m_e \in \{\text{any}, \text{all}\}$ is the match mode. For a declared scope S , with $\mathcal{N}$ the normalization of [Definition 7](#),

$$\text{matches}_e(S) = \begin{cases} \mathcal{N}(T_e) \subseteq \mathcal{N}(S) & m_e = \text{all} \\ \mathcal{N}(T_e) \cap \mathcal{N}(S) \neq \emptyset & m_e = \text{any} \end{cases}$$

and e is *satisfied* by a context C at review date a when C supports every $k \in R_e$. Matching is a declaration; satisfaction is the condition that can narrow a line.

Definition 9 (Red line). A red line is an eleven-field record: a slug id, a title, a first-person standard beginning I, a rationale, a coverage scope, narrative carve-out clauses, a tier floor `max_tier`, a severity of [Definition 2](#), the person stating it, the ISO date it was stated, and a tuple of typed exemptions. A line ℓ covers a declared scope S when $\mathcal{N}(\text{scope}_\ell) \cap \mathcal{N}(S) \neq \emptyset$. The carve-out clauses are prose for a reader; only the typed exemptions execute.

Definition 10 (Proposed action and effective scope). A proposed action is a tuple (description, S, C, t, amb) of free text, a declared scope, a context, a tier $t \in T$ as ranked in [Definition 1](#), and an ambiguity flag. Coverage and exemption matching both run against the *effective* scope $E = \mathcal{N}(S) \cup \{t\}$: the tier value is itself a token, so a line or exemption may name a tier in its scope. The description is never part of E .

Definition 11 (Classification). The classifications are the five-element set C :

$$C = \{ \text{COMPLIANT}, \text{REQUIRES_MODIFICATION}, \text{NON_COMPLIANT}, \\ \text{OUTSIDE_SCOPE}, \text{INSUFFICIENT_INFORMATION} \}$$

`OUTSIDE_SCOPE` is a finding of absence after a complete inspection; `INSUFFICIENT_INFORMATION` is a refusal to inspect. Neither is permission.

Definition 12 (Intake defect). An action has an intake defect at review date a when any of the following holds: a declared token fails normalization; some dimension is unsupported in the sense of [Definition 6](#); some dimension holds a record that is unresolved (self-asserted, unverified, or contradicted in the sense of [Definition 3](#), with no verified record standing in its place) or stale; the context declares an unknown; the normalized scope is empty or contains one of the markers `unknown`, `unspecified`, `tbd`, `unclear`; or the ambiguity flag is set.

Definition 13 (Tier floor). Each line declares a floor `max_tier`. An action at tier t is *below the floor* of line ℓ when $\rho(t) < \rho(\text{max_tier}_\ell)$, with ρ the rank function of [Definition 1](#) — it retains strictly less oversight than the line requires. The field name reads as a ceiling and behaves as a floor on retained oversight; the code, not the name, is the definition.

Definition 14 (Strictness order). On the three verdicts a fully evidenced, line-implicating intake can reach, strictness is ranked `COMPLIANT` < `REQUIRES_MODIFICATION` < `NON_COMPLIANT`. The two remaining classifications carry no strictness rank: an intake stop is upstream of policy and an out-of-scope result implicates nothing, so ranking either against these three would be arbitrary. The implementing predicate raises rather than ranking them.

3.14.2 The decision rule

Proposition 2 (The intake gate precedes policy). If an action has an intake defect in the sense of [Definition 12](#) at review date a , then for every registry — including the empty one — the result is `INSUFFICIENT_INFORMATION`, the implicated-line tuple is empty, and the reason codes include `intake_blocked`. No line is consulted, so no registry content can change the outcome. An action is a proposed action in the sense of [Definition 10](#); the intake gate inspects its context and evidence, not its description.

Proposition 3 (The intake gate is conjunctive over all nine dimensions). Take a baseline action the live registry classifies `COMPLIANT` with a fresh `VERIFIED` record for each of the nine dimensions. Degrading exactly one record — removing it, or setting its status to self-asserted, unverified, or contradicted, or dating it outside the freshness window — withdraws that result. Across the nine dimensions and the five degradations, all forty-five executed evaluations return `INSUFFICIENT_INFORMATION`, and each of the forty-five names exactly the degraded dimension and no other as blocking. No dimension is decorative, and the stop signal identifies the field it stopped on.

Proposition 4 (One verdict, most severe first). Given a defect-free intake, the evaluator reduces over the lines that cover the effective scope and returns exactly one classification, determined in this order: if any covering line has no satisfied exemption in the sense of [Definition 8](#), `NON_COMPLIANT`; else if any satisfied exemption sits below its line's tier floor of [Definition 13](#) or coincides with two or more coverage hits on the same line, `REQUIRES_MODIFICATION`; else if any line was covered, `COMPLIANT`; else `OUTSIDE_SCOPE`. The four cases are exhaustive and mutually exclusive, and a less severe outcome never displaces a more severe one. A line is a red line in the sense of [Definition 9](#); its coverage scope, tier floor, and typed exemptions are the fields this rule reads.

Proposition 5 (All five classifications are reachable). Each of the five classifications of [Definition 11](#) is returned by at least one action against the live registry at a fixed review date. Reachability is a property of the control flow, not evidence that any real engagement was classified correctly; the same harness reports partial coverage honestly when run against an empty registry, where the three implication-dependent outcomes become unreachable.

Proposition 6 (No exemption narrows a line without verified evidence). An exemption narrows its line only when it both matches the effective scope and is satisfied by the context. On the live registry every one of the sixteen typed exemptions requires at least two evidence kinds, so no matching declaration alone can clear a line. The detector that would report a zero-evidence exemption returns nothing on the live registry and fires on a planted one, which is what makes the empty result informative rather than merely absent.

Proposition 7 (ALL-mode triggers cannot be reached by one token). For an **any**-mode exemption every single trigger token matches; for an **all**-mode exemption with more than one trigger token, no single token matches and only the full trigger set does. The live registry carries thirteen **any**-mode and three **all**-mode exemptions, and every **all**-mode exemption has exactly two trigger tokens. Run through the real evaluator against an anchor from the exemption’s own line, each **all**-mode exemption’s line is **NON_COMPLIANT** when one trigger token is declared and **COMPLIANT** when both are — fifty-eight executed evaluations, every row behaving as its mode requires.

Proposition 8 (Reducing oversight never softens a verdict). Fix a declared scope and a fully evidenced context, and vary only the tier along `hosted` $\rightarrow$ `connected` $\rightarrow$ `air_gapped`. Strictness in the sense of [Definition 14](#) never decreases. Every scope keyword of every current line at all three tiers is one hundred and eight executed evaluations with zero inversions. An unexempted covering line is **NON_COMPLIANT** at every tier; a satisfied exemption below its floor is **REQUIRES_MODIFICATION** rather than **COMPLIANT**.

Proposition 9 (Normalization is closed, and failure stops at two layers). $\mathcal{N}$ is idempotent: $\mathcal{N}(\mathcal{N}(S)) = \mathcal{N}(S)$ for every scope it accepts. A token that is not ASCII after NFKC normalization is rejected rather than canonicalized, and the rejection is enforced twice. The action constructor raises, so an ordinary caller cannot build the action at all. Should such a token be written into the frozen record past the constructor, the evaluator normalizes defensively a second time and returns **INSUFFICIENT_INFORMATION** with an `invalid_scope` reason code and an empty normalized scope. A homoglyph or full-width spelling therefore cannot enter the vocabulary as a new token, and cannot pass as an unmatched one either.

Proposition 10 (Reason codes are a stable, duplicate-free audit surface). Every assessment carries reason codes drawn from a closed vocabulary, appended in evaluation order and never repeated; the assessment record rejects a duplicated code at construction. Human-readable reasons can be reworded without changing the codes, which is what lets a downstream consumer regression-test the *route* to a verdict rather than the terminal alone.

3.14.3 The report envelope

A classification word is the *last* step of a review, not the whole of it. **COMPLIANT** is a projection of a derivation that also holds the reasons trail, the matched exemption if any, the evidence sweep, and the authorization arm — and a word that travels without its derivation is exactly the safe-looking projection this instrument must not let harden into the state. The envelope is the transport contract that keeps the two attached: the word travels only alongside a digest pointer to the complete native finding, and the instrument’s non-claims travel inside the same record.

Definition 15 (Report envelope). The report envelope is the frozen record $v = (\text{schema_version}, \text{line_id}, \text{subject_id}, \text{review_date}, \text{native_status}, \text{report_ref})$ with exactly those ten fields, in order, exported under the schema string `line.report-envelope/1.0`. `native_status` is this line’s own classification word from [Definition 11](#); it is one instrument’s word in that instrument’s vocabulary, never to be compared, ranked, averaged, or merged across lines. `report_ref` is the SHA-256 of the canonical native finding (`red-line.report/1.0`), which serializes the complete derivation — including the authorization arm as an explicit `null` when it is absent, so an unreviewed arm is distinguishable from an empty one. Sibling instruments export the same shape by publishing the same schema string, never by importing one another.

Proposition 11 (The envelope points, never reinterprets). For every finding f the evaluator returns, `finding_envelope(f)` produces the [Definition 15](#) and satisfies `envelope_matches_finding(envelope, f)`: the digest pointer, the review date, the registry digest, and the classification word all agree with the finding they were exported from, and editing any checked field afterwards makes the check return false. The envelope adds nothing the finding does not determine except the caller-supplied subject and snapshot references, which are stored, not verified. A matching envelope attests that an archived pair is unedited; it does not certify the finding true, the action safe, or the review well aimed.

3.14.4 What binds each proposition

Every row names a test in `tests/integration/test_formalism_bindings.py` that re-derives the proposition’s content from the code and then asserts the manuscript states it. Corrupting the sentence reddens the row; corrupting the code reddens the derivation inside it.

Proposition	Claim in one line	Verifying test
Proposition 1	staleness uses strict comparison; the window edge is exclusive	<code>test_staleness_exclusive_boundary_is_derived_from_the_code</code>
Proposition 2	a defect stops the evaluation before any line is read	<code>test_intake_precedence_holds_against_every_registry</code>
Proposition 3	all nine dimensions are load-bearing and the stop is localized	<code>test_evidence_conjunction_matches_the_executed_sweep</code>

Proposition	Claim in one line	Verifying test
Proposition 4	one verdict, resolved most-severe-first	<code>test_outcome_precedence_is_exhaustive_and_ordered</code>
Proposition 5	all five classifications are reached	<code>test_outcome_reachability_matches_the_executed_battery</code>
Proposition 6	a matching trigger alone never narrows a line	<code>test_exemption_evidence_floor_is_derived_from_the_registry</code>
Proposition 7	ALL-mode needs every trigger token	<code>test_trigger_mode_counts_match_the_executed_probe</code>
Proposition 8	dropping oversight never softens a verdict	<code>test_tier_monotonicity_numbers_match_the_executed_sweep</code>
Proposition 9	normalization is idempotent and fails closed	<code>test_normalization_closure_is_executed_not_asserted</code>
Proposition 10	codes are closed, ordered, and duplicate-free	<code>test_reason_code_vocabulary_is_closed_and_duplicate_free</code>
Proposition 11	the envelope agrees with its finding, and any edit is visible	<code>test_envelope_pointer_agreement_is_executed_not_asserted</code>

The two propositions that are hardest to see from the code alone have their own plates. fig. 23 renders the forty-five perturbations behind Proposition 3, and fig. 24 renders the fifty-eight probes behind Proposition 7.

3.14.5 What the formalism does not establish

A proposition here says what the program returns for declared inputs. It does not say that the declaration is honest, that the verified record is true, that the nine dimensions exhaust what matters, or that the registry names the right boundaries. Proposition 8 is a consistency property of a decision procedure, not a claim that a hosted deployment is safe. Proposition 3 shows that the gate is strict, which is a different thing from showing that strictness is well aimed: an action can satisfy all nine dimensions with plausible false records and reach COMPLIANT, and the formalism has nothing to say about that case. Those limits are developed in sec. 3.17.

Degrade one intake dimension and the compliant result is withdrawn

One dimension degraded per cell against an otherwise unchanged compliant baseline at review date 2026-07-15.

SOURCE-DRIVEN SCHEMATIC · NOT AN EMPIRICAL MEASUREMENT

BASELINE scope cogsec + education · tier HOSTED · nine VERIFIED records → COMPLIANT

DEGRADED DIMENSION	ABSENT	SELF-ASSERTED	UNVERIFIED	CONTRADICTED	STALE	BLAMED ONLY IT
purpose	INSUFF. INFO missing	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + stale	YES 5/5
end_use	INSUFF. INFO missing	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + stale	YES 5/5
affected_parties	INSUFF. INFO missing	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + stale	YES 5/5
data_provenance	INSUFF. INFO missing	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + stale	YES 5/5
legal_basis	INSUFF. INFO missing	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + stale	YES 5/5
human_control	INSUFF. INFO missing	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + stale	YES 5/5
deployment	INSUFF. INFO missing	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + stale	YES 5/5
downstream_transfer	INSUFF. INFO missing	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + stale	YES 5/5
capability_scope	INSUFF. INFO missing	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + unresolved	INSUFF. INFO missing + stale	YES 5/5

45 of 45 perturbations withdrew the compliant result · 45 of 45 named only the degraded dimension

Every stop is INSUFFICIENT_INFORMATION — an information gap, not a finding against the work.

The sweep shows that the local gate is conjunctive over its nine declared dimensions and that it names the field it stopped on. It does not show that a VERIFIED record is true, that these nine dimensions are the right ones, or that any real intake was reviewed.

Figure 23: Degrade one intake dimension and the compliant result is withdrawn. Single-dimension perturbation sweep computed by `red_line.analysis.evidence_sensitivity.run_evidence_sensitivity`: a compliant baseline is re-run through the real `evaluate_action` with exactly one of its nine verified records removed, downgraded, or aged past the freshness window. Every cell reports the classification actually returned and the reason codes it raised, and the trailing column confirms the gate named only the degraded dimension. Conjunctive behaviour is a property of the local gate — not evidence that a verified record is true, that these are the right nine dimensions, or that any real intake was reviewed.

One convenient word cannot reach an ALL-mode exemption

Each row runs one exemption's trigger tokens singly and then together through the real evaluator at review date 2026-07-15.

SOURCE-DRIVEN SCHEMATIC · NOT AN EMPIRICAL MEASUREMENT

RED LINE · TYPED EXEMPTION anchor token in parentheses	MODE	ONE TRIGGER TOKEN declared beside the anchor		ALL TRIGGER TOKENS declared beside the anchor	
cogsec-integrity (cogsec) defensive-cognitive-security	ANY	3 of 3 match	COMPLIANT	all 3 match	COMPLIANT
epistemic-education	ANY	3 of 3 match	COMPLIANT	all 3 match	COMPLIANT
downstream-transfer (downstream) open-source-no-end-use-binding	ANY	2 of 2 match	COMPLIANT	all 2 match	COMPLIANT
vetted-flow-down	ALL	0 of 2 match	NON-COMPLIANT	all 2 match	COMPLIANT
dual-use-ablation (model_release) ablated-task-specific	ANY	2 of 2 match	COMPLIANT	all 2 match	COMPLIANT
methods-not-weights	ANY	3 of 3 match	COMPLIANT	all 3 match	COMPLIANT
retained-oversight	ANY	2 of 2 match	COMPLIANT	all 2 match	COMPLIANT
open-science-good-faith (benchmark_claim) clearly-labeled-preliminary	ANY	3 of 3 match	COMPLIANT	all 3 match	COMPLIANT
stated-withholding	ANY	2 of 2 match	COMPLIANT	all 2 match	COMPLIANT
provenance-and-consent (consent) own-consented-data	ANY	3 of 3 match	COMPLIANT	all 3 match	COMPLIANT
public-open-synthetic-data	ANY	3 of 3 match	COMPLIANT	all 3 match	COMPLIANT
s1-human-control-force (autonomous_weapon) adjacent-force-support	ANY	6 of 6 match	COMPLIANT	all 6 match	COMPLIANT
defensive-alerting-human-control	ALL	0 of 2 match	NON-COMPLIANT	all 2 match	COMPLIANT
s2-untargeted-profiling (biometric_id) aggregate-research	ANY	2 of 2 match	COMPLIANT	all 2 match	COMPLIANT
identified-lawful-subject	ALL	0 of 2 match	NON-COMPLIANT	all 2 match	COMPLIANT
opt-in-personal-analytics	ANY	2 of 2 match	COMPLIANT	all 2 match	COMPLIANT

16 typed exemptions · 13 ANY · 3 ALL · 58 executed evaluate_action runs · rows behaving as their mode requires: 16 of 16

An ALL-mode exemption cannot be reached by naming one convenient word; an ANY-mode one can.

Every probe carries a fully evidenced fixture intake, so the only variable is which trigger tokens the action declares. A matched trigger is a declaration and never proof: the exemption still cannot narrow its line until the typed evidence it demands is VERIFIED.

Figure 24: One convenient word cannot reach an ALL-mode exemption. Trigger-semantics probe computed by `red_line.analysis.trigger_semantics.run_trigger_semantics`: every typed exemption is run through the real `evaluate_action` twice, once declaring a single trigger token beside an anchor from its own line's coverage scope and once declaring the whole trigger set. ANY-mode rows match on every single token; ALL-mode rows match on none and clear their line only when every trigger token is present. A matched trigger is a declaration and never proof — the typed evidence must still be verified, and the plate reports match semantics rather than whether a declaration describes the work honestly.

3.15 The red-line registry

This is the beacon: the author’s personal security boundary and explicit No document at version 0.3.0. The registry is first-person, dated, revisable, and non-exhaustive. It is not a universal ethics code and not a claim made by any AI system that assisted with its preparation.

The current registry contains seven lines, including two CANARY-grade lines (formalized as [Definition 9](#) and [Definition 8](#)). The canonical registry digest is:

72835fd81d1f7ecf70f47b1e0061cd56c385273dd846879ab639225913f5aad7

Each line has a human-readable standard, rationale, coverage dimensions, narrowing clauses, structured exemptions, required evidence, a tier floor, severity, author, and date. The narrative clauses explain the boundary to a reader; the typed exemptions are the only executable narrowing conditions. Adding a token such as **vetted** or **consented** cannot establish vetting or consent.

The authoritative machine-readable source is `src/red_line/registry/lines.py`. The canary includes exemption semantics in its canonical payload, so changing an exemption or its evidence requirement is a detectable registry change and must be accompanied by a successor rationale.

Because the registry is enumerable data, its narrowing structure can be stated as derived numbers rather than described qualitatively. The analysis module `red_line.analysis.registry_metrics` computes the full exemption $\times$ evidence-kind coverage matrix from the live registry: the seven lines carry 16 typed exemptions that together declare 37 evidence requirements across the nine intake dimensions. Affected parties is the most-demanded dimension (six exemptions require it), followed by purpose, legal basis, and capability scope (five each); end use and deployment are the least demanded (two each). Thirteen exemptions match their trigger scope with ANY semantics and three require ALL trigger tokens to be declared. [fig. 25](#) renders that matrix directly from the analysis code; every filled cell is a precondition for narrowing, not proof that the required evidence is true. The fuller structural profile — severity and tier-floor distributions, scope overlap points, and per-line composition — is derived in [sec. 3.16](#).

3.15.1 s1-human-control-force — Human control over force and harm-capable systems

[CANARY]

Standard: I will not build, tune, or knowingly supply a component to a system that selects and engages targets for force without appropriate human control over each engagement.

Rationale: Turner Standard 1: force-application without an identifiable, accountable human decision-maker removes the moral circuit-breaker. Applies whether I provide targeting directly or as a component.

Coverage dimensions: `targeting, weapons, lethality, force, kinetic, autonomous_weapon`

Does not restrict: defensive-only alerting with a human in the loop

Does not restrict: logistics, translation, maintenance, or research and development

Does not restrict: intelligence analysis reviewed by an accountable human

Typed exemptions and required evidence:

- `defensive-alerting-human-control`: Defensive-only alerting with an accountable human decision-maker; `human_control`, `end_use`.
- `adjacent-force-support`: Logistics, translation, maintenance, research, or intelligence analysis; `purpose`, `human_control`.

Max tier: hosted

Severity: CANARY

Stated by: Daniel Ari Friedman

Stated on: 2026-07-15

3.15.2 s2-untargeted-profiling — No untargeted profiling or mass surveillance

[CANARY]

Standard: I will not build tooling whose purpose is to convert bulk data into individualized intelligence on persons not already identified as subjects of a specific, lawful, individualized process.

Rationale: Turner Standard 2: bulk-to-individual inference on unnamed persons is the engine of mass surveillance. Demographic-, origin-, or belief-based initiation is prohibited outright.

Coverage dimensions: `surveillance, profiling, bulk_data, biometric_id, dragnet, tracking`

Every narrowing of a line names the evidence that must be verified first

Filled cells are typed evidence requirements computed from the live registry, never hand-authored.

SOURCE-DRIVEN SCHEMATIC · NOT AN EMPIRICAL MEASUREMENT

LINE · EXEMPTION	purpose	end use	affected parties	data provenance	legal basis	human control	deployment	downstream transfer	capability scope	REQUIRED · MODE
cogsec-integrity defensive-cognitive-security	●	●	○	○	○	●	○	○	○	3 ANY
epistemic-education	●	○	●	○	○	○	○	○	○	2 ANY
downstream-transfer open-source-no-end-use-binding	○	○	○	○	○	○	○	●	●	2 ANY
vetted-flow-down	○	○	●	○	●	○	○	●	○	3 ALL
dual-use-ablation ablated-task-specific	○	○	○	○	○	○	●	○	●	2 ANY
methods-not-weights	○	○	○	○	○	○	○	●	●	2 ANY
retained-oversight	○	○	○	○	○	●	●	○	○	2 ANY
open-science-good-faith clearly-labeled-preliminary	●	○	○	○	○	○	○	○	●	2 ANY
stated-withholding	○	○	○	○	○	○	○	●	●	2 ANY
provenance-and-consent own-consented-data	○	○	●	●	●	○	○	○	○	3 ANY
public-open-synthetic-data	○	○	○	●	●	○	○	○	○	2 ANY
s1-human-control-force adjacent-force-support	●	○	○	○	○	●	○	○	○	2 ANY
defensive-alerting-human-control	○	●	○	○	○	●	○	○	○	2 ALL
s2-untargeted-profiling aggregate-research	●	○	●	●	○	○	○	○	○	3 ANY
identified-lawful-subject	○	○	●	○	●	○	○	○	○	2 ALL
opt-in-personal-analytics	○	○	●	●	●	○	○	○	○	3 ANY
exemptions demanding each kind	5	2	6	4	5	4	2	4	5	

Computed from the live registry: 16 typed exemptions declare 37 evidence requirements across the nine intake dimensions. A filled cell is a precondition for narrowing, not proof the evidence is true; the matrix is structure, not a safety score.

Figure 25: Every narrowing of a line names the evidence that must be verified first. Registry-derived exemption evidence-requirement matrix computed by `red_line.analysis.registry_metrics.exemption_evidence_matrix`: each row is one typed exemption grouped under its red line, each column one of the nine intake dimensions, and a filled cell means the exemption can narrow its line only when a **VERIFIED** record of that kind is present. The bottom row counts how many exemptions demand each dimension. The matrix describes the structural shape of the author's boundaries, not their moral weight; a satisfied requirement is a locally recorded condition, not independent truth or a safety score.

Does not restrict: aggregate research producing no individualized output

Does not restrict: analysis of an already-identified, lawfully specified subject

Does not restrict: consented, opt-in personal analytics

Typed exemptions and required evidence:

- **aggregate-research:** Aggregate research producing no individualized output; `purpose`, `affected_parties`, `data_provenance`.
- **identified-lawful-subject:** Analysis of an already-identified, lawfully specified subject; `affected_parties`, `legal_basis`.
- **opt-in-personal-analytics:** Consent-based opt-in personal analytics; `affected_parties`, `data_provenance`, `legal_basis`.

Max tier: connected

Severity: CANARY

Stated by: Daniel Ari Friedman

Stated on: 2026-07-15

3.15.3 dual-use-ablation — Scoped release of dual-use models

Standard: I will not release a proprietary or handed-off model beyond my recall (air-gapped) while it retains dangerous dual-use capability that has not been ablated below a repurposing-cost threshold.

Rationale: Turner Tier 3: for work released beyond monitoring, the cost of repurposing a scoped model should exceed the value of doing so.

Coverage dimensions: `model_release`, `weights`, `handoff`

Does not restrict: release of task-specific models with capability removed

Does not restrict: open publication of methods, papers, or benchmarks

Does not restrict: hosted or connected tiers under retained oversight

Typed exemptions and required evidence:

- **ablated-task-specific:** Task-specific release with dangerous capability removed; required evidence: `capability_scope`, `deployment`.
- **methods-not-weights:** Open methods, papers, or benchmark publication without dangerous weights; required evidence: `capability_scope`, `downstream_transfer`.
- **retained-oversight:** Hosted or connected work under retained oversight; required evidence: `deployment`, `human_control`.

Max tier: air-gapped

Severity: STRONG

Stated by: Daniel Ari Friedman

Stated on: 2026-07-15

3.15.4 cogsec-integrity — Cognitive security strengthens, never degrades, the epistemic commons

Standard: I will not build cognitive-security tooling whose function is to manufacture deception, run covert influence operations, or degrade a population's shared ability to reason.

Rationale: My cognitive-security work is defensive by definition: it strengthens information ecosystems. Weaponized persuasion inverts that mission.

Coverage dimensions: `influence_ops`, `disinformation`, `manipulation`, `propaganda`, `deception`, `cogsec`

Does not restrict: detection, red-teaming, or defensive analysis of influence operations

Does not restrict: education, media-literacy, or transparency tooling

Typed exemptions and required evidence:

- **defensive-cognitive-security:** Detection, red-teaming, or defensive analysis; `purpose`, `end_use`, `human_control`.
- **epistemic-education:** Education, media literacy, or transparency tooling; `purpose`, `affected_parties`.

Max tier: connected
Severity: ABSOLUTE
Stated by: Daniel Ari Friedman
Stated on: 2026-07-15

3.15.5 provenance-and-consent — Provenance and consent for data and identity

Standard: I will not train, evaluate, or ship on data acquired without a lawful basis and, where persons are involved, without consent or a legitimate public-interest basis.

Rationale: Turner’s ‘acquisition’ clause: any process by which person data enters my systems is covered regardless of how a source labels it.

Coverage dimensions: data_acquisition, scraping, pii, consent, provenance

Does not restrict: public-domain, openly-licensed, or synthetic data

Does not restrict: my own or explicitly-consented personal data

Typed exemptions and required evidence:

- public-open-synthetic-data: Public-domain, openly licensed, or synthetic data; data_provenance, legal_basis.
- own-consented-data: Author-owned or explicitly consented personal data; data_provenance, legal_basis, affected_parties.

Max tier: hosted
Severity: STRONG
Stated by: Daniel Ari Friedman
Stated on: 2026-07-15

3.15.6 open-science-good-faith — Open-science claims are honest and reproducible

Standard: I will not publish a result, metric, or capability claim I have not verified against a reproducible artifact, nor let proprietary interest launder an unverified claim into the public record.

Rationale: My public research graph is a beacon only if its claims bind to code that runs. Overclaiming corrodes the commons I am trying to build.

Coverage dimensions: publication, benchmark_claim, metric, reproducibility, provenance

Does not restrict: clearly-labeled preliminary, hypothesis, or negative results

Does not restrict: withholding proprietary detail while stating what is withheld

Typed exemptions and required evidence:

- clearly-labeled-preliminary: Clearly labeled preliminary, hypothesis, or negative result;
required evidence: purpose, capability_scope.
- stated-withholding: Withheld proprietary detail explicitly identified;
required evidence: capability_scope, downstream_transfer.

Max tier: air-gapped
Severity: STRONG
Stated by: Daniel Ari Friedman
Stated on: 2026-07-15

3.15.7 downstream-transfer — No knowing transfer to a violating end use

Standard: I will not sell, license, or hand off work where I know or have substantial reason to believe it will be re-transferred to an end use that violates any of these red lines.

Rationale: Turner’s downstream-transfer and ‘knows or has substantial reason to believe’ clauses: flow-down of the substantive boundary, not just the box.

Coverage dimensions: resale, sublicense, integration, downstream, handoff

Does not restrict: transfer to a vetted end user with a flow-down commitment

Does not restrict: fully open-source release under a license that itself binds no end use

Typed exemptions and required evidence:

- **vetted-flow-down:** Transfer to a vetted end user with a flow-down commitment;
required evidence: `downstream_transfer`, `legal_basis`, `affected_parties`.
- **open-source-no-end-use-binding:** Open-source release whose license does not bind end use;
required evidence: `downstream_transfer`, `capability_scope`.

Max tier: connected

Severity: STRONG

Stated by: Daniel Ari Friedman

Stated on: 2026-07-15

The executable records, rather than this prose alone, determine whether an exemption is satisfied. A complete intake can still be `NON_COMPLIANT`, and a complete action with no matching line is `OUTSIDE_SCOPE`, not a safety finding.

3.16 Registry composition as derived data

The registry chapter above states what each line refuses; this section states what the registry *is*, structurally, using only numbers computed from the live registry object. Prose descriptions of a versioned artifact drift; a count computed at read time from `src/red_line/registry/lines.py` cannot. The analysis subpackage `red_line.analysis.registry_metrics` exists for exactly this purpose: every function in it is a pure, zero-I/O summarizer over the same in-memory `RedLine` tuple the evaluator consumes, with deterministic sorted output and fail-closed input validation. None of its outputs is a safety score. A count describes the shape of a boundary — how many tokens it covers, how many conditions narrow it, what evidence those conditions demand — not the strength, correctness, or moral standing of the person committing to it.

3.16.1 Severity and tier floors

The seven lines divide by severity into two **CANARY** lines, one **ABSOLUTE** line, and four **STRONG** lines (`severity_distribution`). Their deployment-tier floors divide into two lines whose floor is `hosted`, three at `connected`, and two at `air_gapped` (`tier_floor_distribution`). The two gradings are deliberately orthogonal: severity records how much process a change to the line itself requires, while the tier floor records the most permissive deployment tier at which the line’s exemptions can operate at all. The registry exhibits that orthogonality directly — the two **CANARY** lines sit at different tier floors (`hosted` and `connected`), and the two `air_gapped` floors belong to **STRONG** lines, not to the most change-protected ones.

3.16.2 Scope vocabulary and overlap points

The seven lines declare 36 scope-token slots over 34 distinct canonical tokens (`scope_token_frequency`). Exactly two tokens are shared between lines, each by two lines: `handoff` (declared by both `dual-use-ablation` and `downstream-transfer`) and `provenance` (declared by both `provenance-and-consent` and `open-science-good-faith`). These are the registry’s only structural overlap points: an action whose declared scope includes one of them implicates two boundaries in a single evaluation, and the severity-monotone reduction described in sec. 3.13 then resolves the pair to the single most severe applicable classification. The near-total disjointness of the remaining 32 tokens is a design consequence of the standalone lines, not an accident — each line names its own coverage rather than inheriting a shared taxonomy.

fig. 26 draws that vocabulary in full — every token against every line — so the overlap is visible as a property of the whole grid rather than as two names in a sentence. The figure also reads out what the evaluator actually returns for each shared token, which is the part that matters operationally: `handoff` is declared by `dual-use-ablation`, whose `retained-oversight` exemption is satisfied by the sweep’s fully evidenced hosted intake, and by `downstream-transfer`, whose exemptions are not — and the returned verdict is `NON_COMPLIANT`. One line’s verified exemption does not clear a token the other line also claims.

The exemption named there is the one the evaluator reports applying, not the one a reader might expect from the line’s carve-out prose: a scope of exactly `handoff` triggers no methods-publication exemption, because that exemption’s trigger tokens are `methods`, `paper`, and `benchmark`. The distinction is the whole point of typed triggers, so the id in the sentence above is re-derived from the assessment’s own reason strings by `tests/integration/test_manuscript_composition_binding.py` rather than read off the carve-out list.

3.16.3 Per-line structure

The table below is computed by `red_line.analysis.registry_metrics.line_summaries` from the live registry. Scope is the count of canonical coverage tokens; carve-outs are narrative clauses; exemptions are the typed, executable narrowing conditions, split by trigger match mode.

Line	Severity	Tier floor	Scope	Carve-outs	Exemptions	ANY / ALL
<code>cogsec-integrity</code>	absolute	connected	6	2	2	2 / 0
<code>downstream-transfer</code>	strong	connected	5	2	2	1 / 1
<code>dual-use-ablation</code>	strong	air_gapped	3	3	3	3 / 0
<code>open-science-good-faith</code>	strong	air_gapped	5	2	2	2 / 0
<code>provenance-and-consent</code>	strong	hosted	5	2	2	2 / 0
<code>s1-human-control-force</code>	canary	hosted	6	3	2	1 / 1

Line	Severity	Tier floor	Scope	Carve-outs	Exemptions	ANY / ALL
s2-untargeted -profiling	canary	connected	6	3	3	2 / 1

The ranges are narrow by construction: scope sizes run from three to six tokens, every line carries two or three narrative carve-outs, and every line carries two or three typed exemptions. No line is an outlier that concentrates most of the registry’s narrowing surface, and no line is a bare prohibition with no stated exemption path. The three ALL-mode exemptions sit on **s1-human-control-for ce**, **s2-untargeted-profiling**, and **downstream-transfer** — one each — where the exemption’s trigger matches only when *all* of its tokens are declared rather than any single one.

3.16.4 Evidence depth and the free-pass check

The 16 typed exemptions distribute their 37 evidence requirements narrowly: eleven exemptions require exactly two evidence kinds and five require exactly three (`exemption_evidence_matrix`; the minimum across all rows is two, the maximum three). Trigger scopes are similarly small — nine exemptions trigger on two tokens, six on three, and one on six.

Trigger scope and evidence are separate gates, and the difference is worth seeing rather than reading: [Proposition 7](#) and [fig. 24](#) probe every exemption with one trigger token and then with all of them, and the three ALL-mode rows stay blocked until every token is declared.

The floor of two is the registry’s most important structural property, so it is checked rather than assumed. `unevidenced_exemptions` returns every matrix row whose exemption requires *no* evidence kind at all — an exemption that any matching declaration would satisfy, which is to say a free pass through its line. On the current registry the function returns an empty tuple. That empty result is meaningful only because the detector is proven able to fire: the test suite constructs a planted registry containing a deliberate zero-evidence exemption and asserts both that `unevidenced_exemptions` reports it and that the registry invariant suite fails on the same input. Absence of a finding from a detector that has demonstrated detection is evidence about the current registry version; absence of a finding alone would be no evidence at all.

[fig. 27](#) puts the two preceding sections on one plate: the per-line table as four bars on a shared scale, the severity and tier-floor splits as a footer strip, and the free-pass count as a band that is drawn whether or not it is zero. Rendering the zero matters. A panel that appeared only when a free pass existed would make today’s clean registry indistinguishable from a figure set that had quietly stopped checking; a band reading EXEMPTIONS REQUIRING NO EVIDENCE AT ALL: 0 is a result. The regression test for this figure injects one synthetic evidence-free exemption into a copy of the registry and asserts the band changes, so the zero is falsifiable in the plate as well as in the analysis module.

These numbers travel with the version. All of them describe the registry state pinned by the canonical digest in [sec. 3.15](#); a future amendment that adds a line, widens a scope, or relaxes an evidence requirement changes the derived numbers, the canary payload, and this section’s claims together. That coupling is intentional: composition claims that are recomputed from source cannot silently outlive the registry state they describe.

Where two boundaries share a word

Every canonical scope token in the registry against every line that declares it; the count column repeats the row in text.

SOURCE-DRIVEN SCHEMATIC · NOT AN EMPIRICAL MEASUREMENT

SCOPE TOKEN sorted	cogsec- integrity	downstream- transfer	dual- use- ablation	open- science- good- faith	provenance- and- consent	s1- human- control- force	s2- untargeted- profiling	LINES declaring it
autonomous_weapon	☐	☐	☐	☐	☐	☒	☐	1
benchmark_claim	☐	☐	☐	☒	☐	☐	☐	1
biometric_id	☐	☐	☐	☐	☐	☐	☒	1
bulk_data	☐	☐	☐	☐	☐	☐	☒	1
cogsec	☒	☐	☐	☐	☐	☐	☐	1
consent	☐	☐	☐	☐	☒	☐	☐	1
data_acquisition	☐	☐	☐	☐	☒	☐	☐	1
deception	☒	☐	☐	☐	☐	☐	☐	1
disinformation	☒	☐	☐	☐	☐	☐	☐	1
downstream	☐	☒	☐	☐	☐	☐	☐	1
dragnet	☐	☐	☐	☐	☐	☐	☒	1
force	☐	☐	☐	☐	☐	☒	☐	1
handoff	☐	☒	☒	☐	☐	☐	☐	2 SHARED
influence_ops	☒	☐	☐	☐	☐	☐	☐	1
integration	☐	☒	☐	☐	☐	☐	☐	1
kinetic	☐	☐	☐	☐	☐	☒	☐	1
lethality	☐	☐	☐	☐	☐	☒	☐	1
manipulation	☒	☐	☐	☐	☐	☐	☐	1
metric	☐	☐	☐	☒	☐	☐	☐	1
model_release	☐	☐	☒	☐	☐	☐	☐	1
pii	☐	☐	☐	☐	☒	☐	☐	1
profiling	☐	☐	☐	☐	☐	☐	☒	1
propaganda	☒	☐	☐	☐	☐	☐	☐	1
provenance	☐	☐	☐	☒	☒	☐	☐	2 SHARED
publication	☐	☐	☐	☒	☐	☐	☐	1
reproducibility	☐	☐	☐	☒	☐	☐	☐	1
resale	☐	☒	☐	☐	☐	☐	☐	1
scraping	☐	☐	☐	☐	☒	☐	☐	1
sublicense	☐	☒	☐	☐	☐	☐	☐	1
surveillance	☐	☐	☐	☐	☐	☐	☒	1
targeting	☐	☐	☐	☐	☐	☒	☐	1
tracking	☐	☐	☐	☐	☐	☐	☒	1
weapons	☐	☐	☐	☐	☐	☒	☐	1
weights	☐	☐	☒	☐	☐	☐	☐	1

34 distinct tokens · 32 declared by exactly one line · 2 shared

handoff → downstream-transfer + dual-use-ablation · executed verdict at hosted: NON COMPLIANT

provenance → open-science-good-faith + provenance-and-consent · executed verdict at hosted: NON COMPLIANT

A shared token implicates both of its lines at once, so one line's verified exemption cannot clear it — the other line is still evaluated on its own terms. This is the registry's declared vocabulary, not a map of what an action actually does: matching stays lexical over the declared scope and is not a semantic safety classifier.

Figure 26: Where two boundaries share a word. Presence grid computed by `red_line.analysis.registry_metrics.scope_token_membership`: every one of the 34 distinct canonical scope tokens against each of the seven lines, with a filled mark where the line declares the token and a count column repeating each row in text. Two tokens — **handoff** and **provenance** — are declared by two lines; the other 32 belong to one line each. The footer prints the verdict the real evaluator returned for each shared token during the tier-monotonicity sweep, so the consequence is executed rather than asserted. Implementation fact: this is the registry's declared vocabulary. Interpretation, stated as a limit: matching remains lexical over a declared scope and is not a semantic classifier, so the grid shows which words can implicate two boundaries, never whether an action truly does.

The shape of the boundary, line by line

Per-line structural counts read from the live registry; bars share one scale and every bar carries its number.

SOURCE-DRIVEN SCHEMATIC · NOT AN EMPIRICAL MEASUREMENT

RED LINE sorted by id	SEVERITY	TIER FLOOR	SCOPE tokens	CARVE-OUTS narrative	EXEMPTIONS typed	EVIDENCE kinds used
cogsec-integrity	ABSOLUTE	CONNECTED	6	2	2	4
downstream-transfer	STRONG	CONNECTED	5	2	2	4
dual-use-ablation	STRONG	AIR-GAPPED	3	3	3	4
open-science-good-faith	STRONG	AIR-GAPPED	5	2	2	3
provenance-and-consent	STRONG	HOSTED	5	2	2	3
s1-human-control-force	CANARY	HOSTED	6	3	2	3
s2-untargeted-profiling	CANARY	CONNECTED	6	3	3	4

severity split — CANARY 2 · ABSOLUTE 1 · STRONG 4

tier-floor split — HOSTED 2 · CONNECTED 3 · AIR-GAPPED 2

EXEMPTIONS REQUIRING NO EVIDENCE AT ALL: 0

every typed exemption demands at least one VERIFIED record

These are structural counts over the author's own registry. A longer bar means a wider declared surface, not a stronger commitment, a safer practice, or a basis for comparing one author's boundary with another's.

Figure 27: The shape of the boundary, line by line. Per-line structural profile computed by `red_line.analysis.registry_metrics.line_summaries`, `severity_distribution`, `tier_floor_distribution`, and `unevidenced_exemptions`. Each row is one of the seven lines in id order with its severity grade and oversight floor as text-labelled chips, and four counts on one shared bar scale: declared scope tokens, narrative carve-outs, typed exemptions, and distinct evidence kinds used. Every bar carries its number, so the plate reads without colour. The bottom band renders the count of typed exemptions requiring no evidence at all, currently zero. Implementation fact: these are counts over registry fields. Interpretation, stated as a limit: a longer bar is a wider declared surface, not a stronger commitment, a safer practice, or a basis for comparing this author's boundary with anyone else's.

3.17 Limitations and negative space

3.17.1 What this document does not decide

Red Line is deliberately narrow. It does not decide:

- **Legal adjudication:** whether an act is lawful, licensed, contractually authorized, or subject to a particular jurisdiction's standard.
- **Truth verification:** whether a cited source, artifact, witness, or evidence record is true, complete, current, or free of strategic misrepresentation.
- **Real-world safety:** whether a system will behave safely, whether harm will occur, or whether a documented control will work under operational pressure.
- **Institutional authorization:** whether an employer, customer, regulator, community, or affected person has granted authority to proceed.
- **Independent external witnessing:** whether anyone outside the author's write boundary has received, checked, or attested to the registry or finding.

The instrument can require escalation when these questions cannot be established, but it cannot answer them by itself.

Auditability, not enforcement. The package records a local result. It does not stop execution, remove access, adjudicate legality, or create an independent review body. A named authorization is visible but never releases a block.

Evidence is not truth. VERIFIED means a reviewable artifact or source was identified. The package does not establish that the artifact is accurate, complete, current, lawfully obtained, or free of strategic misrepresentation. Contradicted or unsupported context blocks the result, but a plausible false record can still pass the local status gate.

Lexical, not semantic. Explicit scope aliases are safer than heuristic stemming, but the evaluator still matches declared tokens. A complete-looking action can be mislabeled. Description mismatch hints expose one symptom; they do not solve semantic interpretation.

False positives and false negatives. The fail-closed intake is intentionally biased toward stopping when required information is absent or unresolved. That can produce a false positive for caution: a legitimate exemption may wait until its evidence is assembled. The lexical boundary also permits a more serious residual false negative: a dangerous action can be described with a benign token set or a complete-looking but misleading evidence packet. The instrument reduces these risks through explicit scope, negative tests, source provenance, and independent review; it cannot remove them without semantic and institutional authority it does not have.

3.17.2 Adversarial declarations

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

Scope-token laundering. A targeting-component action declares only `logistics`, `translation`, and `maintenance` — permissible support tokens that match the `adjacent-force-support` exemption on `s1-human-control-force` — while omitting `targeting`, `weapons`, or `autonomous_weapon`. The evaluator matches declared tokens against scope and exemption triggers; the action's narrowed scope implicates only `s1-human-control-force`, and the verified exemption returns `COMPLIANT`. The evaluator cannot inspect whether the declared tokens honestly describe a component that selects and engages targets.

Evidence-status fabrication. An action provides VERIFIED records with plausible but fabricated references — a purpose statement citing a non-existent contract, a human-control record pointing to a private repository that has never been reviewed. The evidence gate requires VERIFIED status for every dimension the exemption demands; a record whose status field reads VERIFIED satisfies the gate regardless of whether the referenced artifact exists, is accurate, or was genuinely reviewed. The evaluator records epistemic status, not ground truth.

Refresh-date laundering. The freshness window requires evidence dated within 180 days of the review date. An action whose evidence records were last genuinely verified 300 days ago rewrites their dates to the current review date and returns `COMPLIANT`. The evaluator checks the declared date against the window; it cannot distinguish a genuine re-verification — a reviewer re-examining the source — from an edit to a date string.

Declared-unknown gamesmanship. An action declares every sensitive dimension as `not_applicable` and supplies a single VERIFIED record backing that designation, aiming to minimize the evidence surface while staying within the gate. The evaluator still demands VERIFIED evidence for each dimension, and `not_applicable` is a permitted value only when a verified record supports it — but the evaluator cannot assess whether the dimension is genuinely inapplicable or merely declared so to shrink the review burden. An action whose sole remaining required dimension carries a fabricated record passes on the strength of a single false positive.

These are instances of a well-documented dynamic, not defects unique to this design. A refusal instrument that certified safety would make these failure modes catastrophic, because a gamed **COMPLIANT** would launder a dangerous action into apparent permission. Red Line’s design response is to refuse the certifying role entirely: a classification reports what the declarer provided and the evaluator computed, so a gamed **COMPLIANT** overstates nothing but the presence of declared evidence. The attacks also stay inspectable rather than hidden — the declared scope, the evidence status records, and the declared dates are the very record a reviewer reads, so a reviewer who asks “do these tokens describe this work?”, “was this evidence genuinely reviewed?”, or “what changed at this refresh?” is asking questions the declaration itself exposes. The instrument narrows what gaming can counterfeit; it cannot remove the need for the human judgment those questions require, and it never converts any classification into a safety certification, an accreditation, or a permission.

Registry scope is non-exhaustive. **OUTSIDE_SCOPE** is not “safe” and **COMPLIANT** is not “universally acceptable.” Both are bounded by seven personal lines, their current vocabulary, and the quality of the intake.

Personal authority remains personal. The lines are the author’s refusals, not a claim that other people must share them. The global and historical reading base widens the questions and exposes blind spots; it does not turn the author into a representative of the traditions cited.

Scholarship is curated, not representative. The reading base is not a systematic review or a substitute for affected-party testimony, local expertise, or jurisdiction-specific legal analysis. Its purpose is to widen the questions asked of a personal instrument and to make transfer limits visible.

A formal statement is not a stronger claim. The definitions and propositions in sec. 3.14 say what the program returns for declared inputs. Writing that down precisely, and binding each proposition to a test, removes one failure mode — prose drifting away from the procedure — and adds no authority. A proposition can be true of the code and useless in the world at the same time, which is the case **Proposition 3** describes, and the same is true of **Proposition 8** and **Proposition 4**: the gate is strict, and strictness is not accuracy.

Structural analytics describe shape, not strength. The derived numbers in sec. 3.16 and the exercised outcome coverage in sec. 3.13 are registry introspection, not measurement of the world. An exemption that demands three evidence kinds is not thereby “stronger” than one that demands two, and the demand profile across intake dimensions is not a risk model; comparing such counts across authors as if they scored rigor would recreate exactly the safety-score misuse this document disclaims. The free-pass detector reports on structurally degenerate exemptions only — a substantively wrong boundary with well-formed structure passes every metric. Its empty result is bounded to the current registry state, and the outcome-coverage report is a control-flow reachability pin built on fixture evidence: a **complete** report shows the evaluator can produce all five classifications, not that any real engagement was classified correctly.

Canary dependence. A same-author repository fixture is a regression anchor, not an external witness. The hash detects drift only when a prior copy is held outside the writer’s control and checked by someone else. The canary is a tamper-evidence pattern, not legal protection or prevention.

Publication pipeline. Deterministic figures, output validation, and rendered inspection reduce source-to-artifact risk, but they do not prove that a polished figure is rhetorically fair. PDF and HTML review remain required.

Release state. This work is released as a standalone repository at https://github.com/docxology/red_line. It has no external canary witness, no minted DOI, no independent review body, and no execution admission control. A public release should not claim those things until the release evidence bundle in **docs/claim-register.md** exists and an independent reader has checked the source, outputs, and prior statement.

3.18 Conclusion

Red Line is deliberately a refusal instrument. It makes seven personal No's legible before a request becomes a project, then requires an evidence-bearing context before a proposed near-boundary action can receive a policy result. That strictness changes the meaning of the tool's green states: **COMPLIANT** means that this documented intake satisfied this registry, while **OUTSIDE_SCOPE** means only that no current line applied. Neither is a general safety certificate.

The four propositions stated in [the introduction](#) hold.

In use, the sequence is equally simple: read the beacon; declare the action, scope, and tier; assemble the nine evidence dimensions; run the local result; retain the finding; and maintain a prior canary outside the author's write boundary. The order is a safeguard against sunk-cost reasoning, scope laundering, and self-certification. It is not a substitute for legal review, affected-party participation, technical safety work, or institutional authority.

What the instrument actually does is now written down twice, in prose and in [sec. 3.14](#), with a test standing behind every proposition. That duplication is the point. A document about a decision procedure is worth reading only if a reader can find out where it has stopped being true, and the second statement is where they can.

The mechanism source is [sec. 3.3](#); the personal adaptation stands on its own. Its registry, evidence model, typed exemptions, review findings, non-bypassable authorizations, canary, scholarship ledger, figures, and tests are this project's claims. The reading base does not dissolve the first-person nature of the commitments; it makes the author more accountable for the blind spots around them, and the export-control and refusal literatures locate what the instrument is not — neither a control regime with force nor a refusal with standing. The claim register names the evidence and stopping point for each kind of assertion.

The companion works keep the line set non-redundant. Black Line asks how strong work is done; Golden Line asks what is worth reaching toward; White Line asks what is absent, unknowable, withheld, or ethically left unsaid. Red Line comes first because a positive method or aspiration is not a substitute for a clear boundary. The artifact is ready for continued private use and revision; it is not yet externally attested publication.

The work is built for the author's public research index ([docxology/docxology](#)) — machine-readable, cross-linked, and verification-logged — with an eventual public home at the [docxology/red_line](#) repository. That is the boundary kept in plain sight: the refusal is designed to be read before a request becomes a project.

4 Black Line: Strong Work in Public

A Positive Operating Discipline for Concise, Rigorous Research and Engineering

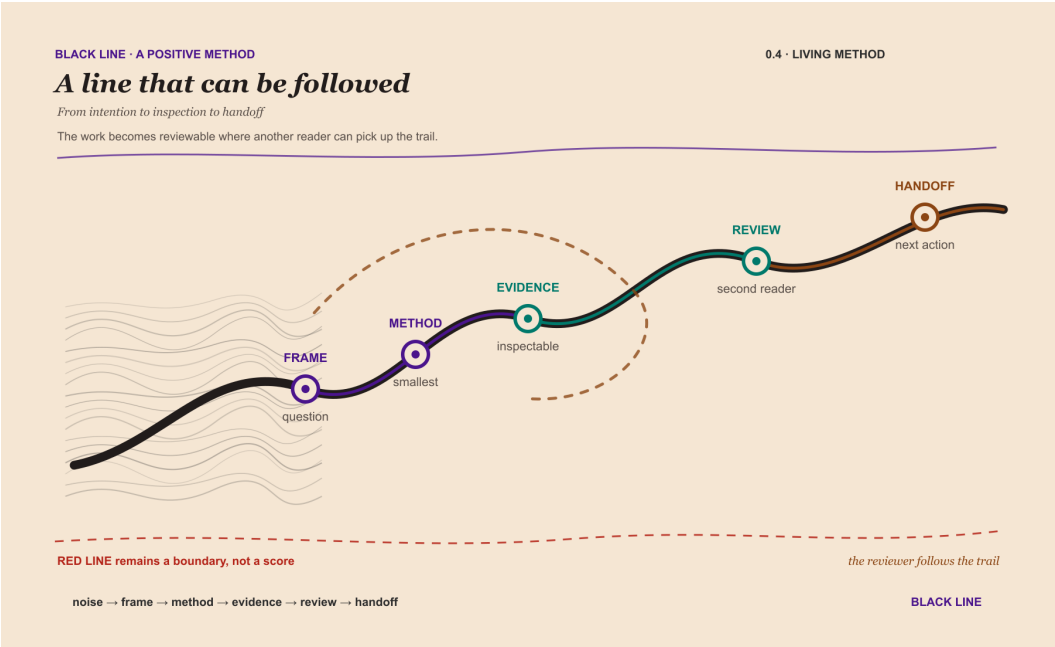


Figure 28: Cover art for Black Line: Strong Work in Public

Reproduced unchanged in substance from its own source at version 0.4.0. It answers one question: How do I do strong work?

4.1 Abstract

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

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

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

4.2 Introduction

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

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

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

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

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

The rest of this paper is organised as follows. [Section sec. 4.5](#) defines the method and its evidence wires. [Section sec. 4.8](#) states the evaluator formally as definitions and propositions. The worked examples in [Section sec. 4.9](#) demonstrate the instrument over real declarations, and [Section sec. 4.10](#) names the epistemic boundaries that the instrument cannot cross.

4.3 Relationship to the line set

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

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

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

4.3.1 Note on the name

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

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

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

BLACK LINE · A POSITIVE METHOD

0.4 · LIVING METHOD

A line that can be followed

From intention to inspection to handoff

The work becomes reviewable where another reader can pick up the trail.

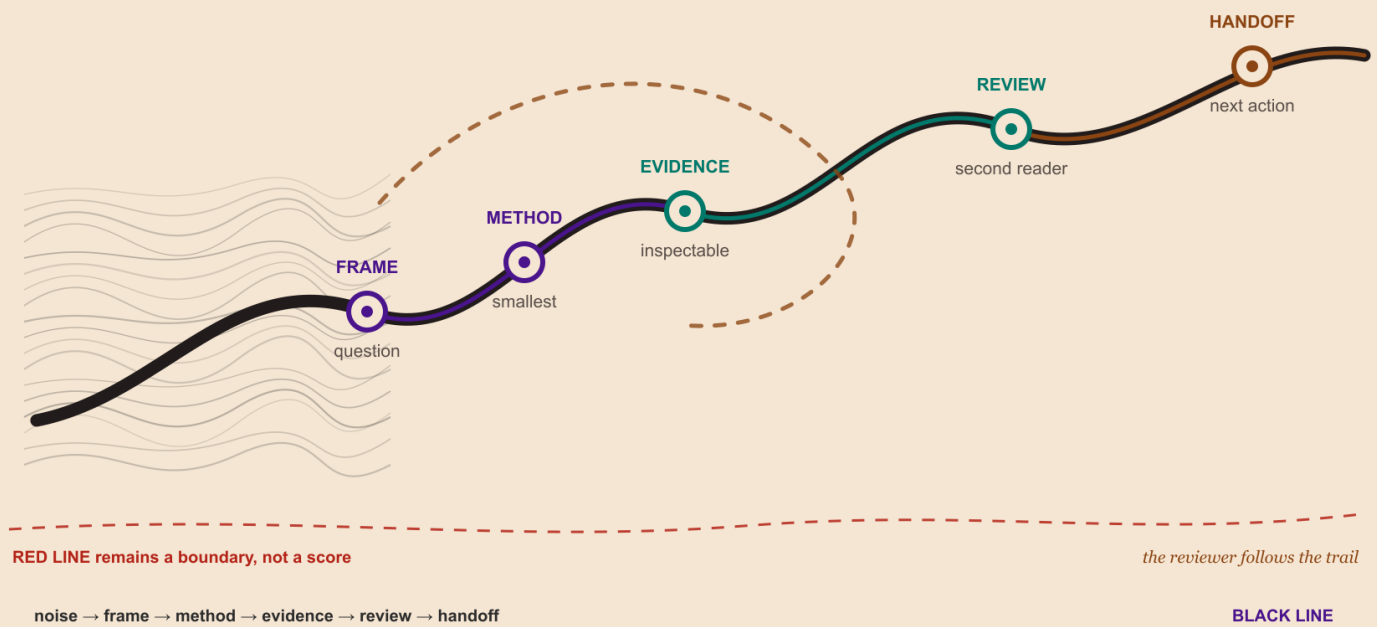


Figure 29: Cover art showing unresolved intention narrowing into a followable review line through method, evidence, review, and handoff.

4.4 Intellectual lineage: where the practices come from

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

4.4.1 Framing and falsification

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

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

4.4.2 Traceability, review, and the norms of science

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

4.4.3 Traceable prose and the smallest sufficient method

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

4.4.4 Reproducibility as the load-bearing modern practice

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

4.4.5 Reproducibility is not replication

The vocabulary needs one further distinction, and the distinction is the whole scope of the **reproducible-from-clean** wire. Goodman, Fanelli, and Ioannidis separate methods reproducibility, results reproducibility, and inferential reproducibility; the word *reproducible* should not quietly become a synonym for *true* [Goodman et al., 2016]. The National Academies report likewise distinguishes computational reproducibility — consistent computation from the same inputs, code, methods, and conditions — from replicability in a new study addressing the same question [National Academies of Sciences, Engineering, and Medicine, 2019].

Black Line’s clean rerun is deliberately the first, narrower claim: whether another reader can re-execute the recorded procedure, not whether the result generalizes or the inference holds.

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

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

4.4.6 Openness is infrastructure; review is a social act

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

4.4.7 Craft, *technē*, and legibility as a designed partial view

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

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

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

4.4.8 Verification is not validation

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

4.4.9 Handoffs are situated coordination objects

The concise handoff also has a social-scientific lineage. Suchman’s analysis of plans and situated action warns that an abstract plan does not determine what people will do in a changing setting [Suchman, 1987]. Star and Griesemer’s account of boundary

objects shows how a shared artifact can support cooperation across social worlds while remaining locally interpretable [Star and Griesemer, 1989]. Black Line’s handoff is deliberately modest in this sense: it preserves the decision, evidence trail, current limit, and next action so another reader can orient themselves, but it does not pretend to transfer the author’s tacit skill or eliminate the need for situated judgment. A handoff is a coordination surface, not a complete substitute for collaboration.

4.4.10 Humility is an operational requirement

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

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

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

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

4.4.11 What the lineage does and does not license

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

4.5 Method: positive wires and observable evidence

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

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

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

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

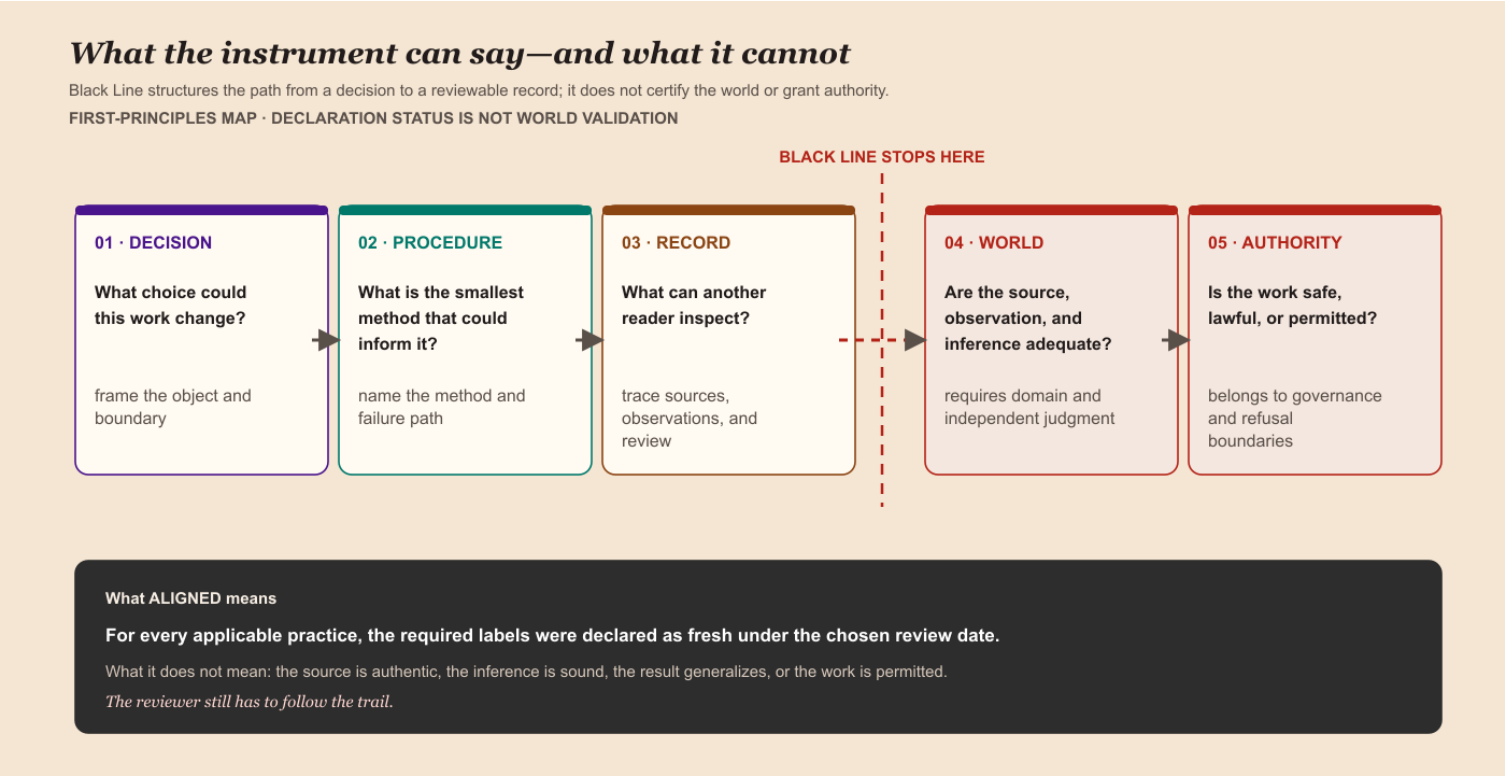


Figure 30: The six-practice-family boundary map separates the decision, procedure, and record that Black Line can structure from world adequacy and authority that require domain, independent, or governance review. **ALIGNED** means fresh declaration coverage for applicable practices, not truth or permission.

4.5.2 Three claim classes, three evidentiary burdens

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

Claim class	Example	What supports it	What it cannot become
Implementation	“The evaluator returns NEEDS_REWORK for a blank description.”	Source inspection and a test that exercises the branch.	A claim that the work itself needs substantive rework beyond the declared contract.

Claim class	Example	What supports it	What it cannot become
Methodological	“A concise handoff is worth requiring.”	Design rationale, scholarly lineage, and a stated limitation.	A causal finding that the practice improves outcomes.
World or authority	“The source is authentic” or “the work is permitted.”	Domain verification, independent evidence, or governance review.	A conclusion licensed by ALIGNED .

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

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

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

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

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

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

4.6 Operating protocol: the smallest honest loop

The protocol operationalizes the six practice families of the method.

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

The executable protocol has five review actions:

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

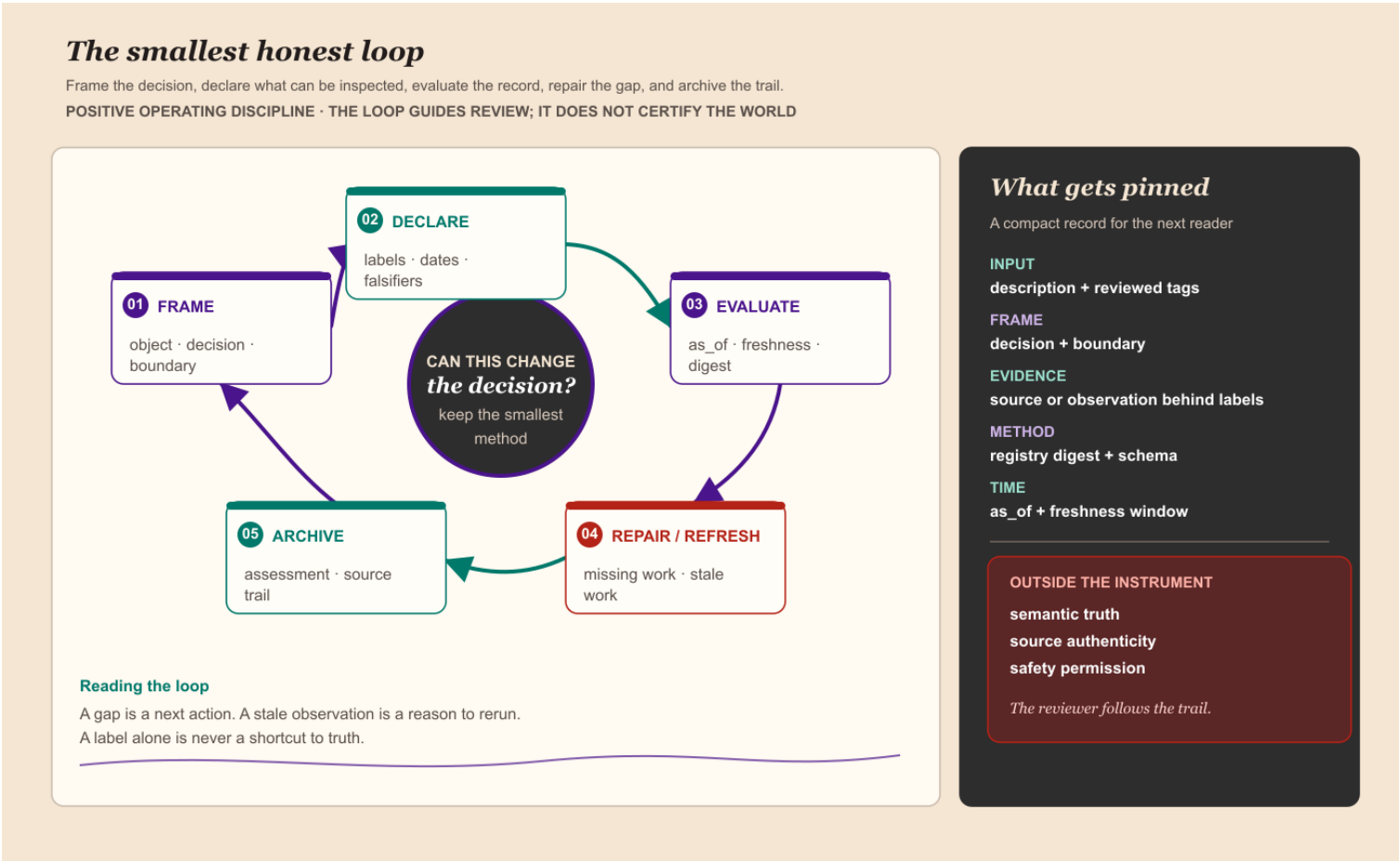


Figure 31: The smallest honest operating loop: frame the decision, declare pointers to inspectable artifacts, evaluate with a pinned date and registry digest, repair or refresh gaps, and archive the trail. The dark panel makes explicit what remains outside the instrument’s authority.

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

4.7 The Black Line practices

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

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

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

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

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

4.7.1 Registry coverage and burden

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

Tag	Practices reached	Required labels	Practices
research	8 of 11	16	question-first, source-traceable, failure-visible, concise-handoff, reproducible-from-clean, stated-uncertainty, negative-results-kept, review-before-reliance
analysis	7 of 11	14	question-first, source-traceable, smallest-sufficient-method, failure-visible, stated-uncertainty, negative-results-kept, data-provenance

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

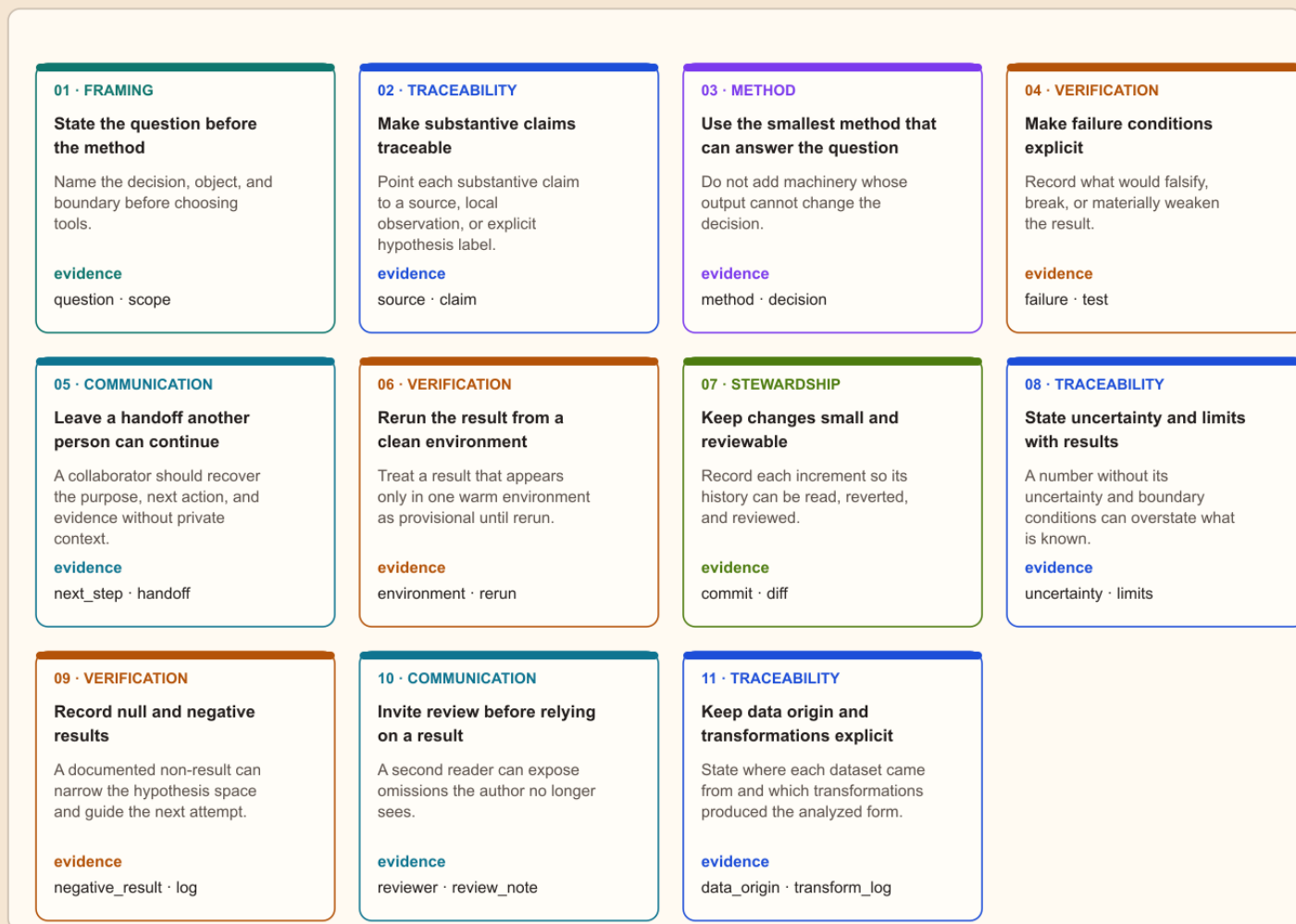
The declaration burden therefore varies eightfold with tag choice — a **data**-only attempt is scored against a single two-label practice, while a **research** attempt must declare sixteen labels to reach **ALIGNED**. Every practice carries exactly two required labels, so burden scales linearly with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

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

Small wires make a review surface visible

Each practice names a review surface; a missing label requests work, not a judgment about intent.

SOURCE-DRIVEN SCHEMATIC · REGISTRY DECLARATION ORDER · NOT A QUALITY OR TRUTH SCORE



Craft families ■ FRAMING ■ TRACEABILITY ■ METHOD ■ VERIFICATION ■ COMMUNICATION ■ STEWARDSHIP

Boundary Black Line describes how to work well; it never grants permission to cross Red Line.

Figure 32: The 11 practices drawn as cards in registry declaration order, color-keyed to the 6 practice families. The order interleaves families and ends at **data-provenance**: it is a declaration order, not a workflow. Each card names a review surface and its required labels; the figure is not a quality or truth score and does not grant permission to cross Red Line.

The registry as a taxonomy of practice families

11 practices across 6 families; each practice is reached by tags from a reviewed 5-tag vocabulary.

SOURCE-DRIVEN SCHEMATIC · FAMILY COVERAGE AND TAG MEMBERSHIP ARE ENFORCED BY THE INVARIANTS

Tag vocabulary: analysis data engineering research writing

FRAMING

1 practice

State the question before the method

tags: analysis · research · writing

evidence: question · scope

METHOD

1 practice

Use the smallest method that can answer the question

tags: analysis · engineering

evidence: method · decision

COMMUNICATION

2 practices

Leave a handoff another person can continue

tags: engineering · research · writing

evidence: next_step · handoff

Invite review before relying on a result

tags: engineering · research · writing

evidence: reviewer · review_note

STEWARDSHIP

1 practice

Keep changes small and reviewable

tags: engineering · writing

evidence: commit · diff

TRACEABILITY

3 practices

Make substantive claims traceable

tags: analysis · research · writing

evidence: source · claim

State uncertainty and limits with results

tags: analysis · research

evidence: uncertainty · limits

Keep data origin and transformations explicit

tags: analysis · data

evidence: data_origin · transform_log

VERIFICATION

3 practices

Make failure conditions explicit

tags: analysis · engineering · research

evidence: failure · test

Rerun the result from a clean environment

tags: engineering · research

evidence: environment · rerun

Record null and negative results

tags: analysis · research

evidence: negative_result · log

Figure 33: The 11 practices grouped by 6 practice families, with the reviewed 5-tag vocabulary that makes each practice applicable to a work attempt. Family coverage and tag membership are enforced by structural invariants; the taxonomy maps reachability, not evidence quality.

The registry's evidence-label contract

Each practice names labels a reviewer can look for; labels are not the underlying artifacts or their verification.

DERIVED FROM REGISTRY · 11 PRACTICES · 22 REQUIRED LABELS

practice	reviewed tags	required evidence labels
01 State the question before the method FRAMING	analysis research writing	question scope
02 Make substantive claims traceable TRACEABILITY	analysis research writing	source claim
03 Use the smallest method that can answer the question METHOD	analysis engineering	method decision
04 Make failure conditions explicit VERIFICATION	analysis engineering research	failure test
05 Leave a handoff another person can continue COMMUNICATION	engineering research writing	next_step handoff
06 Rerun the result from a clean environment VERIFICATION	engineering research	environment rerun
07 Keep changes small and reviewable STEWARDSHIP	engineering writing	commit diff
08 State uncertainty and limits with results TRACEABILITY	analysis research	uncertainty limits
09 Record null and negative results VERIFICATION	analysis research	negative_result log
10 Invite review before relying on a result COMMUNICATION	engineering research writing	reviewer review_note
11 Keep data origin and transformations explicit TRACEABILITY	analysis data	data_origin transform_log

Boundary The evaluator can confirm that labels were declared and matched;
it cannot confirm that a source is real, a test passed, or a claim is true.

Figure 34: The registry's evidence-label contract: each practice names labels a reviewer can look for, while the instrument explicitly does not treat a declaration as independent verification of a source, test, or claim.

Tag choice sets the declaration burden

Filled cells mark which practices a tag reaches; the margins total each tag's practices and required labels.

DERIVED FROM REGISTRY · 27 OF 55 CELLS APPLICABLE · APPLICABILITY MAP, NOT A QUALITY OR TRUTH SCORE

	question-first	source-traceable	smallest-sufficient-method	failure-visible	concise-handoff	reproducible-from-clean	versioned-increments	stated-uncertainty	negative-results-kept	review-before-reliance	data-provenance	tag burden
analysis	2	2	2	2				2	2		2	7 practices 14 labels
data											2	1 practice 2 labels
engineering			2	2	2	2	2			2		6 practices 12 labels
research	2	2		2	2	2		2	2	2		8 practices 16 labels
writing	2	2			2		2			2		5 practices 10 labels

Asymmetry 'research' commits a declarer to 8 practices (16 labels); 'data' to 1 (2). A narrow tag set buys a cheaper **ALIGNED** — a coverage fact reviewers should read alongside any status.

Boundary Cell numbers are required-label counts. Applicability is not evidence, quality, safety, or permission.

Figure 35: The tag-practice coverage matrix derived from the registry: filled cells mark applicability, cell numbers give each practice's required-label count, and the margin totals each tag's reach and declaration burden. The burden is asymmetric — **research** reaches 8 practices (16 required labels) while **data** reaches 1 (2 labels) — so a narrow tag set buys a cheaper **ALIGNED**. The matrix maps applicability, not evidence quality, safety, or permission.

4.8 Formal method: the evaluator and its invariants

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

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

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

4.8.1 Domain objects

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

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

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

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

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

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

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

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

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

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

4.8.2 Status codomains

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

$$S_p = \{\text{ALIGNED}, \text{NEEDS_EVIDENCE}, \text{NEEDS_REWORK}\}.$$

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

$$S_a = S_p \cup \{\text{OUTSIDE_SCOPE}\} = \{\text{ALIGNED}, \text{NEEDS_EVIDENCE}, \text{NEEDS_REWORK}, \text{OUTSIDE_SCOPE}\}.$$

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

4.8.3 The staged evaluator

The function under study is

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

4.8.4 Surfaces, projection, and the report envelope

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

Definition 15 (Evidence surfaces). For an applicable practice p evaluated against (F, Σ) , the *evidence surfaces* are the frozen record $\sigma(p) = (\text{practice_id}, \text{present}, \text{missing}, \text{stale})$ with exactly those four fields: present and missing partition $p.\text{req}$ by membership in F , and stale is the subsequence of missing whose labels are in Σ , so $\text{stale} \subseteq \text{missing}$ always holds. All three tuples keep the

How evaluate_work maps declarations to a review status

A staged pipeline: normalize intake, match practices by tag, score each, then take the most demanding status.

DERIVED FROM EVALUATOR RULES · 3 PRACTICE STATUSES · 4 ASSESSMENT STATUSES

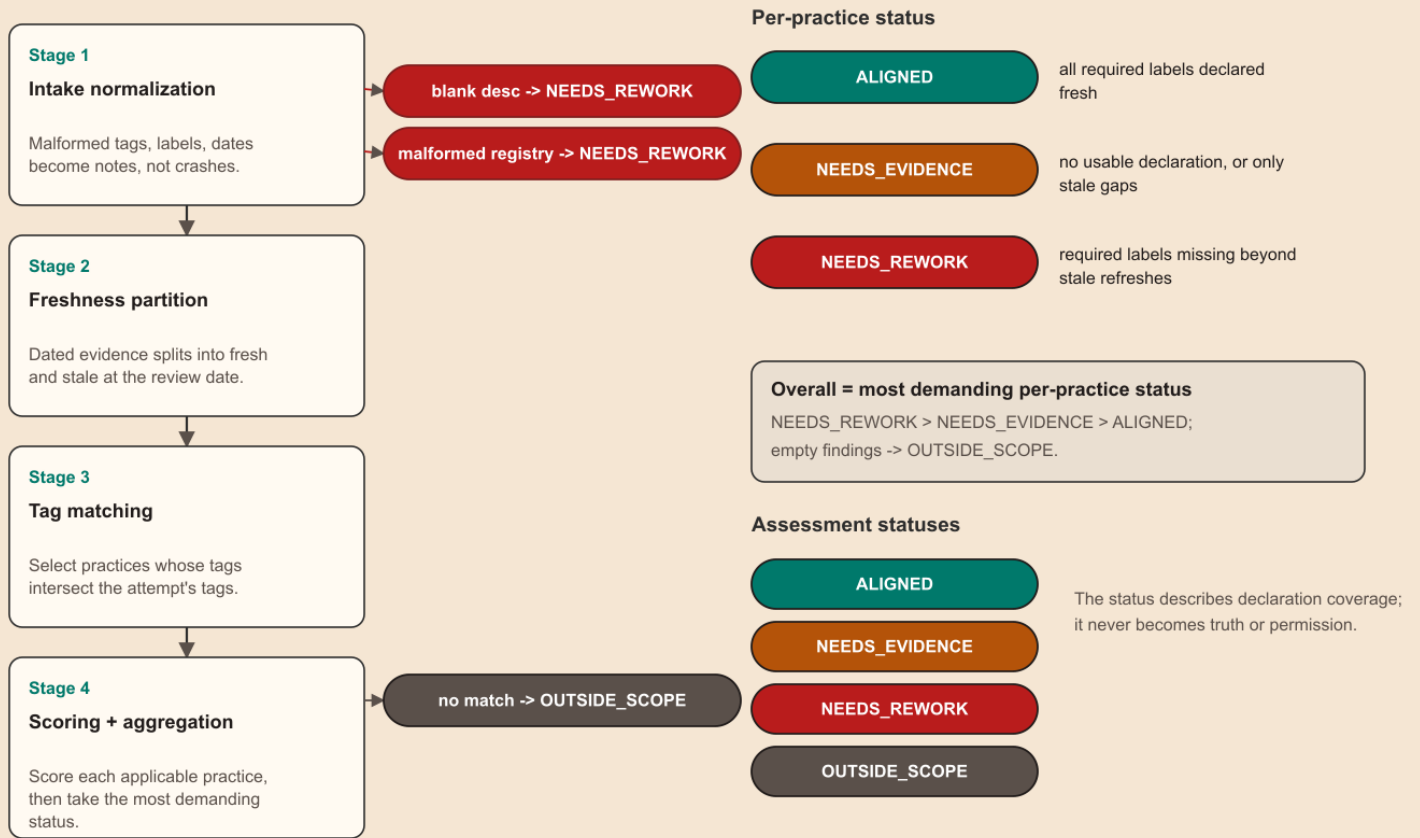


Figure 36: The staged evaluation path. Intake normalization can block on a blank description, and a malformed practice registry fails closed as `NEEDS_REWORK` before any scoring; otherwise practices are matched by tag intersection, each applicable practice is scored `ALIGNED`, `NEEDS_EVIDENCE`, or `NEEDS_REWORK`, and the overall status takes the most demanding per-practice status, or `OUTSIDE_SCOPE` when no practice applies. The diagram is derived from the evaluator rules and status enums; a status reports declaration coverage and freshness, not verified evidence or authorization.

practice’s declared evidence order. The surfaces are co-present state — support and resistance on one practice are both kept, and neither erases the other — and, like every object in this section, they describe declaration coverage only.

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

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

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

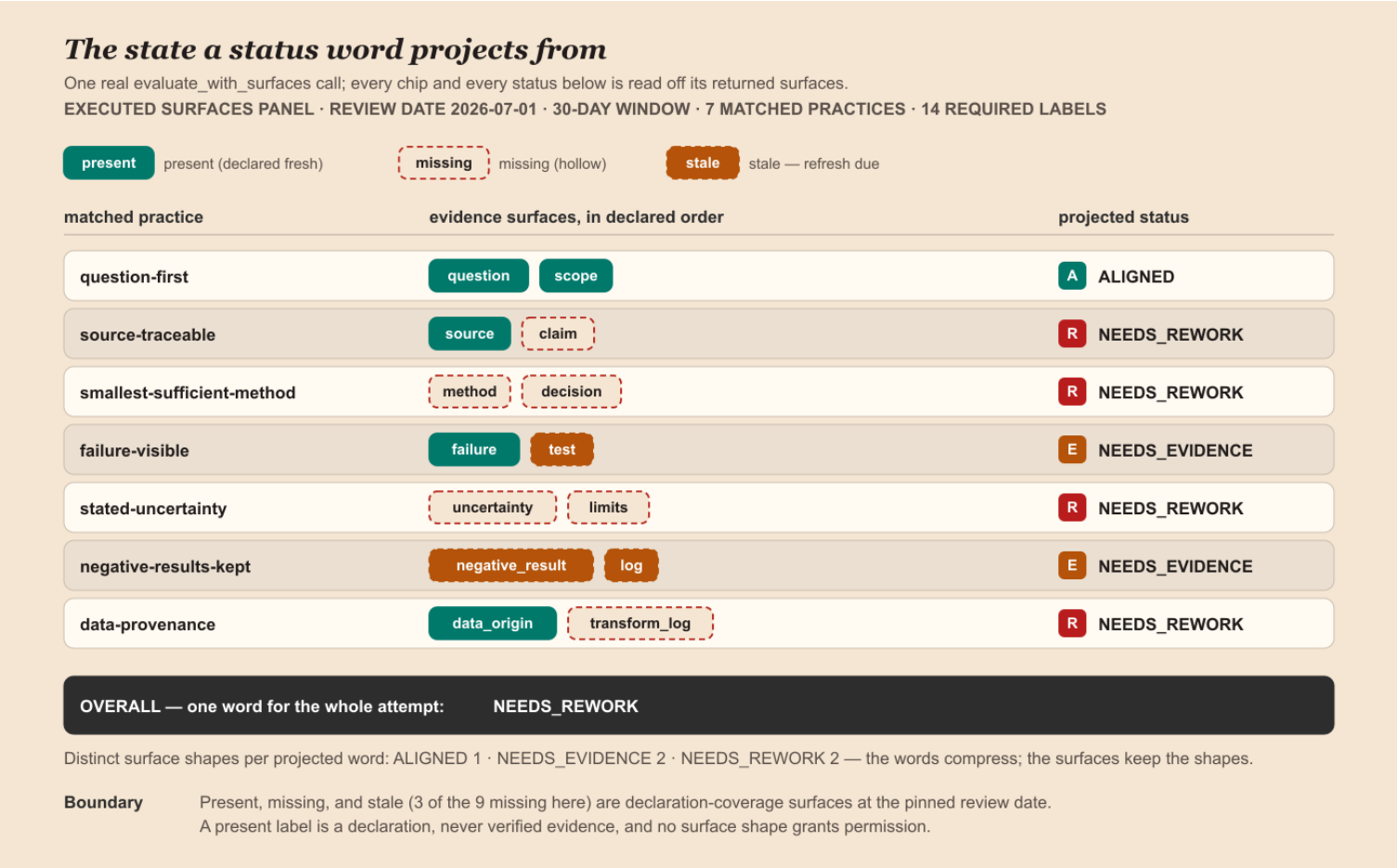


Figure 37: One declaration’s typed evidence surfaces beside the statuses projected from them: a single `evaluate_with_surfaces` call at review date 2026-07-01 under a 30-day window matches 7 practices, whose 14 required labels split into 5 present (filled), 9 missing (hollow), and 3 of the missing merely stale (amber refresh marks). One status word covers distinguishable surface shapes: `NEEDS_EVIDENCE` is projected from 2 distinct present/missing/stale shapes and `NEEDS_REWORK` is projected from 2 distinct present/missing/stale shapes, while the single overall word for the whole attempt is `NEEDS_REWORK`. The surfaces keep what the projection compresses — strong support and strong resistance co-present on one practice are not the same as no evidence — and, like the statuses, they describe declaration coverage only: a present label is a declaration, never verified evidence, and no surface shape grants permission.

A reader holding reports from several independent instruments needs one uniform way to say “this instrument, about this subject, at this review moment, said this — and here is the pointer to its complete native report.” The envelope is that data contract and nothing more.

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

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

4.8.5 Propositions

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

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

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

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

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

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

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

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

A declaration the evaluator cannot read still returns

Each row is one real evaluate_work call on a deliberately malformed WorkAttempt; the notes are the ones it returned.

EXECUTED INTAKE BATTERY · REVIEW DATE 2026-07-01 · 9 DECLARATIONS · 10 NOTES · 0 EXCEPTIONS

2 of the 9 block scoring outright and 1 reaches no practice once its tag declaration is dropped; the other 6 are scored normally.

malformed declaration	status	intake notes returned
description is blank 0 practice findings	R NEEDS_REWORK	· work description is empty or not text; restate the work before assessment
description is not text 0 practice findings	R NEEDS_REWORK	· work description is empty or not text; restate the work before assessment
tags declared as one string 0 practice findings	O OUTSIDE_SCOPE	· tag declaration is not a collection of labels and was ignored
evidence declared as a number 1 practice finding	E NEEDS_EVIDENCE	· evidence declaration is not a collection of labels and was ignored
evidence tokens blank and non-text 1 practice finding	A ALIGNED	· ignored a malformed evidence label · ignored a malformed evidence label
dated record carries no label 1 practice finding	A ALIGNED	· ignored a dated evidence record without a usable label
evidence date is unreadable 1 practice finding	R NEEDS_REWORK	· evidence 'data_origin' has an unreadable date and was not counted
evidence date is in the future 1 practice finding	R NEEDS_REWORK	· evidence 'data_origin' is dated in the future and was not counted
dated evidence is one string 1 practice finding	A ALIGNED	· dated evidence declaration is not a collection of records

Boundary Surviving malformed input is a robustness property of the intake stage, not tolerance of a bad declaration.
A note asks the declarer to fix something; it does not verify anything that was declared correctly.

Figure 38: Stage 1 executed over 9 deliberately malformed declarations at review date 2026-07-01: each is one real evaluate_work call, together they return 10 intake notes, and none raises. 2 block scoring outright (description is blank and description is not text) and 1 reaches no practice once its dropped tag declaration leaves nothing to match, while the other 6 are scored normally; across the battery the outcomes are ALIGNED, NEEDS_EVIDENCE, NEEDS_REWORK, OUTSIDE_SCOPE. Surviving malformed input is a robustness property of the intake stage, not tolerance of a bad declaration, and a note asks the declarer to fix something rather than verifying anything declared correctly.

`k_without_present_reason`. The same sweep is drawn as **the monotonicity lattice** in the worked examples, where the per-row monotonicity mark and the inversion count are read off executed paths rather than asserted.

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

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

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

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

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

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

4.8.6 Structural invariants

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

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

- **I1 — ids distinct.** Every practice id is a distinct, non-blank string. Planted: a duplicated, blank, and non-string id.
- **I2 — fields populated.** Title and wire are non-blank, and required evidence is a non-empty tuple of non-blank strings. Planted: a blank title, an empty evidence tuple, a blank label, and a string evidence field.

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

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

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

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

4.8.7 Claim-to-test binding

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

Definition	What the code must still do	Verifying test
Definition 1	Field names, order, and the <code>METHOD</code> default	<code>tests/test_formalism_definitions.py::test_practice_record_matches_the_definition</code>
Definition 2	V is exactly the five reviewed tags	<code>tests/test_formalism_definitions.py::test_tag_vocabulary_matches_the_definition</code>
Definition 3	R is an ordered tuple of eleven practices with a digest	<code>tests/test_formalism_definitions.py::test_registry_matches_the_definition</code>
Definition 4	K is exactly the six declared families	<code>tests/test_formalism_definitions.py::test_craft_families_match_the_definition</code>
Definition 5	W and the dated-item pair keep their shape	<code>tests/test_formalism_definitions.py::test_work_attempt_matches_the_definition</code>
Definition 6	A and its finding triples keep their shape	<code>tests/test_formalism_definitions.py::test_assessment_record_matches_the_definition</code>
Definition 7	S_p has exactly three members	<code>tests/test_formalism_definitions.py::test_practice_status_codomain_matches_the_definition</code>
Definition 8	$S_a = S_p \cup \{\text{OUTSIDE_SCOPE}\}$	<code>tests/test_formalism_definitions.py::test_assessment_status_codomain_matches_the_definition</code>
Definition 9	Each named bad configuration raises rather than coerces	<code>tests/test_formalism_definitions.py::test_review_configuration_refuses_what_the_definition_names</code>
Definition 10	<code>clean</code> strips, lowercases, and notes every drop	<code>tests/test_formalism_definitions.py::test_intake_normalization_matches_the_definition</code>

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

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

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

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

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

Every check, shown failing on a registry built to break it

Each row runs the whole battery once. The top row is the shipped registry; the rest each carry one planted defect.

EXECUTED BATTERY · 7 CHECKS × 8 REGISTRIES · 56 OUTCOMES · READ OFF RegistryCheck RECORDS

A boxed cell is the check the row was planted to break. A green check that never saw a bad registry proves nothing.

	practice_ids_distinct	practice_fields_populated	practice_tags_reachable	practice_kind_valid	kind_coverage	evidence_labels_matchable	registry_digest_computable	checks failed
the shipped registry nothing planted	P	P	P	P	P	P	P	0 of 7
practice_ids_distinct first practice appended a second time	F	P	P	P	P	P	P	1 of 7
practice_fields_populated title blanked	P	F	P	P	P	P	P	1 of 7
practice_tags_reachable tag set emptied	P	P	F	P	P	P	P	1 of 7
practice_kind_valid kind replaced by a string	P	P	P	F	F	P	F	3 of 7
kind_coverage every STEWARDSHIP practice removed	P	P	P	P	F	P	P	1 of 7
evidence_labels_matchable required label given a capital letter	P	P	P	P	P	F	P	1 of 7
registry_digest_computable tag field set to None	P	P	F	P	P	P	F	2 of 7

P check passed

F check failed

Collateral

2 of the 7 plants also fail a check they were not aimed at:

- the practice_kind_valid plant also fails kind_coverage and registry_digest_computable
- the registry_digest_computable plant also fails practice_tags_reachable

Boundary

A firing check shows the battery can reject a malformed registry. It says nothing about whether the practices are the right ones, whether their evidence is adequate, or whether any work was done well.

Figure 39: The 7-check structural battery run over 8 registries: the shipped registry, which passes every check, and 7 registries each carrying one planted defect. Every plant fails the check it targets, boxed in its row, which is what makes the battery a proof of detection rather than a record of greenness. 2 plants also fail a check they were not aimed at — the practice_kind_valid plant also fails kind_coverage and registry_digest_computable; the registry_digest_computable plant also fails practice_tags_reachable — because a single planted value can break more than one structural property at once; those cells are drawn rather than designed away. A firing check shows the battery can reject a malformed registry, not that the practices are the right ones, that their evidence is adequate, or that any work was done well.

4.9 Executed examples and boundaries

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

4.9.1 An executed incremental-declaration path

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

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

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

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

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

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

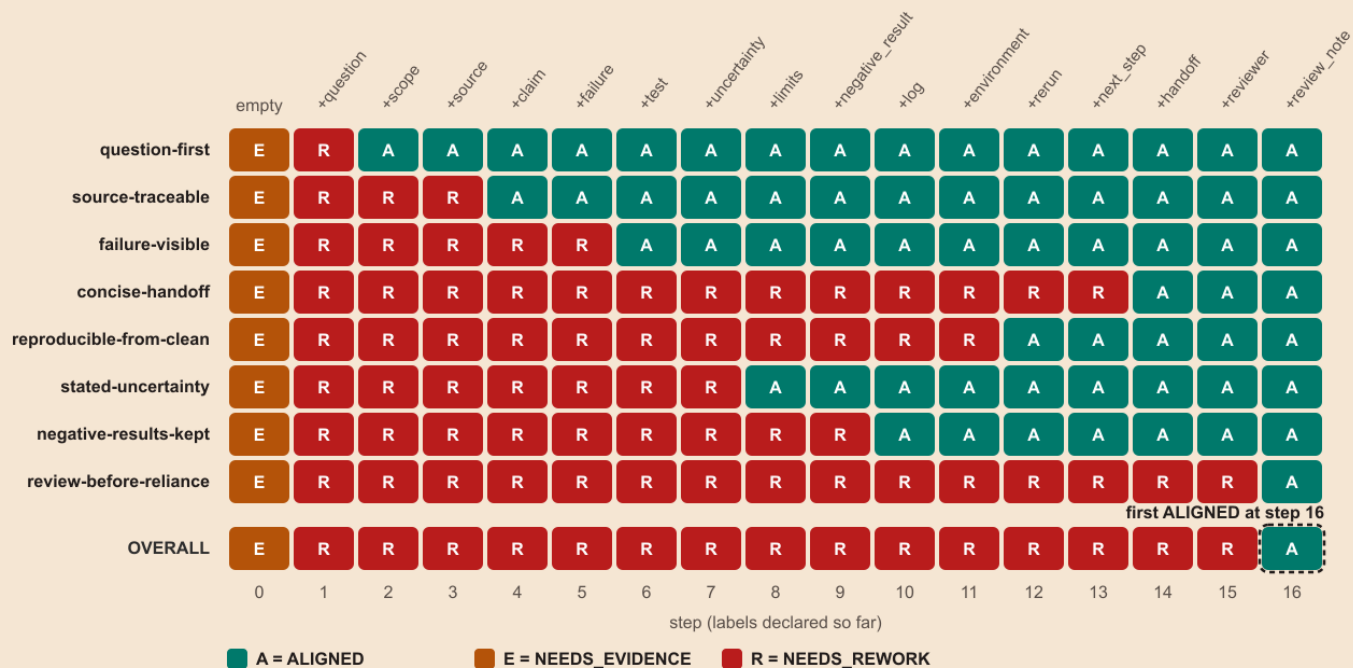
That first transition is intentional, and [Proposition 6](#) says why: an attempt with no usable evidence returns `NEEDS_EVIDENCE` because there is nothing to inspect, while one irrelevant label moves it to `NEEDS_REWORK` because the attempt has begun declaring and still omits every required label. The change reports a more specific request, not worse work, which is why the monotonicity claim is scoped to the non-empty branch. The `ALIGNED` at step 16 carries only its declared labels and the registry digest that pins the method version behind them.

One declaration, re-evaluated after every label

Each column is one real `evaluate_work` call on the same attempt as labels accumulate from none to all 16.

EXECUTED PATH · TAG 'research' · 8 APPLICABLE PRACTICES · 16 REQUIRED LABELS · 17 EVALUATOR CALLS

Rows are per-practice findings; the bottom row is the overall status — the most demanding per-practice status at each step.



Boundary Every cell reports declaration coverage from a real evaluator call; accumulated ALIGNED findings verify no source, test, or claim, and grant nothing.

Figure 40: The executed incremental-declaration path as a status grid: the **research**-tagged attempt selects 8 practices (16 required labels), and every cell is one real `evaluate_work` call as labels accumulate one per step from an empty declaration (step 0, **NEEDS_EVIDENCE**) to full coverage (step 16). Per-practice findings flip to **ALIGNED** as each practice's labels complete, while the overall status — the most demanding per-practice status — stays **NEEDS_REWORK** from step 1 through step 15 and first reaches **ALIGNED** at step 16: conditional fresh-evidence monotonicity made visible. An **ALIGNED** cell records declared labels only; it does not show any source is real, any test passed, or any claim is true.

The same property, swept over declaration orders

Every cell is one real evaluate_work call: 12 seeded orders of the same 16 labels.

EXECUTED SWEEP · SEED 20260727 · TAG 'research' · 12 ORDERS × 17 STEPS · 204 EVALUATOR CALLS

Row 0 is the registry's own declaration order — the single trace the incremental-path figure draws.

The remaining rows are seeded permutations of it, so the claim is sampled rather than proven.

order	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	monotone after step 1
00 · +question	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	yes
01 · +limits	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	yes
02 · +next_step	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	yes
03 · +scope	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	yes
04 · +question	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	yes
05 · +question	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	yes
06 · +handoff	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	yes
07 · +next_step	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	yes
08 · +source	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	yes
09 · +negative_result	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	yes
10 · +reviewer	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	yes
11 · +next_step	E	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	A	yes

■ A = ALIGNED
 ■ E = NEEDS_EVIDENCE
 ■ R = NEEDS_REWORK

Counted 12 of 12 orders are monotone from step 1 onward; 12 rank decreases occur in the whole sweep, and 12 of them are the step 0 to step 1 empty-declaration boundary.

Boundary A seeded sample of orders, not a proof over all of them, and a statement about status ordering only — no cell shows that a declared label points at anything.

Figure 41: Conditional fresh-evidence monotonicity executed as a sweep rather than a single trace: 12 seeded orders (seed 20260727) of the same 16 labels, each evaluated at all 17 steps through the real evaluator, for 204 calls. Every row is monotone from step 1 onward (12 of 12), and all 12 rank decreases in the sweep are the step 0 to step 1 empty-declaration boundary. The overall status path is identical across every sampled order, which is a property of taking the most demanding per-practice status, not evidence that order is irrelevant to a reviewer. The sample is seeded and finite; it is not a proof over all 16! orders.

4.9.2 The decay sweep, executed

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

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

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

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

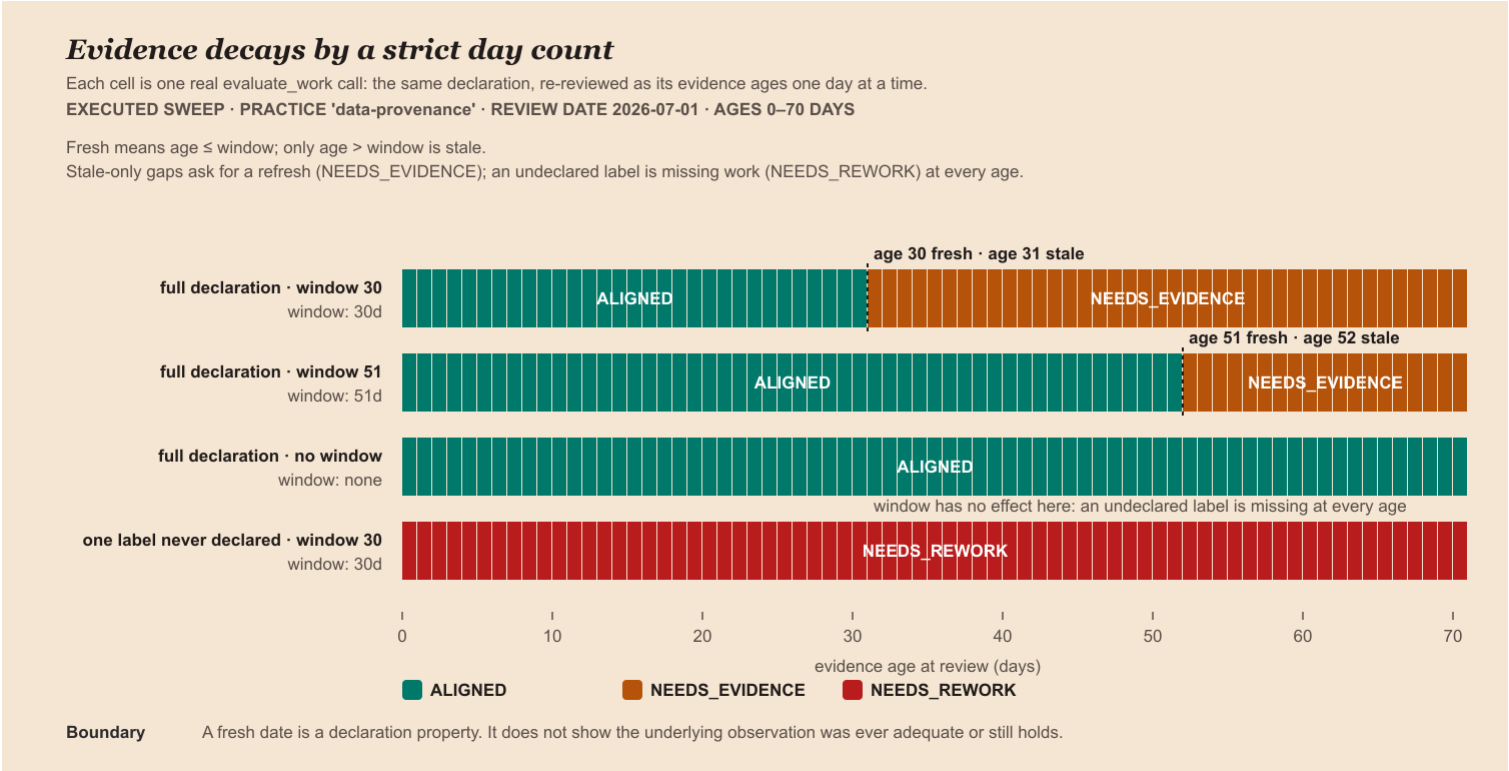


Figure 42: Evidence decay from executed `evaluate_work` sweeps over the `data-provenance` practice with review date 2026-07-01: a full declaration stays `ALIGNED` while its evidence age is at most the freshness window and flips to `NEEDS_EVIDENCE` (a refresh request) exactly one day past it — the threshold is a strict inequality — while a declaration that never included one required label is `NEEDS_REWORK` at every age, and a declaration with no window never goes stale. A fresh date is a declaration property; it does not show the underlying observation was ever adequate or still holds.

4.9.3 The refresh queue, executed

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

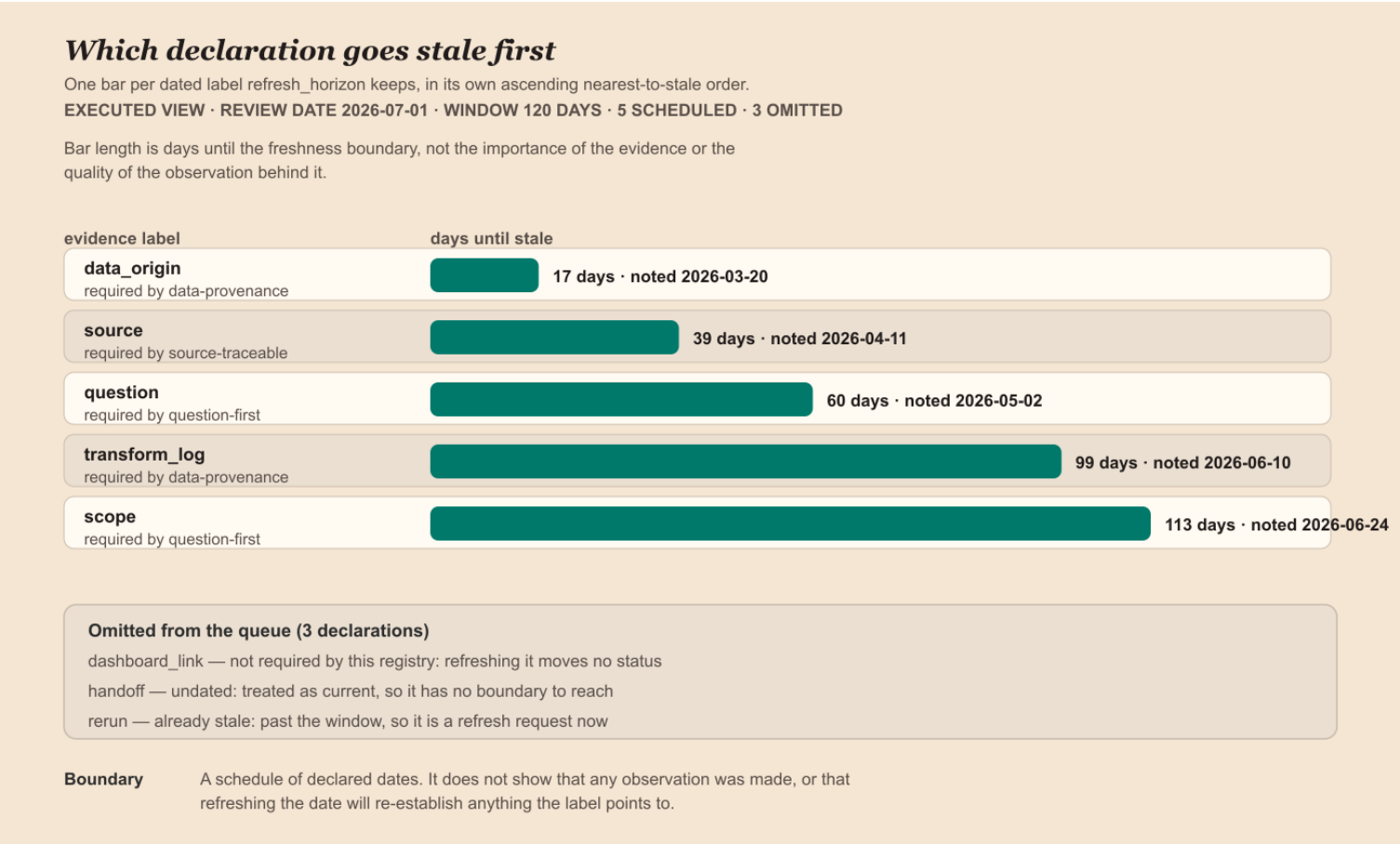


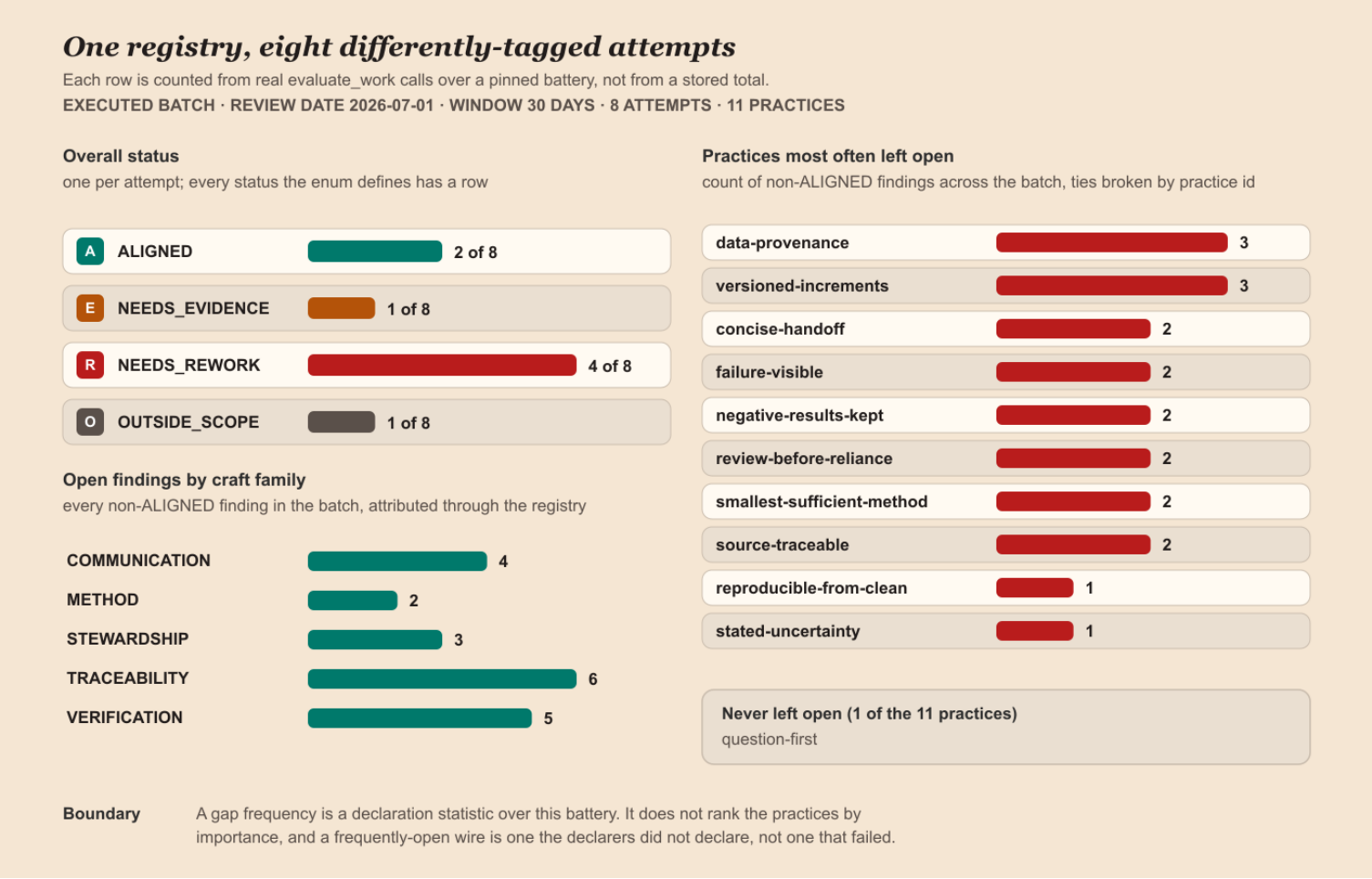
Figure 43: The refresh queue for one declaration, drawn from `refresh_horizon` in its own ascending nearest-to-stale order: at review date 2026-07-01 under a 120-day window, 5 dated labels are scheduled, from ‘data_origin’ at 17 days to ‘scope’ at 113, each naming the practice that requires it. The named band below records the 3 declarations the queue omits and why — one of them undated, which is itself one of the omitted classes — so the omission rule is visible rather than implicit. Bar length is days until a declared date crosses the window; it is not a measure of how much the underlying observation matters or how good it was.

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

4.9.4 A batch, executed

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

The batch shows something no single attempt can. `data-provenance` and `versioned-increments` are open most often not because they are harder practices but because of which tags reach them — `analysis` and `data` for the first, `engineering` and `writing` for the second — and the attempts carrying those tags here rarely declared the matching labels. `question-first` is the only practice the batch never leaves open, which says that the attempts declaring `analysis`, `research`, or `writing` all named a question and a



scope, and nothing more. Read as a review artifact, the panel is a prompt: it names where declarations are thin across a body of work, and it stops there. It does not say those wires were done badly, or that the two **ALIGNED** attempts were done well.

4.9.5 Boundary cases

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

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

4.10 Limits and Epistemic Boundaries

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

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

4.10.1 Adversarial declarations

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

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

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

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

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

4.10.2 Legibility bias

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

4.10.3 A design claim, not an outcome claim

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

4.10.4 Bounded by the line set

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

4.11 Conclusion

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

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

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

5 Golden Line: Toward What Matters

An Aspirational Thread for Long-Horizon Work

Reproduced unchanged in substance from its own source at version 0.4.0. It answers one question: What is worth reaching toward?



Figure 45: Cover art for Golden Line: Toward What Matters

5.1 Abstract

Golden Line is an aspirational thread: a compact, revisable way to say what a piece of work is trying to serve over a long horizon. It asks what is worth reaching toward after Red Line has named the boundary and alongside Black Line's operating discipline. It is deliberately not a compliance score. A directional reading never certifies that work is good, safe, lawful, or complete.

The executable instrument is a small Python package with a versioned aspiration registry of nine entries — four founding aspirations and five further ones — and a staged **progress_report** evaluator. Each aspiration pairs a plain-language thread and a concrete horizon with two short lists: Markers, the observable signs of movement toward it, and Counter-signals, the observable signs of movement away. An observer files a horizon entry, optionally dated, recording which of those were seen. The evaluator screens and normalizes the entry, matches it against the registry, preserves a structured derivation, and returns one of four directional readings: **TOWARD**, **INQUIRY**, **DRIFTING**, or **NOT_OBSERVED**. The report carries the registry digest and the temporal review context, so a reading is never detached from the vocabulary and the date that produced it.

The paper develops the instrument in three registers. A **formal method** section states the domain objects, the staged evaluator, and its decision rule as definitions and propositions that match the code exactly, each bound to the named test that would fail if the code diverged, and records seven structural invariants, each proven by a planted-bad proof-of-detection test. A **scholarship** section situates the founding aspirations — attention before output, usefulness beyond the author, repairable systems, and answerability to human flourishing — in a lineage running from practical wisdom and goods internal to a practice through the capability approach, reflective practice, repair, commons governance, technical power, and the hazards of turning an aspiration into a metric. Thirteen code-derived **figures** render the registry, its signal vocabulary, the decision path, the staged pipeline, the evidence-state matrix, the horizon bands, and a loop for the founding four. Six of the thirteen replay the evaluator rather than diagram it: one entry ageing past the currentness boundary, the whole registry ageing at once across 126 dated readings, every marker subset of every aspiration, the twelve fields of the returned record under six evidence conditions, a worked batch of six entries, and a precedence panel in which complete markers plus one counter-signal still return **DRIFTING** for every aspiration.

Golden Line is standalone and non-redundant with its companions. Its vocabulary is horizon, thread, marker, counter-signal, and direction. The absence of an observation is not a failure finding; it is an invitation to look again, or to leave the claim honestly open.

5.2 Introduction: the work needs a direction

Every method can become technically competent while losing contact with the reason it exists. A test suite stays green, a pipeline keeps shipping, and yet the question of what the work is *for* quietly falls out of view. Golden Line names that danger without pretending to dissolve it with a single principle. It provides a small, revisable set of directions that can orient a decision, a collaboration, or a long project, and a way to ask, at a stated horizon, whether the work is visibly moving toward them.

The idea has many ancestors, and the instrument inherits from all of them without settling their disagreements. Aristotle treats practical judgment as deliberation about a good life rather than the mechanical application of a rule, and makes an *end* (*telos*) the thing that gives an activity its point [Aristotle, 350 BCE]. Confucian teaching connects learning to the cultivation of conduct within relationships, so that skill and character grow together rather than apart [Confucius, 500 BCE]. Ibn Khaldun's account of cooperation, power, and social change reminds us that purposes are carried by institutions and shared cohesion, not by individual intention alone [Khaldun, 1377]. Closer to the present, MacIntyre's distinction between goods internal to a practice and its external rewards helps explain why useful work must remain more than a successful artifact [MacIntyre, 1981]. The scholarship section extends this lineage through capabilities, reflective practice, repair, commons governance, and the politics of technical systems. These references are prompts for comparison and calibration, not authorities the registry claims to reconcile.

Golden Line's role is deliberately limited, and the limit is the point. It cannot authorize work that Red Line refuses. It cannot certify that work was done well, which is Black Line's job. It cannot fill in what White Line marks as absent. The next section places each of those instruments; here it is enough to say that Golden Line absorbs none of their work. What it adds is a direction held lightly, stated in the open, and revisable when the evidence or the values change.

The paper proceeds from prose to formalism to lineage: the method section describes the staged evaluator in words, the formal section restates it as definitions and propositions faithful to the code, and the scholarship section grounds the founding aspirations in their sources. The remaining sections walk the nine-entry registry, worked records, the evidence boundary, and honest limits.

The manuscript is only half of it. Golden Line is operated day to day through the `daf-golden-line` skill in DAF's private `daf-skills` toolchain: this paper states what the instrument means and why, and the skill is how a horizon entry is actually filed.

The paper is organised as follows. [Section sec. 5.4](#) defines the method and evidence protocol. [Section sec. 5.5](#) states the evaluator formally. [Section sec. 5.8](#) presents the four aspirations. [Section sec. 5.9](#) walks through worked examples, and [Section sec. 5.11](#) closes with limits and epistemic boundaries.

5.3 Relationship to the line set

Golden Line is the aspirational work in the four-line set, whose declaration lives in the companion work `line_set`. Red Line is the personal security boundary and explicit No document. Black Line is positive operating discipline. White Line records absence, restraint, and negative space. Golden Line does not copy their registries or evaluators, and a directional finding is never a compliance verdict.

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

5.3.1 Note on the name

The name is not incidental. *Golden Line* openly echoes the yellowing stage of the alchemical *magnum opus*, **citrinitas**, the dawning that classical schemes place between the whitening and the reddening. The echo is meant in Carl Jung’s symbolic-psychological register, where the stages of the opus are read as a map of individuation, and in that register only: it is a metaphor for a practice turning toward what it is for, never an empirical, mystical, or causal claim about matter, minds, or this instrument. The working order of the four-line set refuse, then method, then aspire, then absence, is functional and chosen for how the instruments are actually used, and deliberately does not reenact the opus’s canonical sequence (nigredo → albedo → citrinitas → rubedo). Citrinitas here simply marks Golden Line’s place in the set: the yellowing where a disciplined practice lifts its attention from how the work is done to what the work is worth. The grounding citation is Jung’s reading of the opus stages as figures for individuation rather than as laboratory chemistry [Jung, 1953], and it is carried here rather than deferred, so this paper does not depend on another document to say what its own name means. The set-level version of the framing — the two orders, the caveats, and the non-overlap contract — is declared in the companion work `line_set`; this paper points there for the set, not for its own citation.

5.4 Method: aspiration as a directional record

5.4.1 The registry entry

An aspiration has six fields: an identifier, a plain-language title, a thread that explains its direction, a horizon at which the direction becomes visible, and two short lists. **Markers** are observable signs that the work is moving in the declared direction. **Counter-signals** are observable signs that it is drifting away from it. Neither list is exhaustive, and neither is a questionnaire; they are the small, concrete anchors that let a direction be discussed rather than merely admired. The registry ships nine such entries: four founding aspirations and five further ones that fill out the long-horizon picture.

5.4.2 The horizon entry

An observer records movement by filing a **horizon entry**: the aspiration identifier, the set of markers actually observed, the set of counter-signals actually observed, a free note, and an optional ISO observation date. The entry describes a *record* of work at a moment, not a person and not a project as a whole.

5.4.3 The staged evaluator

`progress_report` reads a batch of horizon entries against the versioned registry in three stages, and every finding it returns carries a full reasons trail plus structured derivation fields so no reading arrives unexplained.

1. **Intake screening.** Each incoming entry is checked for the expected record shape and against the known aspiration identifiers. An entry for an unknown aspiration is set aside with a note; a second valid entry for an aspiration already seen is set aside, and the first entry stands. JSON-like token lists are normalized to sets. Malformed or hostile input is *recorded*, never raised. A sloppy record cannot crash the report or vanish silently.
2. **Matching.** For each aspiration, the observed markers are intersected with the declared markers, and the observed counter-signals with the declared counter-signals. Tokens the observer supplied that the registry never declared are ignored and noted; they cannot smuggle in a status.
3. **Decision.** With optional temporal review folded in, the matched sets and temporal quality determine one of four directional readings. There is no numeric score or aggregate grade.

The four readings are:

- **TOWARD** means every declared marker is observed, no counter-signal is recorded, and (if temporal review is enabled) currentness is auditable: the observation date is present, parseable, and within the declared window.
- **INQUIRY** means the aspiration is named but the evidence is partial, empty, or stale, so the direction remains honestly open.
- **DRIFTING** means a declared counter-signal is present. It describes the record, not the person or project as a whole.
- **NOT_OBSERVED** means no valid horizon entry was admitted for that aspiration. An entry can be present in the input but fail intake because it is malformed, unknown, or a duplicate set aside by the first-valid-entry rule.

The finding also preserves the note, observed date, declared counter-signals, and undeclared tokens that were ignored. This makes the status machine-readable without requiring a downstream reader to reverse-engineer human-readable reason strings.

5.4.4 Conservative precedence

The precedence is intentionally conservative. A counter-signal is surfaced before any markers are counted, and staleness never erases a recorded drift: a declared counter-signal yields **DRIFTING** even when the same observation is old. Partial, stale, or currentness-un-auditable positive evidence reverts to **INQUIRY** rather than being rounded up to **TOWARD**. This mirrors a satisficing stance: the evaluator looks for *good enough and current* evidence of direction and refuses to over-read thin signals [Simon, 1956].

The result is a directional report. It is not an evaluator of worth, a certification, or a substitute for the Red Line, Black Line, or White Line instruments.

5.4.5 Evidence and artifact boundary

The machine has four layers: the versioned registry declares the vocabulary; an admitted entry records a local observation; the evaluator derives a bounded reading; and the generated registries and figures preserve what source was used. No layer is allowed to add a claim that the earlier layer did not carry. The report's registry version, registry digest, review date, and staleness threshold therefore travel with the findings. The local artifact gate checks that the generated JSON, SVG, PNG, and manuscript figure labels still agree before a sibling template render is attempted.

5.4.6 The descriptive analysis layer

A small analysis module sits beside the evaluator and is deliberately weaker than it: every helper is pure, deterministic, and read-only, and none can change what a reading means. Four helpers are provided.

- **signal_inventory** tallies the declared markers and counter-signals across a registry — for the shipped registry, 18 markers and 9 counter-signals, with every token distinct — describing the vocabulary the evaluator can match, never its fulfilment.
- **horizon_distribution** groups aspirations into four declared temporal-reach bands (immediate, recurring cycle, at handoff, open-ended). The band map lives in the analysis layer, not the registry contract, and an unclassified horizon raises an error so a registry change must revisit the map deliberately.
- **temporal_currentness_sweep** replays one fully-marked entry through the public **progress_report** across a range of observation ages, exposing the exclusive staleness boundary (current at age 90, stale at 91 for a 90-day window) as an observed trajectory rather than a claim.
- **report_overview** regroups an existing report by status and totals its ignored tokens, temporal flags, and intake notes, so a batch can be characterized without re-parsing prose reasons.

The analysis layer is also where the project’s experiment plan is grounded. The repository ships a **domain_profile.yaml** naming the validation gates: structural invariants, evidence grounding, artifact chain, render validation, and publication readiness. It also ships an **experiment_plan.yaml** whose three conditions — the source-registry baseline, the malformed-input guard, and the temporal-currentness guard — map onto the evaluator’s intake stage and temporal review. The plan’s expected figures are the ones the deterministic builder produces: five through this analysis layer’s helpers (the temporal-currentness sweep, the currentness lattice, the signal inventory, the horizon bands, and the batch reading overview) and eight straight from the registry, the evaluator, and the status contracts. The protocol compares reproducible contract outcomes only. None of these summaries is evidence that any aspiration is true, and none is a score.

5.5 Formalism: the evaluator and its invariants

The formalism restates the evaluator the method section describes; every result below tracks the code that implements it. This section states the implemented semantics. Every result below describes what the code in `golden_line` actually does, and the reasons and transitions named here are the ones the evaluator produces. The formalism *describes* the instrument; it does not extend or idealize it. Numbering is assigned by the renderer in document order, so no number is written in the source and none can go stale.

5.5.1 Domain objects

Definition 1 (Aspiration). An aspiration a is the six-tuple

$$a = (\text{id}, \text{title}, \text{thread}, \text{horizon}, M, C),$$

where `id`, `title`, `thread`, `horizon` are text fields and M (Markers) and C (Counter-signals) are finite sequences of observable tokens. Markers are signs of movement *toward* the aspiration; Counter-signals are signs of movement *away* from it.

Definition 2 (Registry). The registry $R = \langle a_1, \dots, a_n \rangle$ is an ordered tuple of aspirations with $n = 9$: four founding aspirations (attention before output, usefulness beyond the author, repairable systems, answerability to human flourishing) followed by five further ones (durable understanding, teachable craft, honest uncertainty, unhurried questions, improvements returned to the commons). Write $\text{ids}(R)$ for the set of identifiers appearing in R .

Definition 3 (Horizon entry). A horizon entry is $e = (\text{aid}, O, K, \text{note}, d_{\text{obs}})$, where `aid` names an aspiration, O is the set of observed markers, K is the set of observed counter-signals, `note` is free text, and d_{obs} is an optional ISO date.

Definition 4 (Status codomain). The directional readings form the four-element set

$$\Sigma = \{\text{TOWARD}, \text{INQUIRY}, \text{DRIFTING}, \text{NOT_OBSERVED}\},$$

copied verbatim from the `HorizonStatus` enumeration. These are the *only* statuses the evaluator can emit.

Definition 5 (Finding). A finding is

$$f = (\text{aid}, \sigma, \text{reasons}, \text{observed}, \text{unmet}, \text{countered}, \text{ign}_M, \text{ign}_C, \text{note}, d_{\text{obs}}, \text{stale}, \text{date_issue})$$

with $\sigma \in \Sigma$ (Definition 4). The structured fields list the declared markers observed and unmet, the declared counter-signals, the undeclared marker tokens ign_M and counter-signal tokens ign_C that were ignored, the original note and date, and two temporal quality flags. Ignored markers and ignored counter-signals are kept apart rather than pooled, because a reader checking why a token did nothing needs to know which vocabulary it failed to match. Human-readable **reasons** remain a parallel explanation, not the only source of derivation.

5.5.2 The staged evaluator

Definition 6 (Report function). The evaluator has signature

$$\text{progress_report} : (E, R, \text{as_of}, \tau) \rightarrow \text{HorizonReport},$$

where E is a batch of horizon entries, R defaults to the registry of Definition 2, `as_of` is an optional review date, and τ (`stale_after_days`) is an optional non-negative integer staleness threshold; `as_of` and τ are keyword-only. It runs three stages: intake screening, matching, and decision. The returned report also identifies the registry version/digest and the review context.

Definition 7 (Intake screening). Screening builds an accepted map $A : \text{ids}(R) \rightarrow E$ by a first-wins rule. Iterating E , a record with the wrong type or malformed fields is set aside with an intake note; an entry whose `aid` $\notin \text{ids}(R)$ is set aside with an unknown-id note; an entry whose `aid` is already in A is set aside with a duplicate note and the first valid entry stands; otherwise the entry is admitted. Iterable text-token fields are normalized to sets. Screening never raises on malformed or unexpected input. It is a pure function of the pair $(E, \text{ids}(R))$: records are tested in a fixed order — shape, then registry membership, then prior admission — and iteration follows the order of E with indices counted from 1 in the intake notes.

Definition 8 (Matching). For aspiration a (Definition 1) with admitted entry $e = A(a.\text{id})$ (Definition 3), define

$$\text{observed}(a, e) = O \cap M, \quad \text{unmet}(a, e) = M \setminus O, \quad \text{countered}(a, e) = K \cap C.$$

Undeclared tokens $O \setminus M$ and $K \setminus C$ are ignored and recorded in the reasons; they can never determine a status.

Definition 9 (Temporal review). Given review date d and threshold τ , temporal review is enabled exactly when τ is set. If τ is unset, temporal metadata does not affect the status. If τ is set, a missing or unparseable d_{obs} sets `date_issue` to true: the

record is not fatal, but it cannot support a currentness claim. With a parseable date, $\text{stale}(e, d, \tau)$ holds iff $d_{\text{obs}} > d$ (a future observation) or $d - d_{\text{obs}} > \tau$ days.

Definition 10 (Decision rule). The status $\sigma(a, e)$ is determined by the first matching clause, in order:

1. if $e = \perp$ (no admitted entry): NOT_OBSERVED;
2. else if $\text{countered}(a, e) \neq \emptyset$: DRIFTING;
3. else if $\text{observed}(a, e) = \emptyset$: INQUIRY;
4. else if $\text{unmet}(a, e) \neq \emptyset$: INQUIRY;
5. else if $\text{stale}(e, d, \tau)$ or date_issue : INQUIRY;
6. otherwise: TOWARD.

5.5.3 Propositions about the evaluator

Each proposition below follows from [Definition 7](#) through [Definition 10](#) and is exercised by the package’s test suite; the binding table at the end of this section names the verifying test for every result. Each proposition is a statement about code behavior — never about the world, safety, or persons.

Proposition 1 (Totality). For every aspiration $a \in R$ the evaluator emits exactly one finding ([Definition 5](#)), so a report contains $n = 9$ findings, one per registry entry, whether or not any entry was filed. The report’s `counts` therefore partition the findings across Σ .

Proposition 2 (Counter-signal precedence). If a declared counter-signal is recorded, meaning $\text{countered}(a, e) \neq \emptyset$ ([Definition 8](#)), the status is DRIFTING, regardless of how many markers were observed and regardless of staleness. Clause 2 precedes clauses 3–6, so a positive signal can never launder a recorded drift.

Proposition 3 (Exactness and currency of TOWARD). $\sigma(a, e) = \text{TOWARD}$ iff $\text{observed}(a, e) = M$ (every declared marker seen), $\text{countered}(a, e) = \emptyset$ ([Definition 8](#)), and neither $\text{stale}(e, d, \tau)$ nor date_issue holds. Any missing marker, any counter-signal, stale observation, or unparseable reviewed date reverts the reading to INQUIRY or DRIFTING. Counter-signal precedence is as stated in [Proposition 2](#). There is no partial credit: an entry recording every marker but one reads exactly as an entry recording none.

Proposition 4 (Temporal uncertainty reopens, never condemns). A fully-marked but stale or date-un-auditable observation with no counter-signal yields INQUIRY (clause 5), not DRIFTING. Age or malformed temporal metadata reopens a question; it does not manufacture drift.

Proposition 5 (Absence is not negation). NOT_OBSERVED arises solely from the absence of an admitted entry (clause 1). The admission rule of [Proposition 6](#) determines which entries are admitted; when none is, the reading is NOT_OBSERVED regardless of what records may have been submitted. It asserts nothing about whether the aspiration is being served; it reports only that no valid record was available to this report. The intake notes preserve why submitted records were set aside when that distinction matters.

Proposition 6 (First valid entry stands; the reading is deterministic). For each identifier, screening admits the first valid entry bearing it in the order of E : the first valid entry bearing an identifier stands, and every later entry with that identifier is set aside with a duplicate note. For an identifier with no valid entry, the reading is NOT_OBSERVED ([Proposition 5](#)). Exchanging the positions of two entries that share an identifier can therefore change the finding, but nothing else about the call can: replaying an identical batch with identical `as_of` and τ reproduces the identical report, digest for digest. Determinism is a property of the code path; it does not make the underlying records true.

Proposition 7 (Intake screening determinism). The intake screening function ([Definition 7](#)) is a pure function of the pair $(E, \text{ids}(R))$: records are tested in a fixed order — shape, then registry membership, then prior admission — and iteration follows the order of E with indices counted from 1 in the intake notes. For the identical batch E and the identical known `ids` $\text{ids}(R)$, screening produces the identical accepted map A and the identical intake notes every time. Malformed input is set aside deterministically: a record with the wrong type or malformed fields always gets a malformed note, an entry whose `aid` $\notin \text{ids}(R)$ always gets an unknown-id note, and an entry whose `aid` is already in A always gets a duplicate note. Screening never raises on malformed or unexpected input — it is a pure computation, and its determinism is separate from the first-wins semantics of [Proposition 6](#), which covers which entry stands when multiple valid entries share an identifier. Intake screening determinism covers the screening function’s behavior on all input, including malformed records.

Proposition 8 (The currentness boundary is exclusive). Staleness uses the strict comparison $d - d_{\text{obs}} > \tau$ of [Definition 9](#), so a fully-marked, counter-signal-free entry with $\tau = 90$ reads TOWARD at age exactly 90 days and reverts from age 91 days: age τ is the last current age and $\tau + 1$ the first stale one. The threshold is the caller’s review-cadence choice; nothing in the evaluator endorses any particular number of days as a natural constant.

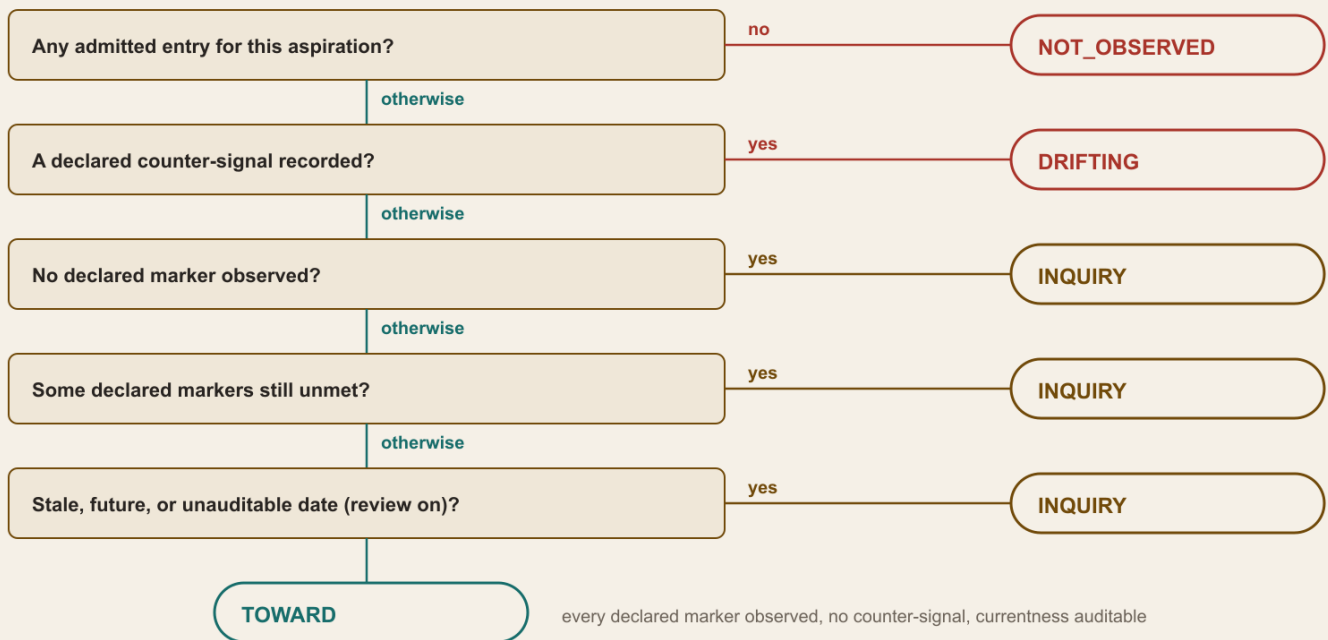
The ordered clauses and the four readings they terminate in are drawn from the `HorizonStatus` enumeration — the decision rule’s codomain, not the registry — in [fig. 46](#).

THE HORIZON DECISION PATH

How one entry becomes one directional reading

Clauses are tested in order; the first that matches decides.

A counter-signal outranks every marker.



SOURCE-DRIVEN SCHEMATIC · statuses quoted from the HorizonStatus enumeration · NOT A COMPLIANCE VERDICT

Figure 46: The evaluator's ordered decision rule. Counter-signal precedence is explicit; TOWARD requires complete markers and auditable currentness when temporal review is enabled. The path reads a record; it is not a compliance verdict.

5.5.3.1 The completeness half of the rule The exactness clause of **Proposition 3** is the one a reader is most likely to soften into “mostly there”. fig. 47 refuses that reading by running it: every subset of every aspiration’s declared markers is filed as an entry and evaluated, and the panel prints the status the evaluator returned alongside the number of markers left unmet. Only the complete subset reads TOWARD; a single missing marker holds the direction open at INQUIRY, exactly as an empty entry does. The panel characterizes the clause order, not any observed practice.

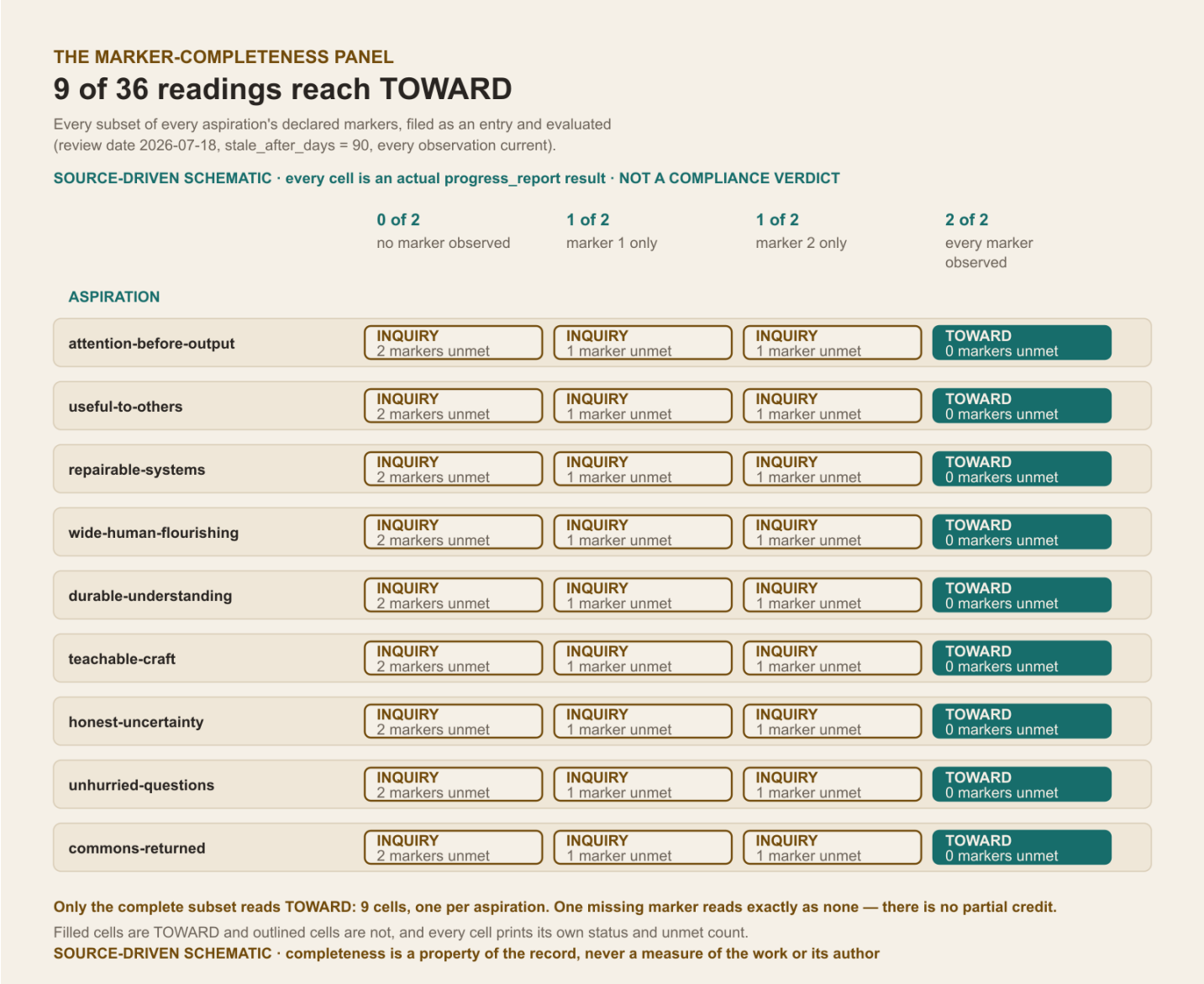


Figure 47: Every observed-marker subset for every aspiration, filed as an entry and evaluated. Each cell prints the status progress_report returned and the count of markers still unmet, so the grid reads without colour; only the complete subset reaches TOWARD, and one missing marker reads exactly as none. Marker completeness is a property of the record, never a measure of the work.

5.5.3.2 The temporal half of the rule **Proposition 4** has a picture. The temporal_currentness_sweep analysis helper replays one fully-marked, counter-signal-free entry through the real evaluator at a range of observation ages with $\tau = 90$: the reading holds TOWARD through age 90 exactly (the boundary is exclusive), reverts to INQUIRY from age 91, and a future-dated observation likewise reads INQUIRY. Every cell in fig. 48 is an actual progress_report result, not an illustration of one; the sweep observes the evaluator and never re-implements it.

Proposition 9 (Sweep delegation and codomain). The sweep computes no status of its own: its only status-producing call is progress_report (Definition 6), and each SweepPoint copies that call’s finding. This is an architectural property, not an empirical one — comparing a sweep point against a second progress_report call would compare the same code path with itself, so the verifying test instead asserts that analysis.py constructs no HorizonStatus anywhere. Because sweep entries are fully marked, counter-signal-free, and bear a known identifier, only clauses 5 and 6 of Definition 10 can fire: DRIFTING and NOT_OBSERVED are

THE TEMPORAL CURRENTNESS SWEEP

How one reading ages under temporal review

A fully-marked entry for 'attention-before-output' replayed through the real evaluator at each observation age (review date 2026-07-18, stale_after_days = 90).

SOURCE-DRIVEN SCHEMATIC · every cell is an actual progress_report result · NOT A COMPLIANCE VERDICT

INQUIRY -7 d (future) 2026-07-25 stale	TOWARD 0 d 2026-07-18 current	TOWARD 15 d 2026-07-03 current	TOWARD 30 d 2026-06-18 current	TOWARD 45 d 2026-06-03 current	TOWARD 60 d 2026-05-19 current	TOWARD 75 d 2026-05-04 current
exclusive boundary: age 90 is current, age 91 is stale						
TOWARD 89 d 2026-04-20 current	TOWARD 90 d 2026-04-19 current	INQUIRY 91 d 2026-04-18 stale	INQUIRY 105 d 2026-04-04 stale	INQUIRY 120 d 2026-03-20 stale	INQUIRY 150 d 2026-02-18 stale	INQUIRY 180 d 2026-01-19 stale

TOWARD **INQUIRY**

READING RULE age or a future date reopens the question (INQUIRY); it never manufactures drift

A stale reading is an invitation to look again, not evidence of failure.

Figure 48: A fully-marked entry replayed through the real evaluator at 14 observation ages with stale_after_days = 90. Every cell names the status progress_report actually returned, so nothing here is carried by colour alone: TOWARD through age 90, INQUIRY from age 91 and for future-dated observations. Age reopens the question; it never manufactures drift.

unreachable in a sweep; that half *is* falsifiable and is tested by enumeration. The sweep characterizes the instrument’s temporal behavior; it does not describe any observed practice.

Proposition 8 states the boundary and the sweep shows it for one registry entry. Whether the boundary belongs to the *evaluator* rather than to that one entry is a separate question, and one that a single row cannot answer.

Proposition 10 (The currentness boundary is uniform across the registry). Replaying the sweep (**Proposition 9**) for every $a \in R$ at the same review date and threshold yields the same TOWARD $\rightarrow$ INQUIRY transition age for every aspiration, because **Definition 9** reads only d_{obs} , d , and τ — no field of a enters the staleness comparison. The sweep confirms that the temporal-inquiry behaviour stated in **Proposition 4** and the exclusive boundary stated in **Proposition 8** are properties of the evaluator’s code path, not of any particular aspiration. The lattice in fig. 49 is that replay: $9 \times 14 = 126$ executed **progress_report** calls, with each row’s own transition age printed beside it and a count of rows that differ. Uniformity here is a fact about the code path; it says nothing about how any aspiration is actually served.

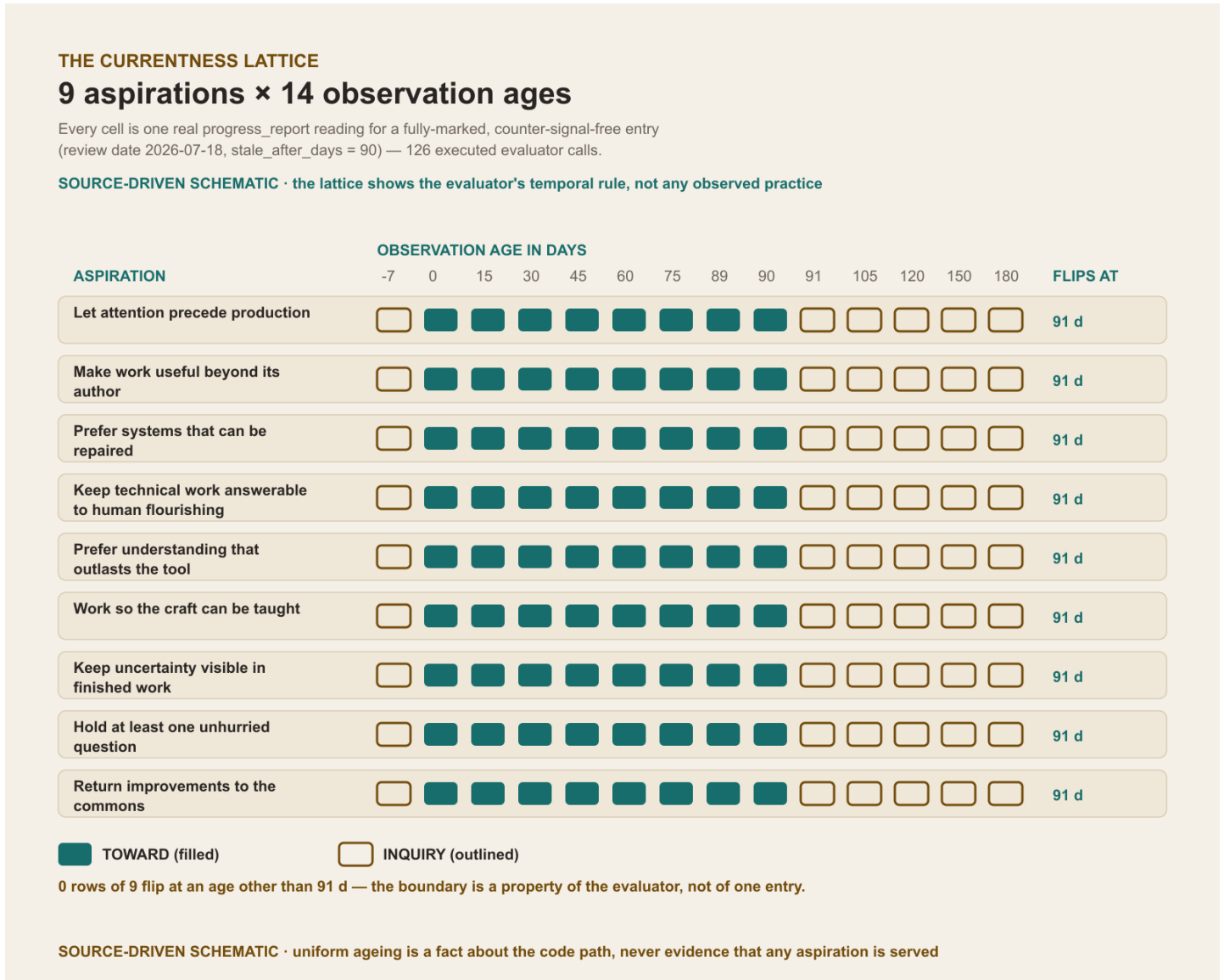


Figure 49: The same currentness replay run across the whole registry: 9 aspiration rows by 14 observation ages, 126 executed progress_report calls at stale_after_days = 90. Filled cells are TOWARD and outlined cells are INQUIRY, so the grid reads without colour; the FLIPS AT column gives each row’s own transition age and the footer counts rows that differ. Uniform ageing is a fact about the evaluator’s code path, not evidence that any aspiration is being served.

5.5.4 Structural invariants

The evaluator reads a registry; a malformed registry would make its readings meaningless. Seven pure-compute structural checks validate the *shape* of the registry, independent of any horizon entry. Let $I_1, \dots, I_7$ be their predicates; **registry_sound**(R) returns

$$\bigwedge_{k=1}^7 I_k(R).$$

- I_1 **distinct identifiers**: no two aspirations share an id, so findings are unambiguous.
- I_2 **fields populated**: id, title, thread, and horizon all carry content.
- I_3 **reachable statuses**: each aspiration has at least one marker and at least one counter-signal, so both TOWARD and DRIFTING are reachable for it.
- I_4 **signal text**: every marker and counter-signal is a non-blank string.
- I_5 **signal uniqueness**: no marker or counter-signal repeats within one declaration.
- I_6 **signal disjointness**: $M \cap C = \emptyset$ within each aspiration, so no token reads as movement toward and away at once.
- I_7 **digest stability**: the canonical registry serializes, and its SHA-256 digest is independent of registry order.

Proposition 11 (Digest order-independence). `canonical_registry` serializes aspirations sorted by identifier, so $\text{registry_digest}(R) = \text{registry_digest}(\pi(R))$ for every permutation π of R , and the digest is exactly 64 lowercase hexadecimal characters (SHA-256). Two readers comparing digests are comparing registry content, never the order in which they happened to list it — and a matching digest attests only sameness of content, not soundness or merit.

Proposition 12 (Proof of detection). For each invariant I_k , the test suite asserts both that I_k *passes* on the real registry and that I_k *fails* on a deliberately planted-bad registry constructed to violate exactly that check. A green check that had never seen a bad input would not count as evidence, so each invariant is paired with the counter-example that gives it meaning.

These digests and checks are review and drift-detection instruments for humans comparing registry revisions. They carry no safety, warranty, or attestation semantics of any kind. The detection proof of [Proposition 12](#) establishes that the structural invariants can fail — each has a planted counter-example — and therefore a green check is a positive result, not a silent absence. [Proposition 11](#) gives the digest its order-independence; detection gives the invariants their falsifiability.

5.5.5 The report envelope

A reader holding reports from several independent instruments needs one uniform way to say “this instrument, about this subject, at this review moment, said this — and here is the pointer to its complete native report.” The envelope is that data contract and nothing more. For this instrument the usual worry — that a selected status becomes a safe projection mistaken for the whole state — takes a specific form: Golden Line has *no* single overall verdict, so the only honest transportable status is the complete ordered vector of per-aspiration readings. The envelope carries exactly that vector, in report order, and invents no summary above it; compressing nine directional readings into one word would manufacture precisely the aggregate virtue score this instrument refuses everywhere else.

Definition 11 (Report envelope). The report envelope is the frozen record $v = (\text{schema_version}, \text{line_id}, \text{subject_id}, \text{review_date}, \text{native_status}, \text{report_ref}, \text{scope_and_nonclaims})$ with exactly those ten fields, in order, exported under the schema string `line.report-envelope/1.0`. `native_status` is the complete ordered sequence of `(aspiration_id, status)` pairs from the report’s findings — this line’s own vocabulary, one pair per registry aspiration ([Proposition 1](#)), never a summary. `report_ref` is the SHA-256 of the complete canonical report, so the envelope points at the full derivation — reasons trails, matched and ignored tokens, temporal flags, intake notes — rather than copying or restating any of it. `scope_and_nonclaims` carries the instrument’s transportable non-claims inside the record itself, 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 on `native_status`.

Proposition 13 (The envelope points, never reinterprets). For every report r , `report_envelope(r)` satisfies `envelope_matches_report(envelope, r)`: the digest pointer, the review date, the registry version and digest ([Proposition 11](#)), and the per-aspiration readings all agree with the report they were exported from, and editing any of them afterwards makes the check return false. The envelope ([Definition 11](#)) adds no field the report does not determine except the caller-supplied `subject_id` and `source_snapshot_refs`, which the evaluator stores and does not verify. A matching envelope attests that the pair was archived unedited; it says nothing about the truth of the report or the merit of anything the report read.

5.5.6 Formalism-to-test bindings

Every definition and proposition above is verified by named tests in the package’s suite; the tables below bind each result to the test that would fail if the code stopped satisfying it. Each row is keyed on the block’s *label*, not on its number, and the label renders as the number the reader sees, so inserting a result rennumbers the prose and the table together and can never split them.

Two binding tests police the tables. `tests/test_formalism_bindings.py::test_binding_tables_bind_every_declared_block` fails if the set of row labels stops matching the set of labels declared in this section, and `tests/test_formalism_bindings.py::test_every_binding_row_names_an_existing_test` fails per row if any row’s verifying-test cell names no test or names one that does not exist. Neither checks that a named test is a *good* test, only that every declared block is bound to one that exists. The boundary column restates what each result does *not* claim.

Definition	Statement essence	Verifying test	Boundary
Definition 1	six fields, two of them token sequences	<code>tests/test_formalism_bindings.py::test_aspiration_tuple_matches_the_dataclass</code>	names the fields, not what a marker is worth
Definition 2	nine entries, four founding then five further	<code>tests/test_formalism_bindings.py::test_registry_definition_matches_the_source_tuple</code>	a versioned design choice, not a canon
Definition 3	five fields, the observation date optional	<code>tests/test_formalism_bindings.py::test_horizon_entry_tuple_matches_the_dataclass</code>	records an observation, not a person
Definition 4	exactly the four <code>HorizonStatus</code> values, in enum order	<code>tests/test_formalism_bindings.py::test_status_codomain_matches_the_enumeration</code>	four readings, not a scale from bad to good
Definition 5	twelve fields, ignored markers kept apart from ignored counter-signals	<code>tests/test_formalism_bindings.py::test_finding_tuple_matches_the_dataclass</code>	exposes a derivation, not a justification
Definition 6	four parameters, two keyword-only; three named stages	<code>tests/test_formalism_bindings.py::test_report_function_signature_matches_manuscript</code>	a call shape, not a guarantee about inputs
Definition 7	shape, then membership, then prior admission; notes indexed from 1; never raises	<code>tests/test_formalism_bindings.py::test_intake_screening_tests_records_in_the_stated_order</code> , <code>tests/test_formalism_bindings.py::test_intake_screening_never_raises_on_hostile_input</code> , <code>tests/test_formalism_bindings.py::test_intake_first_wins_and_replay_determinism_match_manuscript</code>	screening judges records, never their authors
Definition 8	the three set operations, and undeclared tokens discarded	<code>tests/test_formalism_bindings.py::test_matching_set_operations_match_manuscript</code>	set membership, not sufficiency of evidence
Definition 9	enabled iff τ is set; missing or unparseable dates flag rather than fail	<code>tests/test_formalism_bindings.py::test_temporal_review_rule_matches_manuscript</code>	a data-quality window, not a decay law
Definition 10	six clauses, first match wins	<code>tests/test_formalism_bindings.py::test_decision_clauses_fire_in_the_stated_order</code>	clause order, not moral order

Proposition	Statement essence	Verifying test	Boundary
Proposition 1	one finding per registry aspiration; counts partition $n = 9$	<code>tests/test_progress.py::test_counts_summary</code> , <code>tests/test_formalism_bindings.py::test_totality_count_matches_manuscript</code>	report shape, not coverage of a life
Proposition 2	a declared counter-signal forces <code>DRIFTING</code> over every later clause	<code>tests/test_progress.py::test_drifting_survives_staleness</code>	flags a recorded signal, not a failing person

Proposition	Statement essence	Verifying test	Boundary
Proposition 3	TOWARD iff all markers observed, none countered, currentness auditable	tests/test_progress.py::test_observed_and_unmet_fields_are_populated, tests/test_progress.py::test_fresh_observation_stays_toward, tests/test_figures.py::test_completeness_panel_gives_no_partial_credit	a reading of one record, not an accreditation
Proposition 4	temporal uncertainty yields INQUIRY, never DRIFTING	tests/test_progress.py::test_stale_observation_reverts_toward_to_inquiry, tests/test_progress.py::test_undated_entry_cannot_certify_currentness	age reopens a question, never condemns
Proposition 5	NOT_OBSERVED arises solely from no admitted entry	tests/test_progress.py::test_unknown_aspiration_id_is_noted_not_fatal, tests/test_progress.py::test_empty_entries_yield_all_not_observed	absence of signal is not evidence of drift
Proposition 6	first valid entry per id stands; identical calls reproduce identical reports	tests/test_progress.py::test_duplicate_entries_first_wins_and_is_noted, tests/test_formalism_bindings.py::test_intake_first_wins_and_replay_determinism_match_manuscript	determinism of the reading, not truth of the records
Proposition 7	screening is a pure function of (E, ids(R)); malformed input is set aside deterministically; never raises	tests/test_formalism_bindings.py::test_intake_screening_never_raises_on_hostile_input, tests/test_formalism_bindings.py::test_intake_screening_is_deterministic_on_all_input	determinism of the screening function, not truth of records
Proposition 8	staleness is strict: TOWARD at age τ , stale from $\tau + 1$	tests/test_progress.py::test_observation_exactly_at_staleness_boundary_stays_fresh, tests/test_progress.py::test_observation_one_day_past_staleness_boundary_is_stale, tests/test_formalism_bindings.py::test_currentness_boundary_constants_match_manuscript	the threshold is a review-cadence choice, not a decay law
Proposition 9	sweep delegates to progress_report and computes no status; codomain is TOWARD/INQUIRY only	tests/test_analysis.py::test_sweep_module_construction_no_status_of_its_own, tests/test_analysis.py::test_sweep_never_produces_drifting_or_not_observed, tests/test_formalism_bindings.py::test_sweep_codomain_matches_manuscript	characterizes the instrument, not any observed practice

Proposition	Statement essence	Verifying test	Boundary
Proposition 10	every aspiration's reading flips at the same observation age	<code>tests/test_figures.py::test_lattice_cells_are_executed_evaluator_readings, tests/test_figures.py::test_lattice_reports_a_uniform_flip_age, tests/test_figures.py::test_lattice_deviation_count_detects_a_planted_outlier</code>	uniformity of the code path, not of any practice
Proposition 11	registry digest is permutation-invariant, 64 lowercase hex characters	<code>tests/test_serialization.py::test_canonical_registry_is_deterministic_json, tests/test_golden_line.py::test_registry_is_unique_and_order_independent, tests/test_formalism_bindings.py::test_digest_order_independence_matches_manuscript</code>	a drift-review handle, no attestation semantics
Proposition 12	every invariant passes on the real registry and rejects a planted-bad one	<code>tests/test_invariants.py::test_battery_passes_on_real_registry, tests/test_invariants.py::test_registry_sound_false_on_any_planted_bad</code>	detection proof concerns the checks, not registry merit

The envelope section declares its blocks after the propositions above, so its rows sit in their own table, in the same document order:

Envelope block	Statement essence	Verifying test	Boundary
Definition 11	ten fields, <code>native_status</code> the complete ordered per-aspiration pairs	<code>tests/test_formalism_bindings.py::test_report_envelope_tuple_matches_the_dataclass</code>	a data contract, not a summary and not a score
Proposition 13	the envelope agrees with its report field for field; any post-export edit is visible	<code>tests/test_formalism_bindings.py::test_envelope_pointer_matches_manuscript, tests/test_report_envelope.py::test_envelope_matches_report_verifies_an_archived_pair</code>	archival agreement, not truth of the report

The bindings are themselves code behavior: they show which claims the suite would catch, not that the registry's aspirations are wise or well served.

5.6 Scholarship: an aspiration is a direction, not a score

Golden Line’s four founding aspirations are not inventions. Each translates a larger argument into the narrow idiom of a versioned registry: attention before production, usefulness beyond the author, repairable systems, and technical work answerable to human flourishing. The translation is deliberately incomplete. Golden Line borrows questions and warnings from these traditions; it does not claim to reconcile them, turn them into a moral canon, or use their authority to certify a project.

5.6.1 A translation, not a synthesis

The scholarship contributes constraints on how to hold a direction. It does not provide a universal scoring function:

Lineage	What it contributes here	What Golden Line refuses to import
Aristotle and MacIntyre	Work becomes intelligible through ends, practical judgment, and goods internal to a practice [Aristotle, 350 BCE, MacIntyre, 1981].	A fixed rule that settles every particular case.
Sen, Nussbaum, Robeyns, and Jonas	Capability is answerable to real human possibilities and to future life, not valuable merely because it is power [Sen, 1980, 1999, Nussbaum, 2011, Jonas, 1984]. Which capabilities matter is settled by public reasoning inside an open framework, not fixed by a theorist [Sen, 2004, Robeyns, 2017].	The instrument’s author deciding what flourishing means for everyone affected, and any claim that a marker is a functioning.
Schön, Polanyi, Sennett, and Jackson	Skilled work is situated, partly tacit, reflective, and sustained through maintenance and repair [Schön, 1983, Polanyi, 1966, Sennett, 2008, Jackson, 2014].	Expert opacity, heroics, or a productivist bias that treats repair as secondary.
Ostrom and Winner	Shared goods need governance, while technical arrangements can embody power [Ostrom, 1990a, Winner, 1980].	Treating “the commons” as frictionless openness or technology as socially neutral.
Campbell, Goodhart, Strathern, Merton, Espeland and Sauder, Manheim and Garrabrant, and Simon	Proxy targets distort conduct through several distinct mechanisms, and bounded agents need judgment rather than aggregate maximization [Campbell, 1979, Goodhart, 1975, Strathern, 1997a, Merton, 1948, Espeland and Sauder, 2007, Manheim and Garrabrant, 2018, Simon, 1956].	A numeric Golden Line grade that would recreate the problem it names.

This is the relevant sense in which Golden Line is scholarly: its registry is a small design intervention informed by disagreements among practical philosophy, capability theory, sociology of evaluation, studies of work, and technology critique. None of those sources validates the registry as a universal theory of value. They help specify what the instrument must ask, what it must not pretend to know, and where it can be gamed.

The groupings above are not a claim of agreement. Aristotle and MacIntyre offer accounts of ends, practices, and judgment; Sen’s comparative capability approach does not collapse into Nussbaum’s list-and-threshold proposal, and Sen has said why he declines to supply a fixed list [Sen, 2004]; Jonas extends responsibility toward future life. Ostrom studies institutional arrangements for common-pool resources, not a universal recipe for openness. Winner makes a political argument about artifacts; Jackson gives an interpretive account of repair in a broken world. Goodhart’s original argument concerns monetary-policy targets, not a general law of human behavior. Golden Line cites these works as distinct design resources, not as a single theoretical foundation or empirical validation.

The evidentiary status matters. Most of this section is normative or interpretive scholarship: it clarifies concepts and exposes failure modes. Two citations carry empirical weight of different kinds — Ostrom’s comparative institutional research and Espeland and Sauder’s field study of law-school rankings — while the remaining measurement literature is historical, methodological, and sociological. None of the cited work tests Golden Line’s registry, evaluator, or visual language. The package’s own tests establish implementation properties such as determinism and precedence; they do not establish that an aspiration is true, that a marker is socially sufficient, or that a directional reading predicts an outcome.

5.6.2 The shape of an end

The oldest commitment is that an activity has a point beyond its own motion. Aristotle’s ethics is organized around the *telos*, the end that makes a practice intelligible, and around *phronesis*, practical wisdom that reads a particular situation rather than applying a fixed rule [Aristotle, 350 BCE]. Golden Line translates that distinction into a horizon and a human reading of the record. A horizon says what the work is trying to serve; it does not predetermine what a good next decision must look like.

MacIntyre sharpens the point for modern practice. The goods *internal* to a craft, the excellences reachable only by doing the work well, differ from the external rewards the same work can earn; a practice is weakened when the latter replace the former [MacIntyre, 1981]. The aspiration *make work useful beyond its author* therefore names more than distribution. It asks whether the work has acquired an excellence that can travel without being severed from the practice that made it trustworthy. The aspiration *prefer systems that can be repaired* makes a parallel demand: the work must remain open to correction rather than preserve the appearance of finishedness.

5.6.3 Capability and flourishing

Keep technical work answerable to human flourishing draws on the capability approach. Sen’s Tanner Lecture put the prior question first — equality of *what?* — and argued that neither utility nor a bundle of primary goods answers it, because what matters is what a person is actually able to be and to do [Sen, 1980]. *Development as Freedom* carries that into policy: development is judged by real freedoms rather than by the accumulation of means [Sen, 1999]. Nussbaum develops a threshold account of central human capabilities, making the question concrete without reducing dignity to one aggregate index [Nussbaum, 2011]. Sen declines to fix such a list, holding that which capabilities matter depends on the purpose of the assessment and has to be settled by public reasoning rather than by the theorist [Sen, 2004]. Jonas extends the horizon: modern technology can alter distant and future life, so responsibility must include people who are not present at the moment of design [Jonas, 1984].

That disagreement about lists is the part of the literature Golden Line uses most directly, because the registry *is* a short list. Sen’s objection applies to it without modification: nine entries chosen by one author are a starting point, not a settled account of what matters. So the registry is versioned and digest-identified rather than canonical, and revising it is an ordinary operation rather than an admission of error.

Robeyns marks the limit that matters most here. The capability approach is an open framework rather than a finished theory, and it supplies no evaluation metric by itself; the theories built inside it are what make the further choices about which capabilities count and how they are weighed [Robeyns, 2017]. Golden Line borrows the framework’s question and none of those choices, and the transfer stops well short of the theory: a marker is not a functioning, an aspiration is not a capability, and a horizon report says nothing about any person’s real freedoms. What the evaluator can do is make “capability treated as its own justification” visible as a counter-signal. It cannot determine whose flourishing is at stake, resolve a conflict among affected people, or convert a local record into a public welfare claim. The code records a direction and its derivation; it does not authorize the person who filed the record to speak for everyone downstream.

5.6.4 Attention, craft, and transfer

Let attention precede production is a claim about the sequence of judgment, not a celebration of slowness for its own sake. Schön’s account of the reflective practitioner shows why competent work cannot be reduced to applying a formula: practitioners often discover and revise the problem while acting, through reflection-in-action [Schön, 1983]. Golden Line’s “next decision” horizon is small enough to preserve that situated judgment. It asks for a visible pause before the next output, not for a ritual of documentation.

Polanyi explains why transfer is difficult: experts know more than they can straightforwardly tell [Polanyi, 1966]. Sennett’s craftsman gives that difficulty a material and ethical form: quality grows through repeated attention to materials, error, and the desire to do a job well for its own sake [Sennett, 2008]. Jackson’s “broken world” account pushes the argument further. Maintenance and repair are not merely costs after the productive act; they are sites where knowledge, creativity, power, and care become visible [Jackson, 2014].

These sources explain why the registry uses both Markers and Counter-signals. The markers declared for *make work useful beyond its author* are “handoff used” and “reader question answered”: both test transfer by its effect on another person, without assuming that every piece of tacit knowledge can be compressed into a checklist. The counter-signal declared for *prefer systems that can be repaired* is “defect hidden to preserve appearance”, which tests whether repair has been made ordinary. The signal is not a verdict on character; it is a prompt to inspect the conditions under which knowledge travels and failure can be discussed.

Confucian teaching adds the relational dimension: learning is cultivated through conduct with others, not detached from the practice of living together [Confucius, 500 BCE]. Frankl’s account of meaning explains why a direction can remain orienting through difficulty without becoming a guarantee of success [Frankl, 1959]. Ibn Khaldun supplies a political caution: purposes persist when institutions and shared cohesion carry them beyond an individual’s intention [Khaldun, 1377]. Golden Line therefore treats usefulness, teachability, and durable understanding as social properties, not merely private virtues.

5.6.5 Commons, power, and return

The aspiration to return improvements to the commons needs more precision than a generic call for openness. Ostrom’s work on common-pool resources shows that shared goods are sustained by situated institutions: boundaries, rules, monitoring, graduated response, and the capacity of participants to govern themselves [Ostrom, 1990a]. The lesson for Golden Line is not that every artifact should be released without conditions. It is that “returned to the commons” should invite questions about who can use, inspect, adapt, maintain, and contest an improvement, and about which dependencies make that return possible.

Winner’s question, *Do Artifacts Have Politics?*, offers the complementary argument that technical things and systems can embody forms of power and authority [Winner, 1980]. Human flourishing cannot therefore be checked only at the level of a feature or repository. It also requires attention to the institutional arrangement in which the feature operates: who sets the terms, who bears the risk, whose labor remains invisible, and who can refuse. Golden Line does not answer those questions for the affected community. It keeps the questions from disappearing behind the language of capability or contribution.

5.6.6 Why it must not become a score

The literature on evaluation supplies Golden Line’s strongest negative design constraint, and it is older and more specific than the slogan it usually gets compressed into. Campbell stated the mechanism for social indicators: the more a quantitative indicator is used for social decision-making, the more it is subject to corruption pressures, and the more apt it is to distort the process it was meant to monitor [Campbell, 1979]. Goodhart’s 1975 paper addressed problems of monetary management and was not a universal moral law [Goodhart, 1975]; Strathern’s study of audit culture is where the compressed “measure becomes a target” phrasing entered wide circulation, along with the finding that target-driven measurement can displace the performance it was supposed to represent [Strathern, 1997a].

Manheim and Garraabrant separate the failure into four mechanisms — regressional, extremal, causal, and adversarial — and note Campbell’s law arguably has scholarly precedence over the Goodhart formulations [Manheim and Garraabrant, 2018]. The taxonomy shows exactly how far Golden Line’s design helps. Emitting no aggregate removes one target: there is no single Golden Line number for selection pressure to push to an extreme. It does not remove the proxies. Each marker is itself a proxy, and an observer working for “marker present” rather than for the direction reproduces the regressional case one token at a time. Surfacing Counter-signals before Markers and reading missing evidence as inquiry keeps thin evidence from being rounded up. Nothing touches the adversarial case: an observer who wants a TOWARD can file the tokens that produce one, because the evaluator checks tokens against the registry and never the record against the world.

Espeland and Sauder give the empirical version of the same worry. Studying United States law-school rankings, they identify two mechanisms of reactivity — self-fulfilling prophecy and commensuration — by which a public measure changes the conduct it claims to describe [Espeland and Sauder, 2007]. Merton’s earlier account supplies the first in general form: a definition of a situation can evoke behavior that makes the definition come true [Merton, 1948]. Golden Line blocks commensuration by construction, because it emits no comparable number and never aggregates the nine entries into a grade. It does not escape the first. Its scope is also narrower than theirs in a way worth stating: Espeland and Sauder study a public ranking of organizations, while a horizon report is a local record with no audience beyond the people who filed it. Whether a private directional record produces reactivity at all is a question their study does not answer.

Simon supplies the constructive counterpoint. Under bounded rationality, agents work with limited information in structured environments, and the practical task is satisficing — deciding what is good enough for the situation at hand — rather than maximizing an abstract proxy [Simon, 1956]. Golden Line makes that restraint executable. It emits TOWARD, INQUIRY, DRIFTING, or NOT_OBSERVED; it surfaces counter-signals before markers; it treats missing evidence as inquiry; and it refuses to aggregate the nine entries into a single grade. A status is a directional reading of one admitted record, never a measure of a person, a project, or a whole moral life.

5.6.7 From theory to instrument

The translation can be stated compactly:

- *Telos* becomes a stated horizon, while *phronesis* remains with the reader.
- Capability theory becomes a question about affected lives, and a reason to keep the list of aspirations short, versioned, and open to revision by the people the work touches — not a license for capability to justify itself.
- Reflective practice, tacit knowing, craft, and repair become observable markers and counter-signals about how work is done and handed on.
- Commons scholarship becomes a demand to name governance, reciprocity, and maintenance rather than celebrate openness abstractly.
- Technology critique becomes a reminder that the surrounding institution is part of the object of attention.
- The sociology of targets becomes a hard prohibition on a Golden Line score, and an admission that no arrangement of clauses stops an observer who wants a particular reading from filing one.

The result is intentionally modest. Golden Line is a local instrument for keeping a purpose visible while work changes. Its scholarship makes that modesty more intelligent: a direction is useful only when it remains revisable, situated, answerable to people beyond its author, and honest about what the record cannot show.

5.7 Evidence boundary and artifact chain

Golden Line does not begin with a score. It begins with a separation of things that are easy to conflate: what the registry declares, what an observer records, what the evaluator can derive, and what a generated artifact can prove about its own provenance (formalized in Definition 9 and Proposition 8).

5.7.1 Hard constraints and soft choices

The hard constraints are epistemic and semantic. An observation cannot create a signal that the registry did not declare. Missing evidence cannot become evidence of absence. A status must be reproducible from its inputs. Malformed temporal metadata cannot establish currentness. An aspiration cannot authorize what the Red Line refuses or resolve what the White Line leaves absent.

The soft choices are the nine-entry registry, exact token matching, the first-valid-entry rule for duplicates, the optional staleness window, and the names of the four statuses. They are versioned design decisions, not universal truths. A future registry may revise them, but it must revise the digest, figures, tests, and manuscript together.

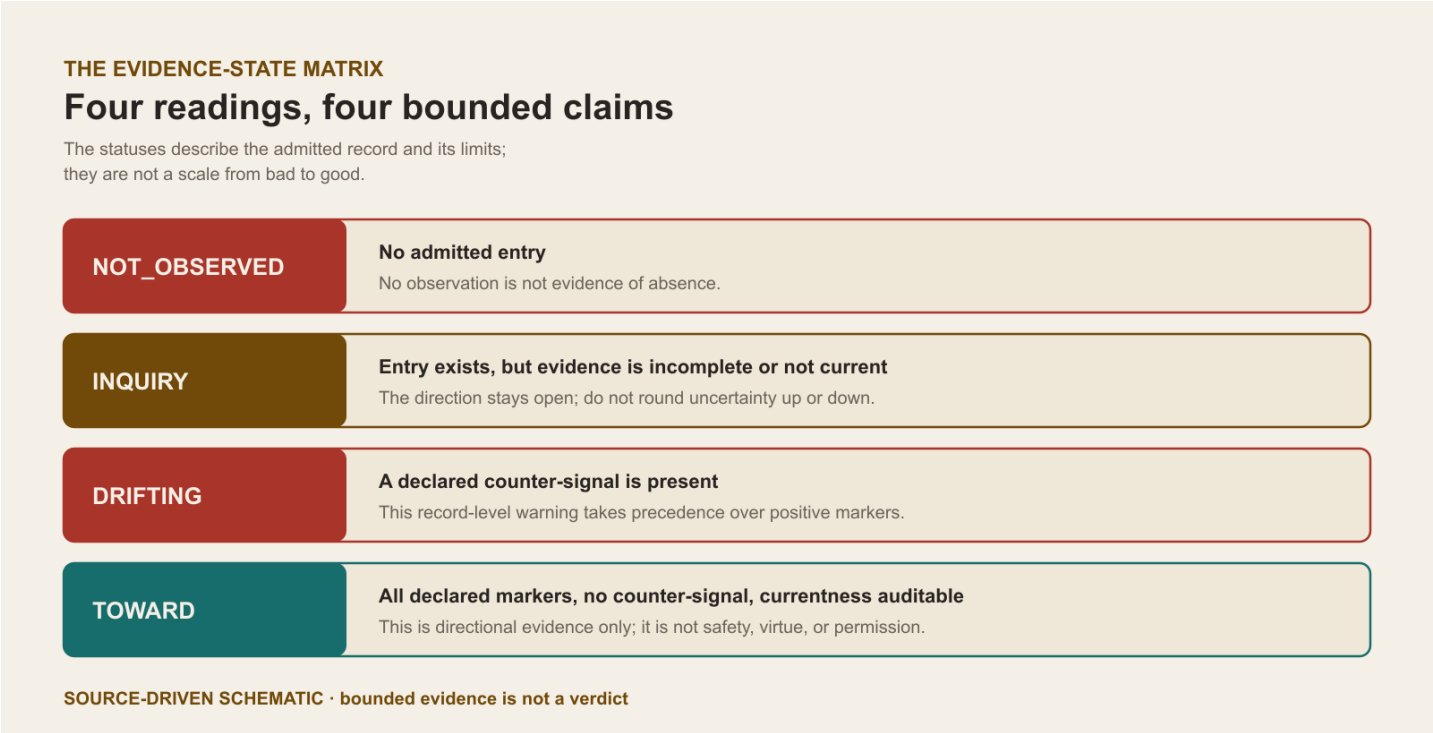


Figure 50: The four HorizonStatus readings mapped to bounded evidence conditions: no admitted entry; incomplete, stale, or unauditable evidence; a declared counter-signal; and complete current/auditable evidence. The matrix is not a ranking.

5.7.2 Statuses as bounded claims

NOT_OBSERVED means that no valid entry was admitted. INQUIRY means that an entry exists but does not support a complete current positive reading. DRIFTING means that a declared counter-signal was recorded and takes precedence over markers. TOWARD means that every declared marker was observed, no declared counter-signal was recorded, and the observation was current when temporal review was enabled. None of these readings is a grade, a diagnosis, a safety claim, or a permission mechanism. fig. 50

The full reasons trail remains useful to a human reader, but structured fields (observed, unmet, countered, ignored tokens, note, date, and temporal flags) make the derivation directly inspectable. A consumer need not turn explanatory prose back into data before checking what happened.

Every reading returns the same record; which of its fields carry content is decided by what the matching stage found. fig. 51 runs one entry per evidence condition and reports, field by field, what came back. Three fields are carried under every condition — the aspiration id, the status, and the reasons trail — so no reading arrives unexplained, and the remaining nine are carried exactly when the record supplied something for them. An empty field is an absent observation and nothing more.

5.7.3 Source to publication

The local release chain is:

THE SHAPE OF A FINDING

12 fields × 6 evidence conditions

Every reading returns the same record; the matching stage decides which fields carry content.
One executed progress_report call per column (review date 2026-07-18, stale_after_days = 90).

SOURCE-DRIVEN SCHEMATIC · a derivation trail, never a justification of what was recorded

	NOT_OBSERVED no admitted entry	DRIFTING counter-signal, one undeclared	INQUIRY markers incomplete, one undeclared	INQUIRY complete but stale	INQUIRY complete but undated	TOWARD complete and current
FINDING FIELD						
aspiration_id	carried	carried	carried	carried	carried	carried
status	carried	carried	carried	carried	carried	carried
reasons	carried	carried	carried	carried	carried	carried
observed	empty	carried	carried	carried	carried	carried
unmet	empty	empty	carried	empty	empty	empty
countered	empty	carried	empty	empty	empty	empty
ignored_markers	empty	carried	carried	empty	empty	empty
ignored_counter_signals	empty	carried	carried	empty	empty	empty
note	empty	carried	carried	carried	carried	carried
observed_on	empty	carried	carried	carried	empty	carried
stale	empty	empty	empty	carried	empty	empty
date_issue	empty	empty	empty	empty	carried	empty

3 fields carried under every condition: aspiration_id, status, reasons — no reading arrives unexplained.

Filled cells are carried and outlined cells are empty, and every cell prints which it is.

A carried flag is a recorded condition, not a good result: stale and date_issue are carried exactly when currentness could not be established.

An empty field is an absent observation: the record says what was seen, never what is true.

SOURCE-DRIVEN SCHEMATIC · field contents describe one synthetic record, never a person or a project

Figure 51: The twelve structured fields of a finding against six evidence conditions, one executed progress_report call per column. Filled cells are fields the returned record carries and outlined cells are empty, and every cell prints which it is. A carried temporal flag records a currentness problem, not a good result; an empty field records an absent observation and never asserts that what it names is false.

registry + status enum
↓
progress_report + invariant battery
↓
deterministic SVG, PNG, and JSON registries
↓
check_artifacts.py
↓
template PDF/HTML render and release audit

`check_artifacts.py` verifies the registry version and digest, the expected figure inventory, SVG/PNG pairs, generator identity, and figure-label citations in the manuscript. It is a consistency gate, not independent truth verification. A rendered artifact still needs visual and publication validation in the sibling template checkout.

5.8 The Golden Line aspirations

The registry opens with four founding aspirations. They are broad enough to travel across research and engineering, but each is paired with a concrete horizon so that it can be tested against a record rather than merely admired.

- 1. **Let attention precede production.** Make enough room to see what the work is actually doing at the next decision. *Horizon:* the next decision. *Counter-signal:* automatic output without review.
- 2. **Make work useful beyond its author.** Leave knowledge, tools, and explanations that another person can carry to a collaborator or public reader. *Horizon:* a collaborator or public reader. *Counter-signal:* private cleverness without transfer.
- 3. **Prefer systems that can be repaired.** Expose failure early and make correction ordinary in the revision cycle. *Horizon:* the revision cycle. *Counter-signal:* a defect hidden to preserve appearance.
- 4. **Keep technical work answerable to human flourishing.** Treat capability as a means whose value depends on the lives around it over a long horizon. *Horizon:* the long horizon. *Counter-signal:* capability treated as its own justification.

Five further entries extend the founding four in the versioned registry, filling out the long-horizon picture: **durable understanding** that outlasts the tool, **teachable craft** that can be handed to the next learner, **honest uncertainty** kept visible at the same prominence as the claim, at least one **unhurried question** measured in years rather than sprints, and improvements **returned to the commons** they came from. Each follows the same shape: a horizon, observable markers, and counter-signals (Definition 1), with the same reading rule (Definition 10). Together the nine entries and their signal structure are laid out in fig. 53.

The order is not a ranking. The aspirations can conflict: usefulness beyond the author can pull against protecting an unhurried question; returning everything to the commons can pull against an obligation to a specific collaborator, and an honest record may need to show that conflict rather than resolve it. The companion White Line is the proper place to record what the registry cannot see or what should not be claimed.

The founding four are best read as a loop of correction, drawn in fig. 52. Attention protects perception before output; usefulness tests whether what was learned can travel; repair keeps failure from becoming identity; answerability to human flourishing asks what the capability is for. No point is a maturity level. A project can move around this loop, lose one of its conditions, or find that two aspirations pull against each other. The useful question is therefore not *How high are we?* but *What does this record make visible, and what remains unasked?*

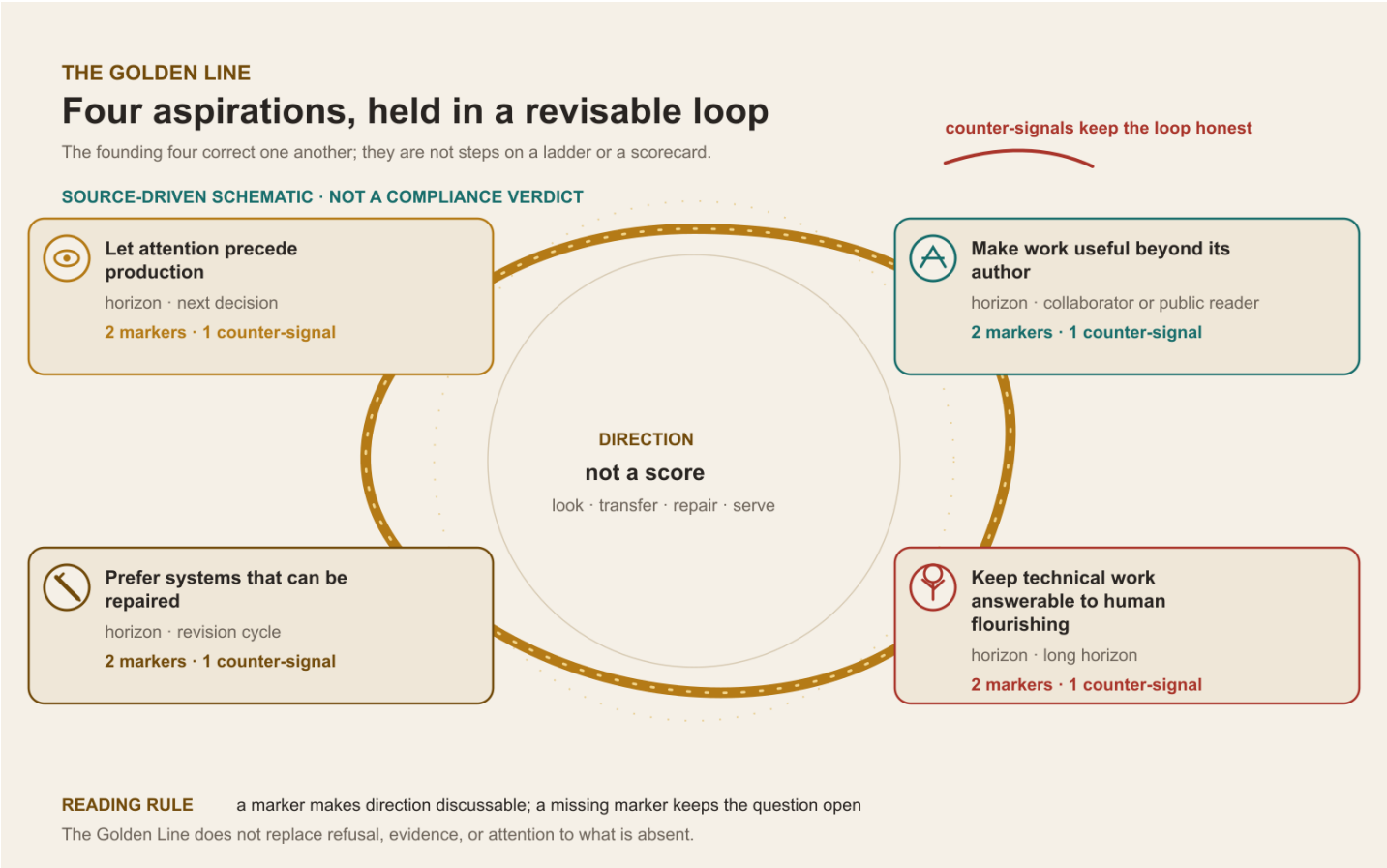


Figure 52: The four founding aspirations held in a revisable loop. Titles, horizons, and marker/counter-signal counts are source-derived; the connecting loop and icons are interpretive. The loop is direction, not a score or ranking.

THE ASPIRATION REGISTRY

9 versioned aspirations, drawn from source

Each row shows a horizon and the declared markers and counter-signals that make its readings reachable. No cell is a grade.

FOUNDING · 4 entries

Let attention precede production

horizon: next decision

2 markers

1 counter-signal

Make work useful beyond its author

horizon: collaborator or public reader

2 markers

1 counter-signal

Prefer systems that can be repaired

horizon: revision cycle

2 markers

1 counter-signal

Keep technical work answerable to human flourishing

horizon: long horizon

2 markers

1 counter-signal

FURTHER · 5 entries

Prefer understanding that outlasts the tool

horizon: tool turnover

2 markers

1 counter-signal

Work so the craft can be taught

horizon: next learner

2 markers

1 counter-signal

Keep uncertainty visible in finished work

horizon: public claim

2 markers

1 counter-signal

Hold at least one unhurried question

horizon: multi-year inquiry

2 markers

1 counter-signal

Return improvements to the commons

horizon: shared pool

2 markers

1 counter-signal

SOURCE-DRIVEN SCHEMATIC · marker and counter-signal counts describe reachability, not performance

Figure 53: The full nine-entry aspiration registry: four founding aspirations and five further entries, drawn from the versioned source. Each row shows its horizon and declared marker/counter-signal counts; the layout is a taxonomy for review, not a performance scale.

5.8.1 The signal vocabulary in aggregate

Read as a whole, the registry declares a deliberately small vocabulary: 18 markers and 9 counter-signals across the nine aspirations — two markers and one counter-signal per entry, with all 18 marker tokens and all 9 counter-signal tokens distinct across the registry. The `signal_inventory` helper in the analysis layer derives these counts from the live source, and fig. 54 draws them one block per declared token. The distinctness matters: because no token is shared between aspirations, an observed marker can never accidentally support two directions at once. The blocks are vocabulary the evaluator can match, never observations and never points.

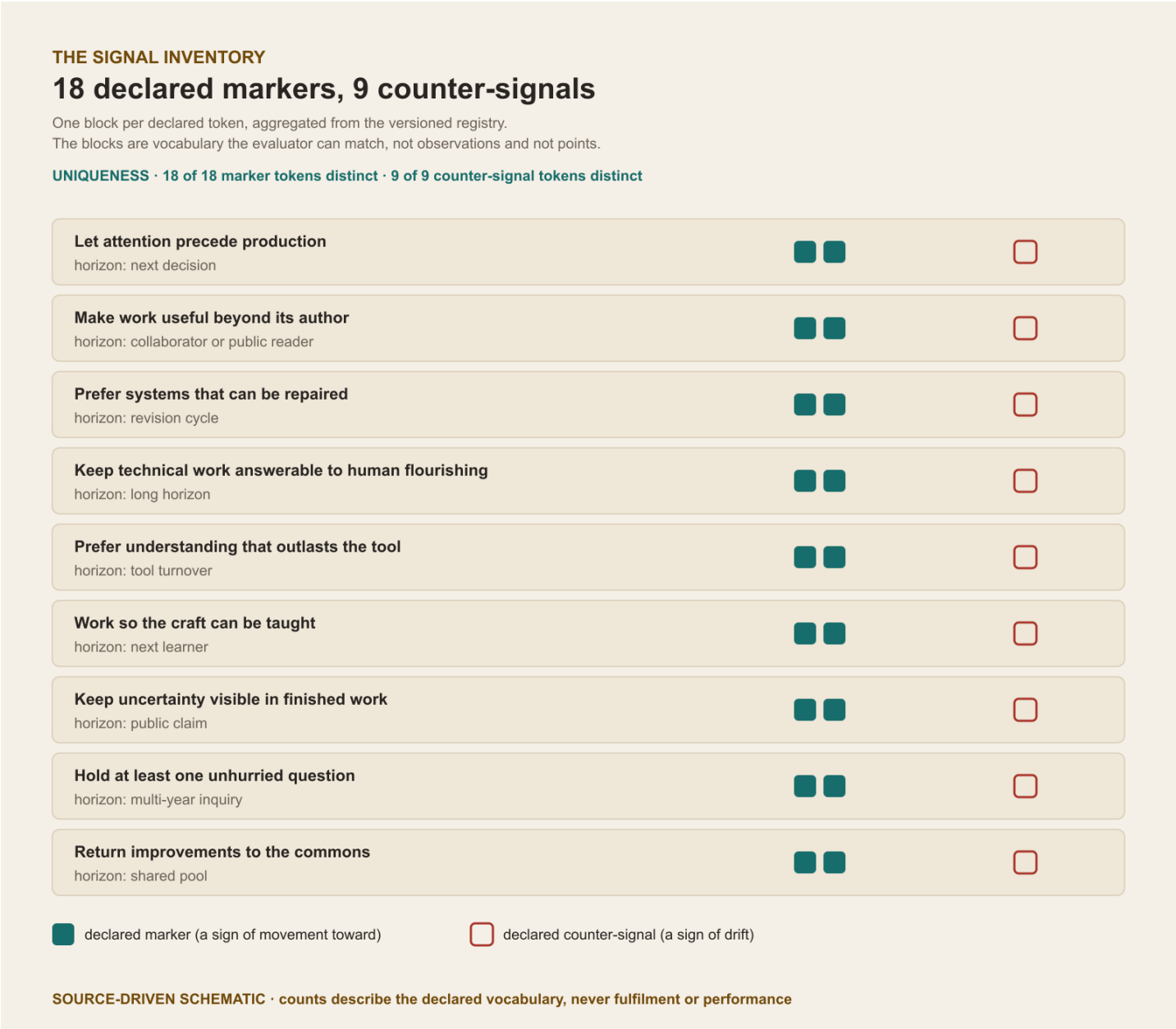


Figure 54: The aggregate declared-signal vocabulary of the registry, one unit block per token: filled blocks for markers, outlined blocks for counter-signals. The counts describe what the evaluator can match, never fulfilment or performance.

5.8.2 The reach of the nine horizons

The nine horizon phrases can also be grouped by *when* their direction becomes visible. The analysis layer declares four interpretive temporal-reach bands — immediate (1 aspiration, at the next decision), recurring cycle (2, at revision or tool turnover), at handoff (4, when the work reaches another person), and open-ended (2, over years) — and `horizon_distribution` places every registry entry into exactly one band, refusing loudly if a future registry adds a horizon the map does not classify. fig. 55 shows the grouping. The bands are a reading aid declared outside the registry contract: band order widens reach, and a wider horizon is not a higher rank.

THE HORIZON-BAND DISTRIBUTION

9 horizons, 4 reaches of visibility

An interpretive grouping declared in the analysis layer: each aspiration is placed by when its direction becomes visible. Band order widens reach; it does not rank importance.

SOURCE-DRIVEN SCHEMATIC · the bands are a reading aid, not registry contract and not a maturity ladder

IMMEDIATE

1 aspiration

Let attention precede production

"next decision"

RECURRING CYCLE

2 aspirations

Prefer systems that can be repaired

"revision cycle"

Prefer understanding that outlasts the tool

"tool turnover"

AT HANDOFF

4 aspirations

Make work useful beyond its author

"collaborator or public reader"

Work so the craft can be taught

"next learner"

Keep uncertainty visible in finished work

"public claim"

Return improvements to the commons

"shared pool"

OPEN-ENDED

2 aspirations

Keep technical work answerable to human flourishing

"long horizon"

Hold at least one unhurried question

"multi-year inquiry"

SOURCE-DRIVEN SCHEMATIC · a wider horizon is not a higher rank; every band is revisited in its own time

Figure 55: The nine aspiration horizons grouped into four temporal-reach bands — immediate, recurring cycle, at handoff, and open-ended. An interpretive reading aid, not a maturity ladder and not part of the registry contract.

The aspirations are exercised in the worked examples and the batch reading.

5.9 Worked records

The model is intentionally small enough to inspect at the command line. A single horizon entry against one aspiration produces a full report over the whole registry:

```
from golden_line import HorizonEntry, progress_report

report = progress_report([
    HorizonEntry(
        aspiration_id="repairable-systems",
        observed_markers=frozenset({"failure named", "revision attempted"}),
        note="The failure was recorded before the release note was written.",
    )
])
```

That entry yields **TOWARD** for **repairable-systems** and **NOT_OBSERVED** for the other eight registry items, because no horizon note was recorded for them. The readings move exactly as the decision rule prescribes:

- If the same entry also records the counter-signal **defect hidden to preserve appearance**, the finding becomes **DRIFTING**. Counter-signal precedence (**Proposition 2**) means this holds even if both markers are still present; a positive signal cannot launder a recorded drift.
- If only one of the two declared markers is present, the finding becomes **INQUIRY** with an **unmet** list naming the marker still outstanding, so the direction is recorded as open rather than reached — no partial credit (**Proposition 3**).
- If both markers are present but the observation carries an old, future, or missing/unparseable **observed_on** date and the caller passes **stale_after_days**, the finding reverts to **INQUIRY**: temporal uncertainty reopens the question without inventing drift (**Proposition 4**).
- An entry naming an aspiration that is not in the registry is set aside during intake (**Definition 7**) and reported in the intake notes; it never crashes the report and never silently disappears.
- A malformed record, such as one whose marker field is **None**, is also set aside and named in the intake notes (**Definition 7**). If a dated positive record is supplied with temporal review enabled but its date is not ISO-parseable, the finding is **INQUIRY** with **date_issue=True** (**Proposition 4**): malformed time metadata cannot certify currentness.

The first of those bullets is the one most worth distrusting, because it is the rule a reader is most likely to assume has an exception. It does not. Running all nine aspirations through four evidence conditions — complete markers alone, complete markers with one declared counter-signal, a counter-signal with no markers, and a counter-signal on an observation four hundred days old — returns **DRIFTING** in every cell where a counter-signal is present, including all nine cells where every declared marker is also present. fig. 57 is that run: each cell in it is a **progress_report** return value rather than a box drawn to illustrate one.

These distinctions keep a positive signal from laundering a counter-signal (**Proposition 2**), and keep the absence of evidence from masquerading as either success or failure (**Proposition 5**). The path from an entry to its reading is shown in fig. 46, and the three evaluator stages an entry passes through — including the set-aside branch that carries malformed, unknown-id, and duplicate records into the intake notes — are shown in fig. 56, while the bounded meaning of all four readings is summarized in fig. 50.

A single entry shows the decision rule in isolation. The next section runs a whole batch — positive, drifting, partial, malformed, and duplicate entries at once — through the same evaluator and the descriptive analysis layer, so the gap between what was submitted and what was read becomes fully accountable.

THE STAGED EVALUATION PIPELINE

progress_report runs three stages

Malformed or unexpected input is recorded, never raised;
every finding carries a full reasons trail.

1 · INTAKE SCREENING

One entry per known id (first wins). Malformed records, unknown ids, and duplicates are set aside into intake notes.

2 · MATCHING

Observed markers and counter-signals are intersected with the declared sets. Undeclared tokens are ignored and noted.

3 · DECISION

The matched sets, with optional temporal review, select one of the four directional readings.

OUTCOME: 9 findings — one per aspiration, each a direction, never a grade.

SOURCE-DRIVEN SCHEMATIC · set-aside notes are actual progress_report intake notes · NOT A COMPLIANCE VERDICT

SET ASIDE · 3 kinds → intake_notes

notes quoted verbatim from the evaluator

MALFORMED RECORD

entry #1 was set aside: expected a HorizonEntry record

UNKNOWN ID

entry for unknown aspiration
'not-a-real-id' was set aside

DUPLICATE ENTRY

duplicate entry for
'attention-before-output' was set aside;
the first entry stands

set aside, named, and counted — never silent

UNDECLARED TOKENS

ignored and noted; they can never
determine a status

Figure 56: A batch of horizon entries passes through intake screening, signal matching, and decision. Records that fail intake — malformed, unknown-id, or duplicate — are set aside into visible intake notes (quoted verbatim from a real progress_report replay), undeclared tokens are ignored and noted, and the output is one bounded finding per registry aspiration. Findings are directional readings, never grades.

COUNTER-SIGNAL PRECEDENCE

9 of 9 aspirations still read DRIFTING

Each cell is one real progress_report call on a synthetic entry for that aspiration
(review date 2026-07-18, stale_after_days = 90).

SOURCE-DRIVEN SCHEMATIC · a replayed clause ordering, never a judgement of the work or its author

	complete markers, no counter-signal	complete markers, one counter-signal	no markers, one counter-signal	complete markers, one counter-signal, 400 d old
ASPIRATION				
attention-before-output	TOWARD	DRIFTING	DRIFTING	DRIFTING
useful-to-others	TOWARD	DRIFTING	DRIFTING	DRIFTING
repairable-systems	TOWARD	DRIFTING	DRIFTING	DRIFTING
wide-human-flourishing	TOWARD	DRIFTING	DRIFTING	DRIFTING
durable-understanding	TOWARD	DRIFTING	DRIFTING	DRIFTING
teachable-craft	TOWARD	DRIFTING	DRIFTING	DRIFTING
honest-uncertainty	TOWARD	DRIFTING	DRIFTING	DRIFTING
unhurried-questions	TOWARD	DRIFTING	DRIFTING	DRIFTING
commons-returned	TOWARD	DRIFTING	DRIFTING	DRIFTING

Complete markers plus one declared counter-signal returns DRIFTING for 9 of 9 aspirations; at an observation 400 days old it still does, for 9.

Staleness reopens a positive reading; it never erases a recorded drift, and drift never becomes a verdict on a person.

SOURCE-DRIVEN SCHEMATIC · filled cells are DRIFTING, outlined cells are not · NOT A COMPLIANCE VERDICT

Figure 57: Counter-signal precedence replayed rather than drawn: for each of the nine aspirations, four progress_report calls at review date 2026-07-18 with stale_after_days = 90. Complete markers with no counter-signal read TOWARD; adding one declared counter-signal returns DRIFTING even though every marker is still present, and it still returns DRIFTING when the same observation is stale. Filled cells are DRIFTING and outlined cells are not, so the panel reads without colour. The panel characterizes clause order; it does not grade any work or person.

5.10 Reading a batch: the descriptive layer at work

A single horizon entry is easy to read by eye. A real review is rarely a single entry: it is a batch of observations filed against several aspirations at once, some positive, some drifting, some partial, and a few malformed. The descriptive analysis layer exists to characterize such a batch without adding any claim the evaluator did not already make. This section runs one batch end to end so the helpers can be seen doing exactly — and only — what the formalism permits.

5.10.1 A worked batch

Consider six submitted horizon entries evaluated against the shipped nine-entry registry with `progress_report`:

```
from golden_line import HorizonEntry, progress_report, report_overview

entries = [
    HorizonEntry("attention-before-output",
        observed_markers=frozenset({"question revisited", "context named"})),
    HorizonEntry("repairable-systems",
        observed_markers=frozenset({"failure named", "revision attempted"}),
        counter_signals=frozenset({"defect hidden to preserve appearance"})),
    HorizonEntry("useful-to-others",
        observed_markers=frozenset({"handoff used"})),
    HorizonEntry("honest-uncertainty",
        observed_markers=frozenset({"limit stated beside claim",
                                     "confidence qualified in print",
                                     "extra token that is undeclared"})),
    HorizonEntry("not-a-real-id",
        observed_markers=frozenset({"whatever"})),
    HorizonEntry("attention-before-output",
        observed_markers=frozenset({"question revisited"})),
]
report = progress_report(entries)
overview = report_overview(report)
```

Six entries were submitted, but the report still contains exactly nine findings — one per registry aspiration — because totality (Proposition 1) does not depend on what was filed. `report.counts()` partitions those nine findings as **two TOWARD, one INQUIRY, one DRIFTING, and five NOT_OBSERVED**. The `report_overview` helper regroups the same findings by status and totals the intake so the batch can be described without re-parsing prose:

Reading	Count	Aspirations
TOWARD	2	attention-before-output, honest-uncertainty
INQUIRY	1	useful-to-others
DRIFTING	1	repairable-systems
NOT_OBSERVED	5	wide-human-flourishing, durable-understanding, teachable-craft, unhurried-questions, commons-returned

Each reading follows the decision rule of the formalism, and each illustrates one of its guarantees:

- **attention-before-output** recorded both of its declared markers and no counter-signal, so it reads TOWARD.
- **repairable-systems** recorded **both** declared markers *and* a declared counter-signal. Counter-signal precedence (Proposition 2) surfaces the drift first: the reading is DRIFTING, and the two present markers cannot launder it.
- **useful-to-others** recorded only one of its two declared markers, so the direction is held open as INQUIRY with the second marker still unmet — not rounded up and not condemned (Proposition 3).
- **honest-uncertainty** recorded both declared markers plus one token, "extra token that is undeclared", that the registry never declares. The undeclared token is ignored, counted once in the overview's `ignored_marker_total`, and changes nothing: because both declared markers were present and no counter-signal was recorded, the reading is TOWARD. An observer-supplied token can never smuggle in a status (Definition 8).
- The five aspirations with no admitted entry read NOT_OBSERVED. This is not evidence that those directions are absent or failing; it reports only that no valid record reached this report (Proposition 5).

5.10.2 What intake set aside

Two of the six submitted entries never became findings, and the overview makes the reason legible without any narrative: `intake_note_count` is **2**. The `report.intake_notes` are, verbatim:

- "entry for unknown aspiration 'not-a-real-id' was set aside"
- "duplicate entry for 'attention-before-output' was set aside; the first entry stands"

The unknown identifier was screened out during intake rather than crashing the report; the duplicate `attention-before-output` entry — a weaker record with only one marker — was set aside by the first-valid-entry rule, and the first, complete entry stands. Neither disappeared silently: both are named in the intake notes so the gap between six submissions and four incorporated readings is fully accounted for. For this batch the overview’s `ignored_marker_total` is **1**, its `ignored_counter_signal_total`, `stale_count`, and `date_issue_count` are all **0** (temporal review was left disabled here), so the entire delta between raw input and final readings is visible in five small integers.

The whole batch is drawn in fig. 58: the figure builder replays exactly this batch through `progress_report` and `report_overview`, so every count, grouping, and quoted intake note in the picture is the evaluator’s actual output for the code block above, not an illustration of it.

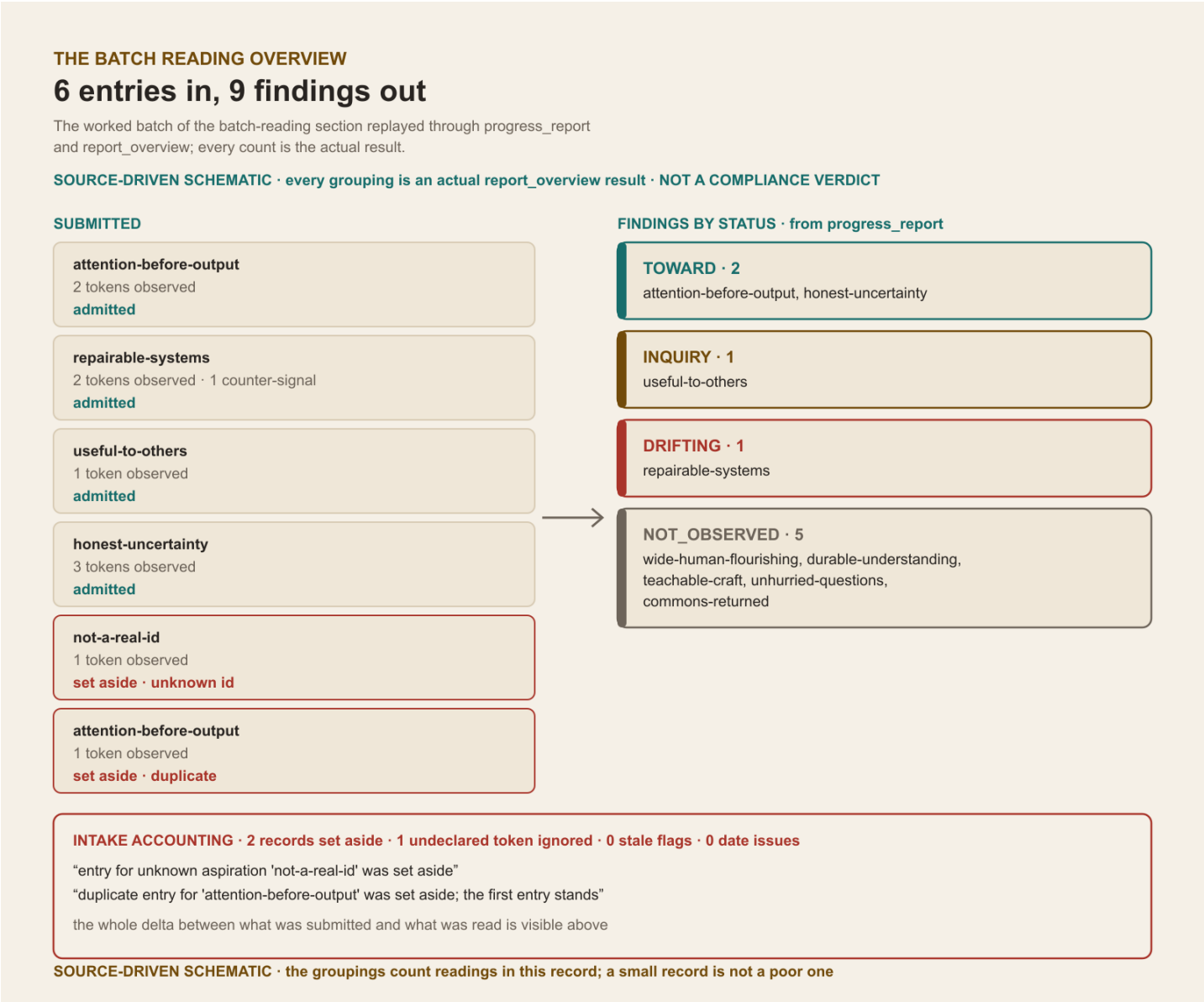


Figure 58: The 6-entry worked batch of this section replayed through the real evaluator: 2 TOWARD, 1 INQUIRY, 1 DRIFTING, and 5 NOT_OBSERVED across 9 findings — one per registry aspiration whether or not an entry was filed — with 2 intake set-asides and 1 ignored undeclared token quoted verbatim from the report. The groupings count readings in this record; they do not grade the work or the people behind it.

5.10.3 Vocabulary and reach as reading context

The overview describes what *this batch* produced; the signal inventory and horizon-band distribution describe the fixed vocabulary and temporal reach that any batch is read against. Because every one of the registry’s eighteen markers and nine counter-signals is a distinct token (fig. 54), the `ignored_marker_total` above is unambiguous: the ignored token matched no declared marker of any aspiration, not merely the one it was filed under. And because the four aspirations that drew readings here — attention, repair, usefulness, honest uncertainty — sit in three different temporal-reach bands (fig. 55), a single review batch routinely mixes an immediate-horizon reading with an at-handoff one. The bands are a reminder that these readings become visible on different clocks, not a schedule on which they should be expected to agree.

None of these summaries is a grade. `report_overview` adds no claim the findings did not already carry; it only counts them. A batch that reads two `TOWARD` and five `NOT_OBSERVED` is not “worse” than one that reads nine `TOWARD`. It is a smaller record.

5.11 Limits and safeguards

Aspiration language can become sentimental, coercive, or falsely universal, and a directional instrument can be misread as a verdict. Golden Line therefore carries six hard limits. Three of them have a mechanical core the code enforces, and the proposition that pins each one is named inline; the other three are commitments the code cannot hold for us, and saying so is part of keeping them.

- *Commitment.* A registry entry is a design choice, not a discovery of humanity’s single highest good. The nine aspirations are a starting set, versioned and revisable, not a closed canon.
- *Enforced (Proposition 3).* A TOWARD result says only that declared markers were observed, no counter-signal was recorded, and (when temporal review is enabled) currentness was auditable from a present, parseable, in-window date. Those conditions are necessary and sufficient in the code. That the result does not prove the work good, safe, lawful, or beneficial is a reading rule, not a check.
- *Enforced (Proposition 5).* A NOT_OBSERVED result arises solely from the absence of an admitted entry. It is not evidence that the aspiration is absent; submitted records may have been malformed, unknown, or duplicate, and the intake notes say which.
- *Commitment.* Counter-signals are local warnings about a record, not psychological diagnoses or public labels for a person or project. Nothing in the evaluator can stop a reader from using one that way.
- *Enforced (Definition 7, Proposition 4).* A malformed entry is visible as an intake set-aside, and a missing or malformed date in an enabled temporal review prevents a positive record from being read as current. These are data-quality safeguards, not proof that the underlying work failed.
- *Commitment.* The instrument cannot resolve conflicts among affected people; it can only make the chosen direction and its tradeoffs easier to discuss. This is an absence rather than a check, and no test can confirm it.

The sharpest risk is the one the scholarship section names: that a published aspiration hardens into a target and is gamed rather than served [Strathern, 1997a], or that stating the direction as a scorecard evokes the very behavior it was meant to describe [Merton, 1948]. The design pushes against this at every stage: no numeric score, no aggregate grade, counter-signals surfaced first, absence read as inquiry, and no positive currentness claim from malformed temporal data. What it does not touch is the adversarial case [Manheim and Garra-brant, 2018]: the evaluator matches the tokens an observer filed against the registry, so an observer who wants a TOWARD can file the tokens that produce one, and no clause ordering prevents it. The honest safeguard is to keep the readings directional and the responsibility human, in the spirit of Jonas’s long-horizon duty rather than a compliance ritual [Jonas, 1984].

5.11.1 Limits of the descriptive analysis layer

The descriptive helpers — `signal_inventory`, `horizon_distribution`, `temporal_currentness_sweep`, and `report_overview` — introduce their own, smaller risks of over-reading, and each is deliberately constrained so it cannot outrun the evaluator it sits beside.

- The inventory counts declared *vocabulary*, never fulfilment. That the registry declares eighteen markers and nine counter-signals says nothing about how many have ever been observed; a larger marker count is not a higher standard, only a longer list of things one could look for. Reading the inventory as a scoreboard would invert its purpose.
- The horizon bands are an interpretive reading aid declared in the analysis layer, not part of the registry contract. Their order widens temporal reach; it does not rank merit, and an “open-ended” horizon is not superior to an “immediate” one. The band map is intentionally brittle: an aspiration whose horizon it does not classify raises an error rather than being dropped, so a future registry change forces the map to be revisited by a human instead of silently mis-grouping an entry.
- The currentness sweep replays **one** fully-marked, counter-signal-free synthetic entry to expose where a reading loses currency. It is a probe of the evaluator’s temporal rule under a chosen `stale_after_days`, not a claim about any real observation, and the staleness boundary it draws is a data-quality threshold — a `stale` cell is an invitation to look again, never evidence that the underlying work failed. The lattice widens the probe to every registry entry, which rules out the boundary being an artifact of the one entry chosen; it does not make the entries any less synthetic.
- `report_overview` regroups and counts findings that already exist; it adds no claim the report did not carry and derives nothing from outside the report’s own structured fields. Its per-status tallies are a summary, and a batch that reads mostly NOT_OBSERVED is a small record, not a poor one.

Because all four helpers are pure, deterministic, and read-only, they cannot change what a reading means — but a reader can still misuse a number. The safeguard is the same one the whole instrument relies on: keep the summaries descriptive and the judgement human.

For security boundaries, use Red Line. For how the work was performed, use Black Line. For what is missing, withheld, unobserved, or ethically left unclaimed, use White Line. A horizon report must never be used to bypass those scopes, and a TOWARD on any Golden Line aspiration can never authorize what Red Line refuses.

5.12 Conclusion

Golden Line keeps a question alive: what is this work for, and what would movement toward that purpose look like at the next honest horizon? Its answer is not a universal doctrine. It is a versioned set of nine threads, each with a horizon, observable markers, and counter-signals, read by a staged evaluator that returns a direction and never a grade.

The formalism shows how modest the machine is: four directional readings ([Definition 4](#)), a decision rule in which a counter-signal outranks any marker ([Proposition 2](#)) and staleness or a malformed reviewed date reopens rather than condemns ([Proposition 4](#)), and seven structural invariants each kept honest by a planted-bad proof-of-detection test ([Proposition 12](#)). Six of the thirteen figures replay those rules rather than draw them: precedence ([Proposition 2](#)), marker completeness ([Proposition 3](#)), the currentness boundary ([Proposition 8](#)), and the shape of the returned record are all read off executed output ([Proposition 9](#)). source, figures, and manuscript citations remain joined.

The scholarship shows both the age and the stakes of the commitments (see [the scholarship section](#)): ends and practical wisdom, internal goods, capability and flourishing, reflective craft, repair, commons governance, the political character of technical systems, and the long-horizon duty that technological power now demands. It also shows why the instrument must stay anti-metric: a direction hardened into a target stops measuring what it named, and an open commons without governance can simply hide who carries the cost.

The instrument is strongest when paired with the restraint of its companions, each of which does a job it cannot. Golden Line adds a direction without turning aspiration into authority, a thread worth reaching toward, held in the open, and revised whenever the evidence or the values change.

The work is built for DAF's public research index at [docxology](#), machine-readable, cross-linked, and verification-logged, with its eventual public home at [docxology/golden_line](#). The aspirational thread is meant to be kept in plain sight.

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

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

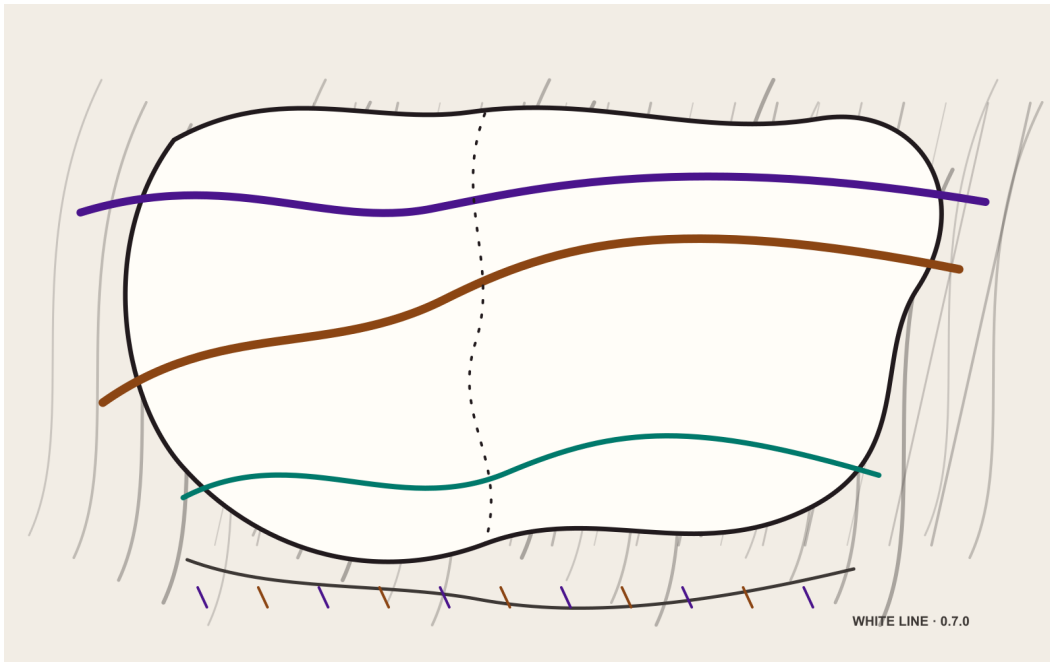


Figure 59: Cover art for White Line: A Typed Ledger for the Edge of the Claim

Reproduced unchanged in substance from its own source at version 0.7.0. It answers one question: What is absent or unknowable?

6.1 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.

6.2 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.

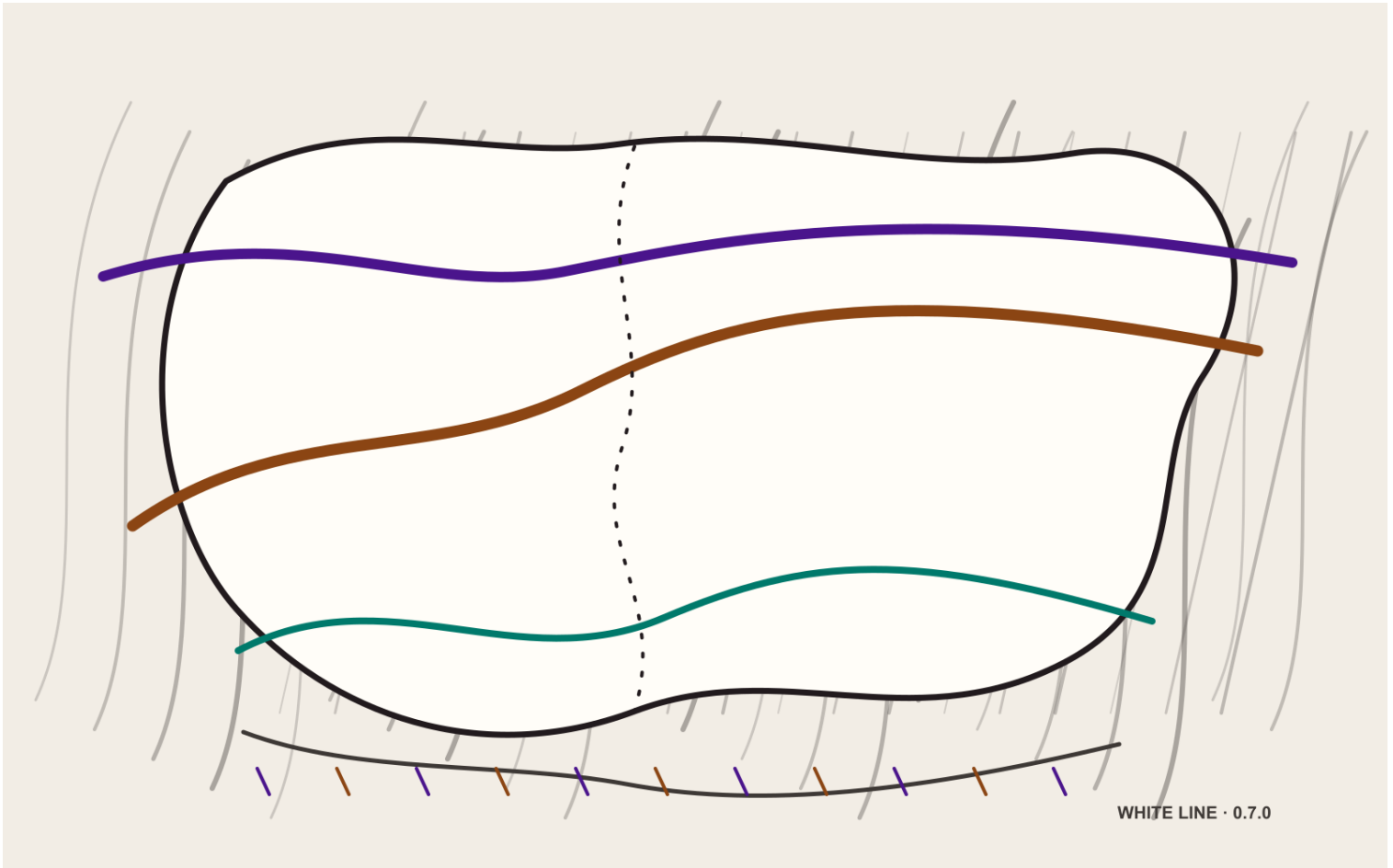


Figure 60: White Line’s cover art: congested graphite marks press toward an irregular open field while three typed traces cross it without closing it. The art is a deterministic conceptual composition for interpretation, not evidence of a hidden cause.

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, 2007a].

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. 6.5 defines the three kinds of absence and the four ledger states. Section sec. 6.9 states the evaluator and its invariants formally. Section sec. 6.10 walks through worked ledger entries, and Section sec. 6.12 closes with the instrument’s safeguards and limits.

6.3 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.

6.3.1 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`.

6.4 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.

6.4.1 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, 1988a]. 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.

6.4.2 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, 2020a]. 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.

6.4.3 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.

6.4.4 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, 2014a]. 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, 2007a]. 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, 2004a]. 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.

6.4.5 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, 1988a, 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, 2014a, Nissenbaum, 2004a].
- **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, 2007a, 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 61: 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.

6.5 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

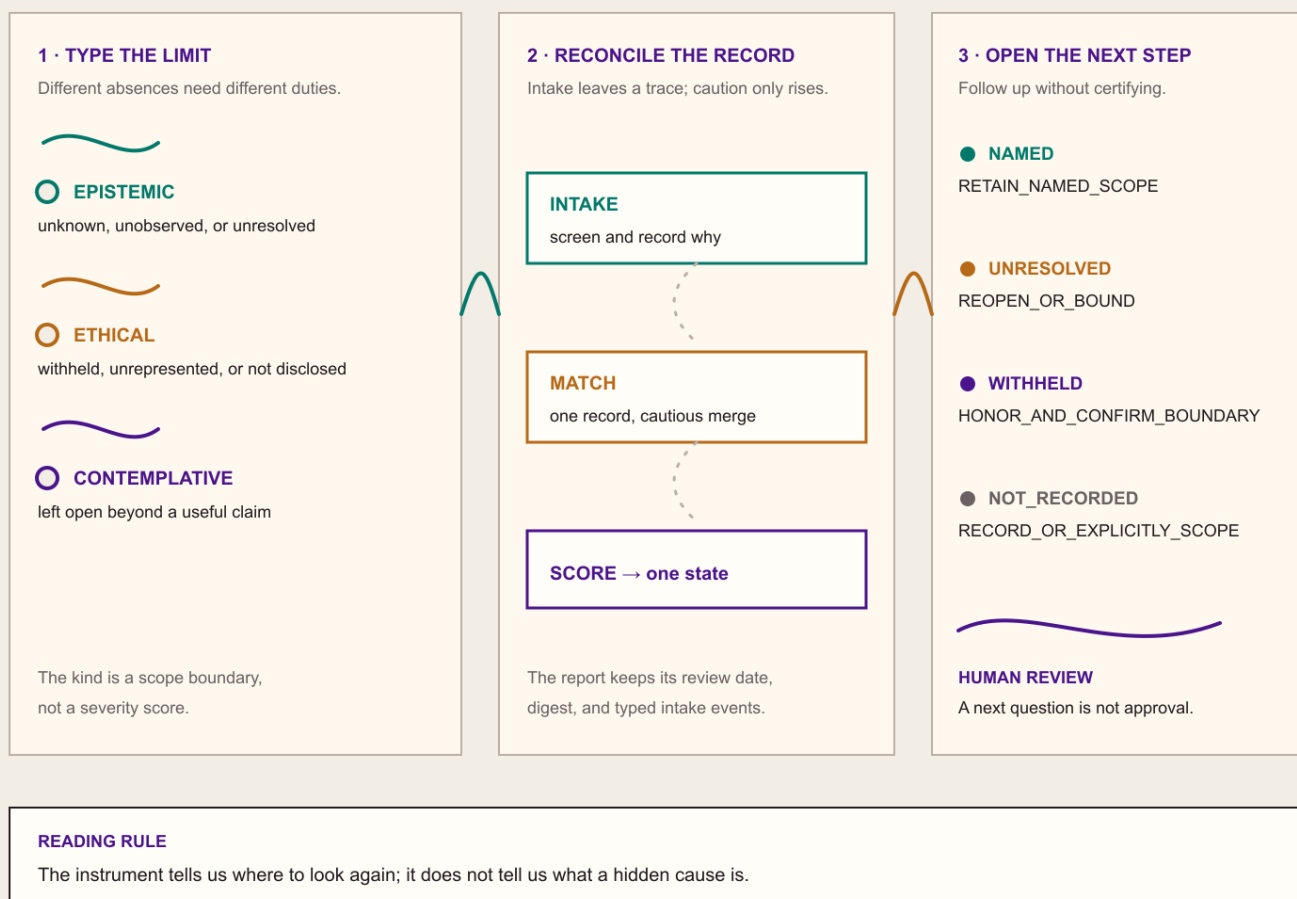


Figure 62: The White Line review atlas: type the limit, reconcile the record, and open a bounded next question. Counts and actions are derived from the live registry, state alphabet, and review protocol; the atlas is not a summary of an observed population.

6.6 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.

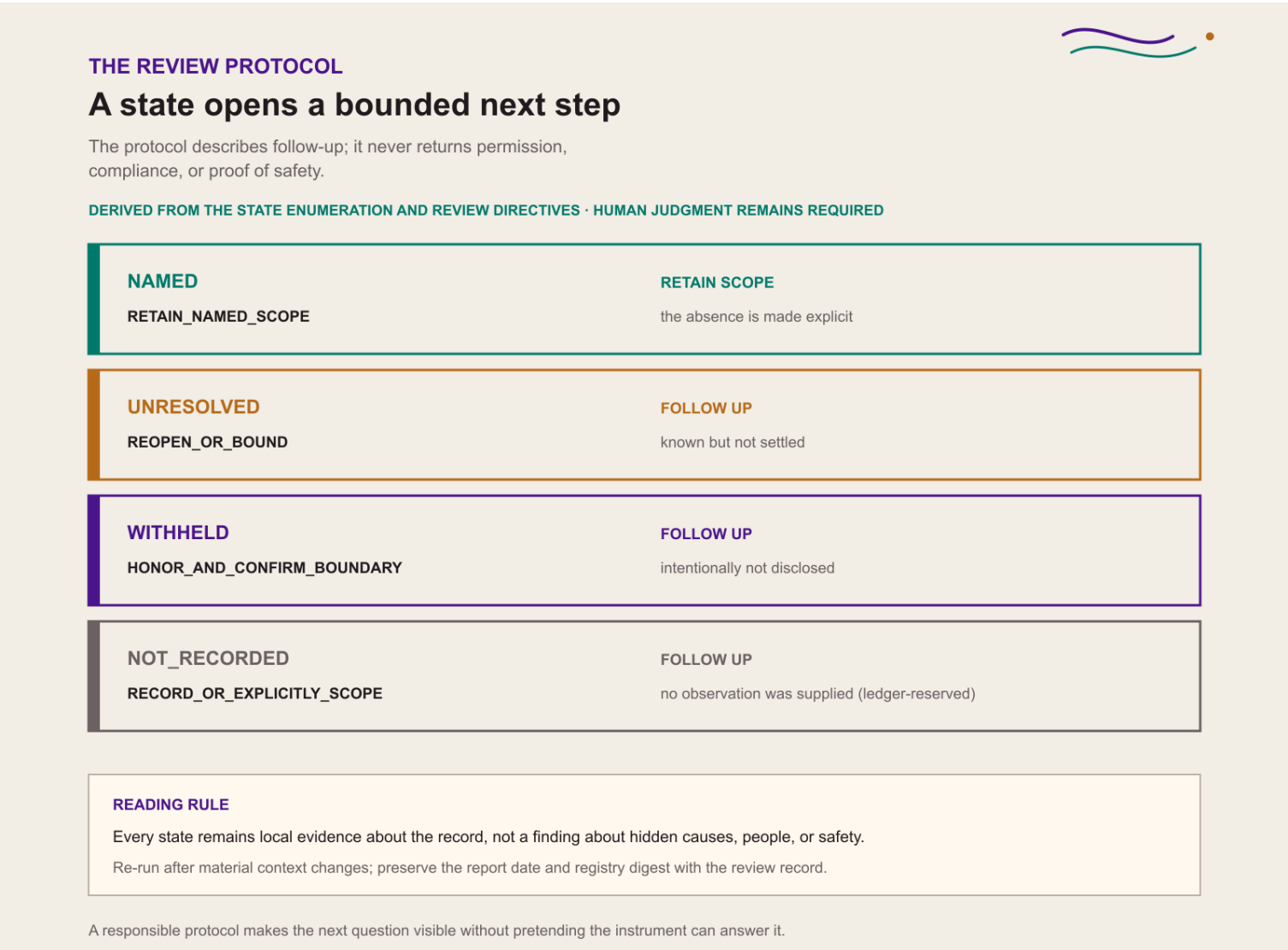


Figure 63: The bounded White Line review protocol maps each state to human follow-up without producing permission, compliance, or safety approval. It is derived from the state enumeration and review directives, not from observed safety data.

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.

6.7 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.

6.8 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.

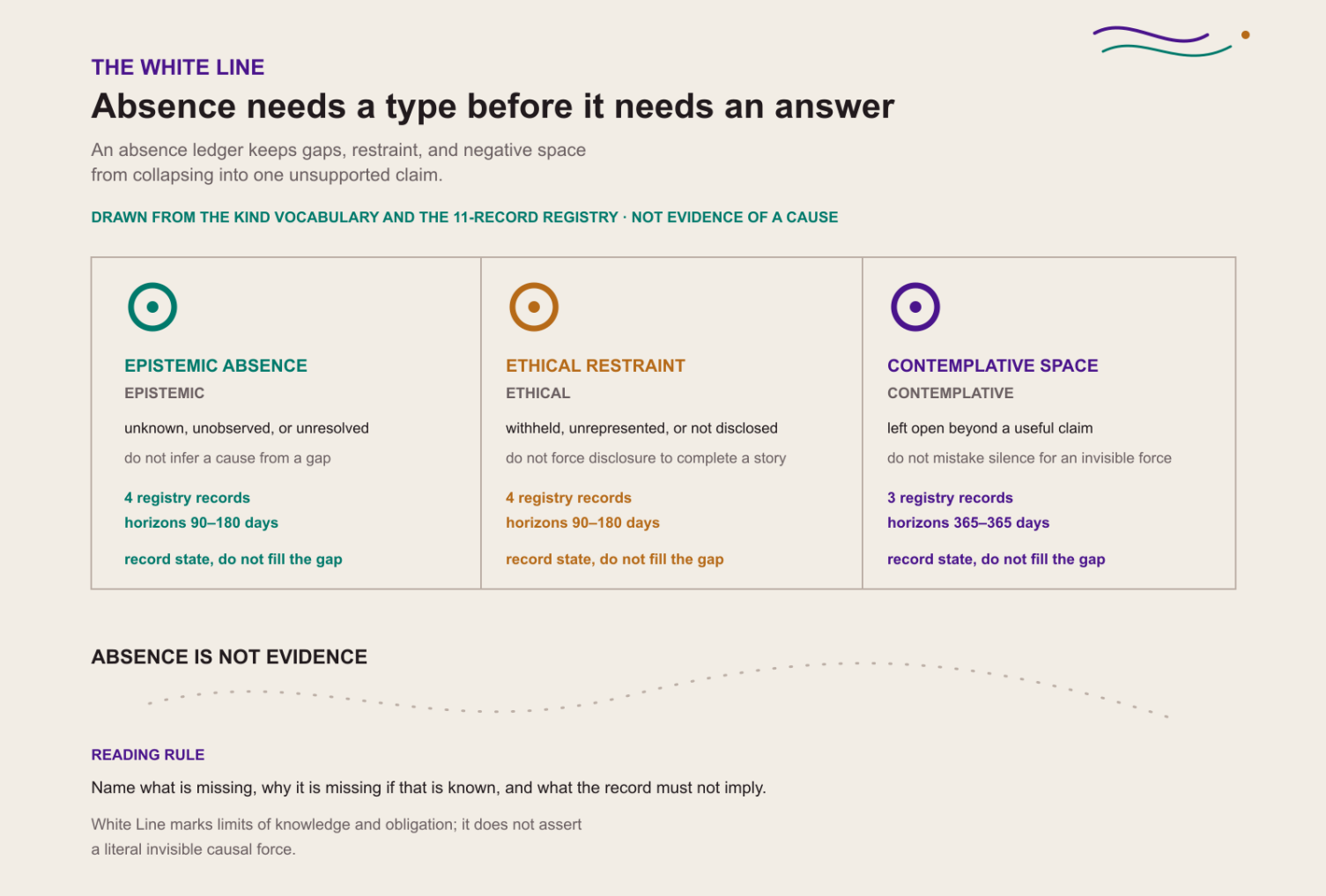


Figure 64: White Line separates three kinds of absence: epistemic absence, ethical restraint, and contemplative space. Each panel is generated from the **AbsenceKind** vocabulary and the shared kind glosses, with its record count and review-horizon range read from the live versioned registry, and each 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 65: 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

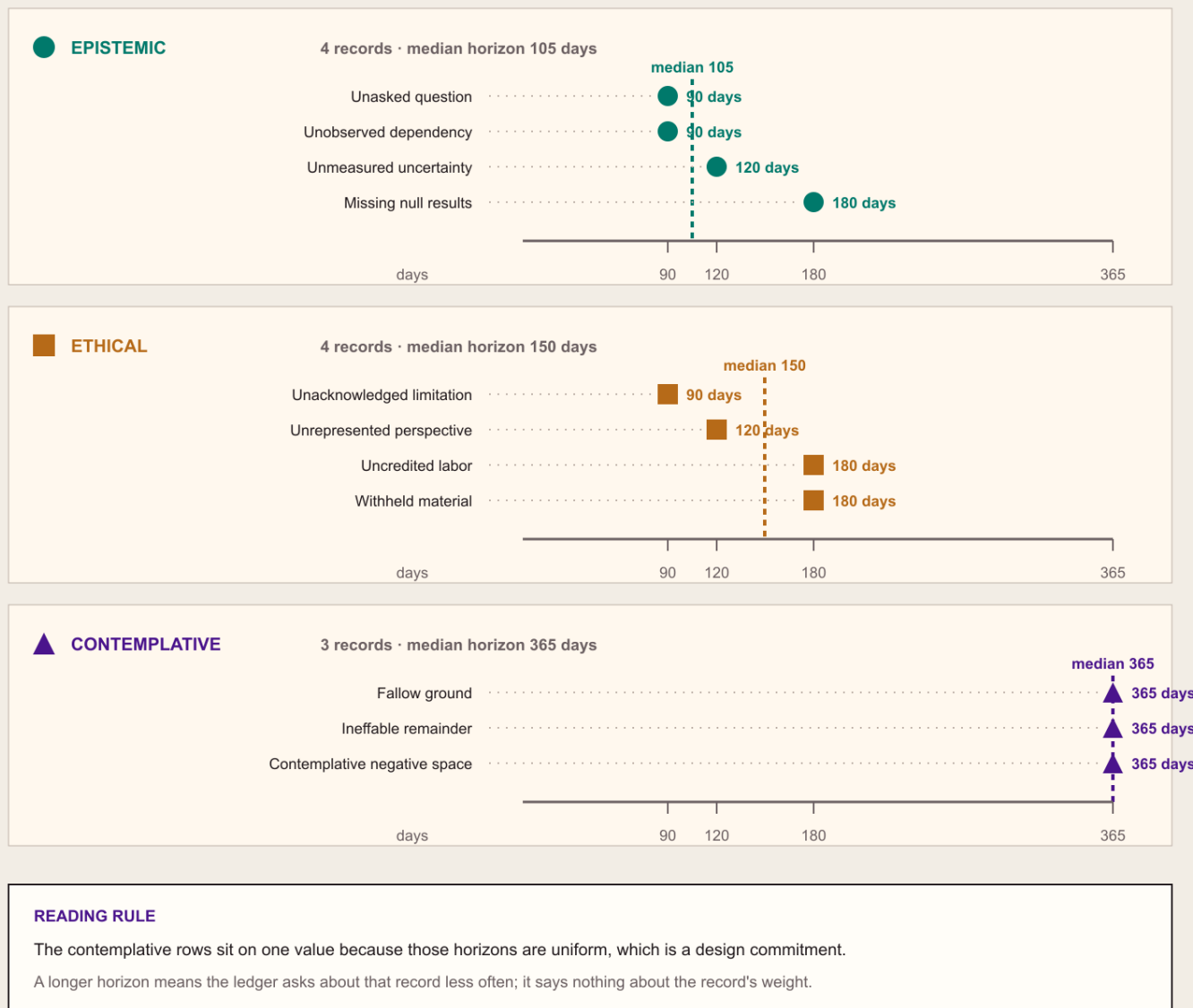


Figure 66: Every registry record placed on a shared days axis by its `review_horizon_days`, grouped into one band per `AbsenceKind` and keyed by colour and by marker shape, with each kind's median horizon marked. Read from `WHITE_RECORDS`. The dispersion shows a structural asymmetry the per-kind range 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.

6.9 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.

6.9.1 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.

6.9.2 The staged evaluator

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

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

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 Π 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:

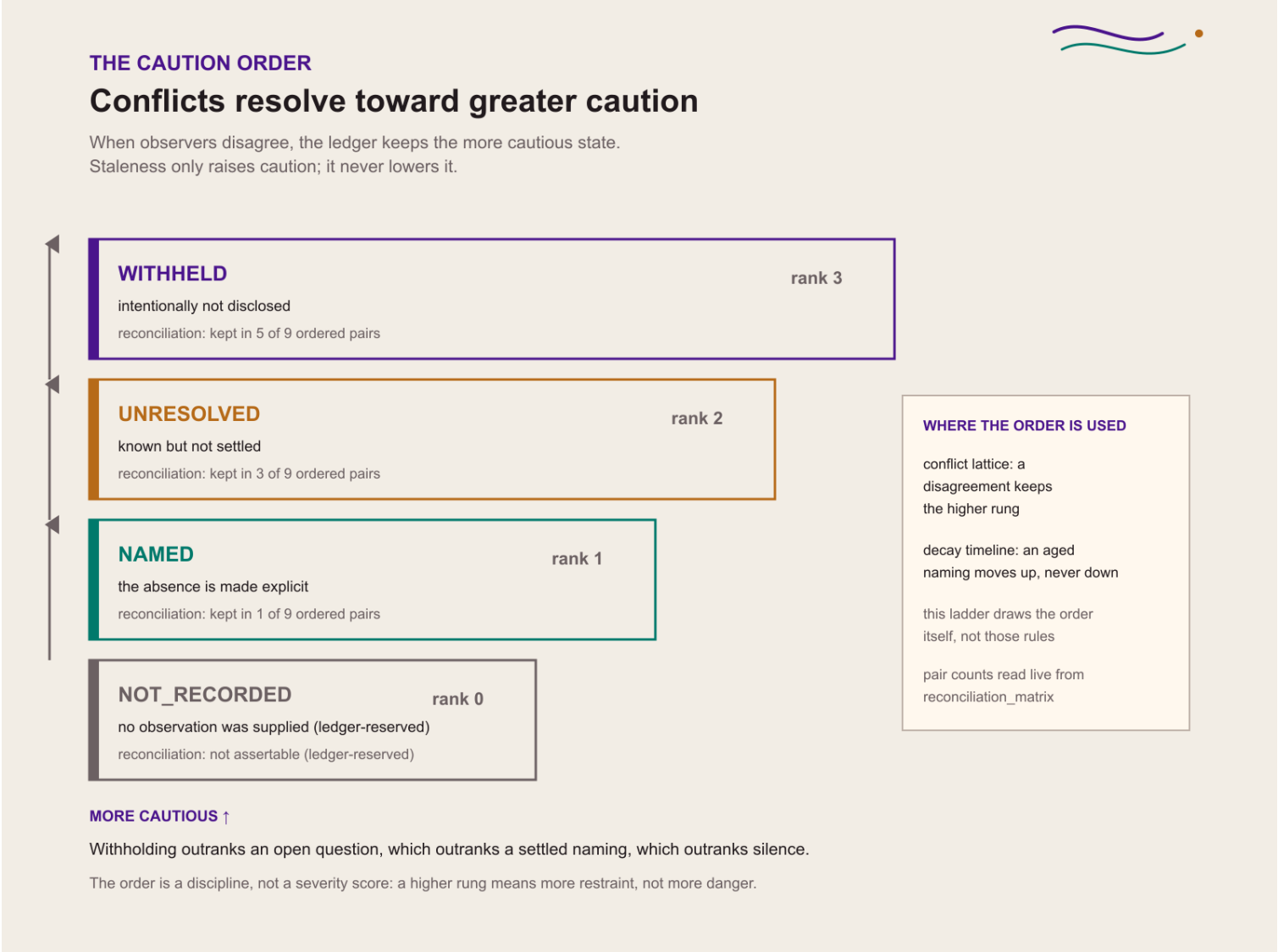


Figure 67: The defined caution order, read directly from `caution_rank`. When observers of one record disagree, the evaluator keeps the higher rung; staleness moves a naming from `NAMED` to `UNRESOLVED`, toward greater caution. Each rung also reports, live from `reconciliation_matrix`, in how many of 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` S 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.

6.9.3 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

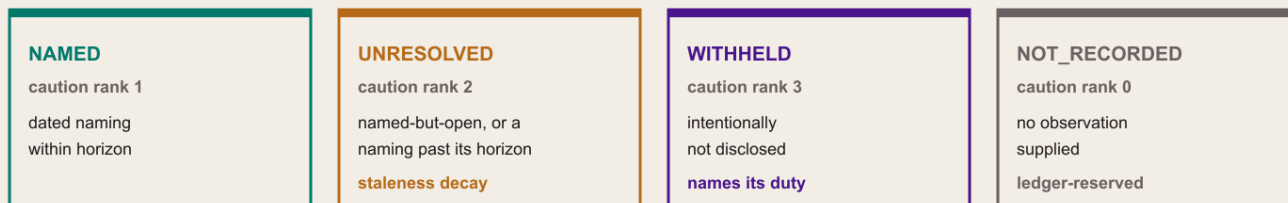
conflicting states for one record resolve toward the more cautious (higher caution rank)

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 68: 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 69: 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.

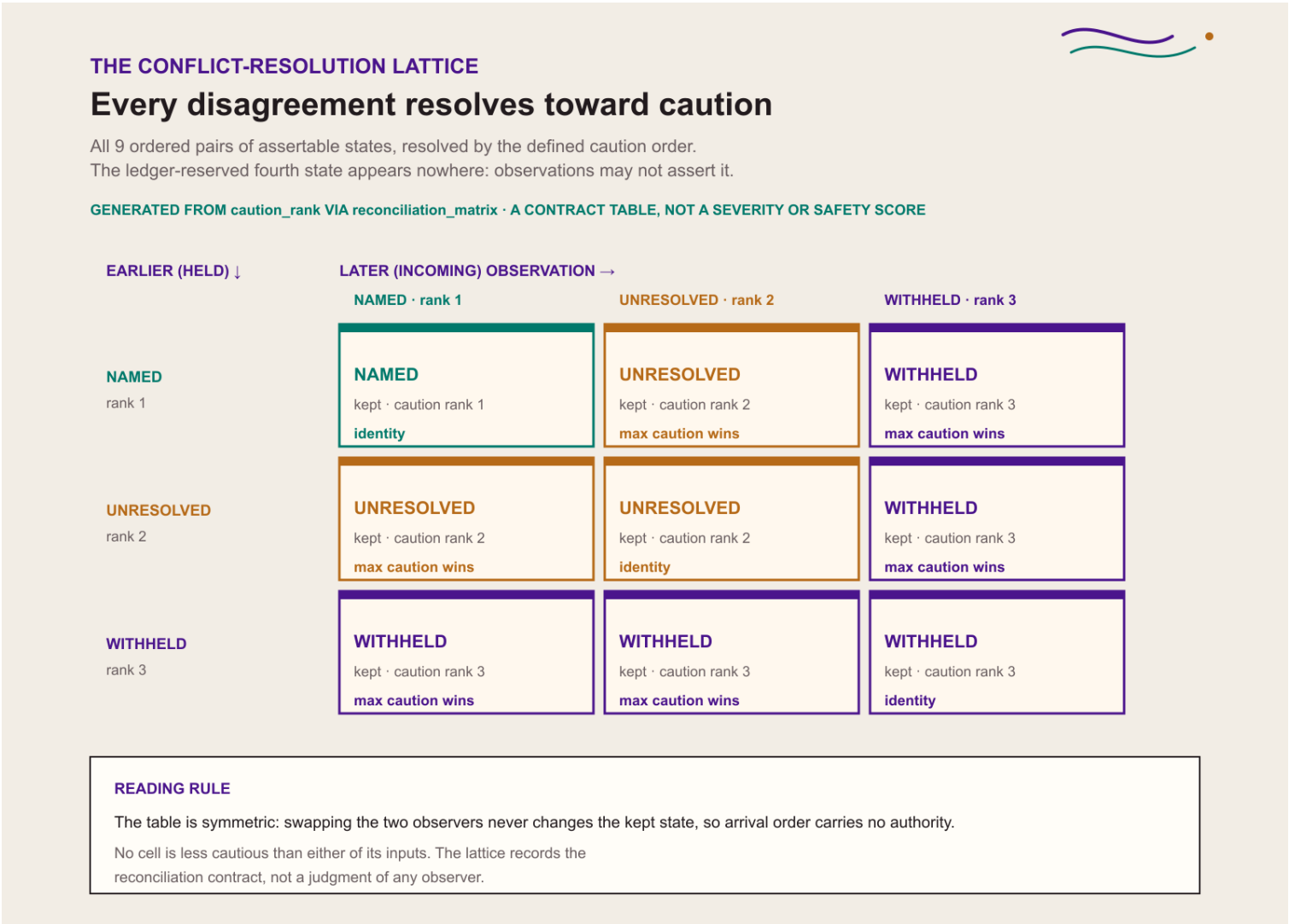


Figure 70: The full ordered-pair conflict-resolution table for assertable states, generated from `caution_rank` via `reconciliation_matrix`. Every cell keeps the more cautious input, the table is symmetric under swapping the observers — arrival order carries no authority — and the ledger-reserved `NOT_RECORDED` 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.

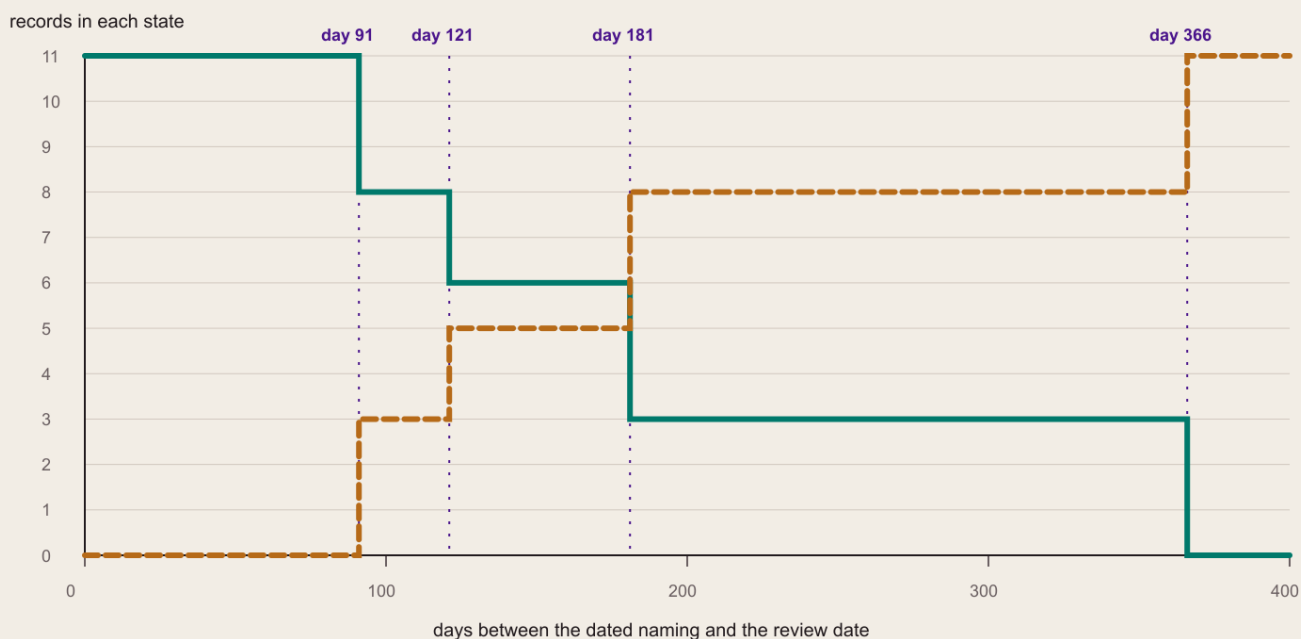
Proposition 4 (No silent discard). Every set-aside input, resolved conflict, and same-state duplicate contributes a typed event and a note to `intake_notes`. Discarding is therefore always visible in the report.

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 71: 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.

6.9.4 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.

6.9.5 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 `^[a-z][a-z0-9]*(-[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.

6.9.6 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 72: 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.

6.10 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.

6.11 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.

6.11.1 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.

6.11.2 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 73: 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.

6.11.3 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.

6.11.4 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.

6.11.5 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 ledger can have recorded; and the marked cells are where the live scoring battery’s findings land when each is factored through



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 74: 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.

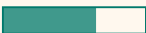


THE REFRESH QUEUE

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 75: 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.

`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 76: 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.

6.12 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.

6.13 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

- African Union Commission. African union data policy framework. African Union, 2022. URL <https://au.int/fr/node/42078>. Accessed 2026-07-17.
- Aristotle. *Politics*. -350. URL <https://classics.mit.edu/Aristotle/politics.html>. Ancient Greek political text; accessed 2026-07-17.
- Aristotle. *Nicomachean Ethics*. Hackett Publishing Company, Indianapolis, 2 edition, 1999. ISBN 9780872204645. URL https://archive.org/details/isbn_9780872204645. Translated, with introduction, notes, and glossary, by Terence Irwin. Accessed 2026-07-28.
- Aristotle. *Nicomachean Ethics*. Various critical editions, 350 BCE. Traditionally dated to the fourth century BCE; on telos, eudaimonia, the mean, and practical wisdom (phronesis). Translated editions vary. Accessed 2026-07-18.
- Monya Baker. 1,500 scientists lift the lid on reproducibility. *Nature*, 533(7604):452–454, 2016. doi: 10.1038/533452a. URL <https://www.nature.com/articles/533452a>. Survey of 1,576 researchers. Accessed 2026-07-28.
- Abeba Birhane. Algorithmic colonization of africa. *SCRIPTed: A Journal of Law, Technology and Society*, 17(2):389–409, 2020. doi: 10.2966/scrip.170220.389. URL <https://doi.org/10.2966/scrip.170220.389>.
- 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 systems and standards as consequential, and largely invisible, information infrastructures. Accessed 2026-07-27.
- Tim Bray, Dave Hollander, and Andrew Layman. Namespaces in XML. W3c recommendation, World Wide Web Consortium, January 1999. URL <https://www.w3.org/TR/1999/REC-xml-names-19990114/>. W3C Recommendation of 14 January 1999. Qualifies element and attribute names by binding a short prefix to a URI, so that names minted by different authorities cannot collide. Bibliographic record verified 2026-07-28.
- Simone Browne. *Dark Matters: On the Surveillance of Blackness*. Duke University Press, 2015. doi: 10.1215/9780822375302. URL <https://read.dukeupress.edu/books/book/147/Dark-MattersOn-the-Surveillance-of-Blackness>. Accessed 2026-07-17.
- Miles Brundage, Shahar Avin, Jack Clark, Helen Toner, Peter Eckersley, Ben Garfinkel, Allan Dafoe, Paul Scharre, Thomas Zeitsoff, Bobby Filar, Hyrum Anderson, Heather Roff, Gregory C. Allen, Jacob Steinhardt, Carrick Flynn, Seán Ó hÉigeartaigh, S. J. Beard, Haydn Belfield, Clare Lyle, Rebecca Crotoft, Owain Evans, Michael Page, Joanna Bryson, Roman Yampolskiy, and Dario Amodè. The malicious use of artificial intelligence: Forecasting, prevention, and mitigation. arXiv preprint arXiv:1802.07228, 2018. URL <https://arxiv.org/abs/1802.07228>. Accessed 2026-07-17. Research report/preprint; not a forecast of this project’s threat likelihood.
- 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.
- Alberto Cairo. *The Truthful Art: Data, Charts, and Maps for Communication*. New Riders, 2016. URL <https://www.oreilly.com/library/view/the-truthful-art/9780133440492/>. Accessed 2026-07-18.
- Donald T. Campbell. Assessing the impact of planned social change. *Evaluation and Program Planning*, 2(1):67–90, 1979. doi: 10.1016/0149-7189(79)90048-X. URL [https://doi.org/10.1016/0149-7189\(79\)90048-X](https://doi.org/10.1016/0149-7189(79)90048-X). Accessed 2026-07-22.
- Jon F. Claerbout and Martin Karrenbach. Electronic documents give reproducible research a new meaning. In *SEG Technical Program Expanded Abstracts 1992*, pages 601–604. Society of Exploration Geophysicists, 1992. doi: 10.1190/1.1822162. URL <https://library.seg.org/doi/abs/10.1190/1.1822162>. Accessed 2026-07-28.
- Harry Collins. *Tacit and Explicit Knowledge*. University of Chicago Press, Chicago, 2010. ISBN 9780226113807. URL <https://press.uchicago.edu/ucp/books/book/chicago/T/bo8461024.html>. Accessed 2026-07-28.
- Confucius. *The Analects*. Various critical editions, 500 BCE. Compiled and transmitted in early Chinese traditions; date is approximate. On learning as the cultivation of conduct within relationships. Accessed 2026-07-18.
- Melvin E. Conway. How do committees invent? *Datamation*, 14(5):28–31, 1968. URL https://www.melconway.com/Home/Committees_Paper.html. Argues that a system’s structure tends to copy the communication structure of the organization that designed it. Accessed 2026-07-27.
- Sasha Costanza-Chock. *Design Justice: Community-Led Practices to Build the Worlds We Need*. The MIT Press, 2020. ISBN 9780262043458. URL <https://mitpress.mit.edu/9780262043458/design-justice>. Accessed 2026-07-18. Used for community-led design, power, and participation; not a safety certification or consent shortcut.
- Nick Couldry and Ulises A. Mejias. *The Costs of Connection: How Data Is Colonizing Human Life and Appropriating It for Capitalism*. Stanford University Press, 2019. doi: 10.1515/9781503609754. URL <https://www.sup.org/books/sociology/costs-connection>. Accessed 2026-07-17.

- Penny Crofts and Honni van Rijswijk. Negotiating ‘evil’: Google, project maven and the corporate form. *Law, Technology and Humans*, 2(1):75–90, 2020. doi: 10.5204/lthj.v2i1.1313. URL <https://doi.org/10.5204/lthj.v2i1.1313>. Accessed 2026-07-28. Used as the documented case of collective worker refusal of military AI work and its corporate absorption; not evidence about the outcome of any individual refusal.
- Ottobah Cugoano. *Thoughts and Sentiments on the Evil and Wicked Traffic of the Slavery and Commerce of the Human Species*. London, 1787. URL <https://quod.lib.umich.edu/e/eccodemo/K046227.0001.001/1%3A3?rgn=div1&view=fulltext>. Primary text; accessed 2026-07-17. Used as a situated Black Atlantic source on liberty, consent, responsibility, and moral self-deception.
- Adam Dahl. Ottobah cugoano. Stanford Encyclopedia of Philosophy, 2025. URL <https://plato.stanford.edu/entries/cugoano/>. Scholarly overview accessed 2026-07-17; used to situate interpretive debates rather than to certify a universal African philosophy.
- Linda T. Darling. Social cohesion (asabiyya) and justice in the late medieval middle east. *Comparative Studies in Society and History*, 49(2):329–357, 2007. doi: 10.1017/S0010417507000515. URL <https://www.cambridge.org/core/journals/comparative-studies-in-society-and-history/article/abs/social-cohesion-asabiyya-and-justice-in-the-late-medieval-middle-east/3117D292647ECD6472E9C77AA294D2A2>.
- Catherine D’Ignazio and Lauren F. Klein. *Data Feminism*. The MIT Press, 2020a. ISBN 9780262044004. URL <https://mitpress.mit.edu/9780262044004/data-feminism/>. Intersectional account of data science, power, classification, and data ethics. Accessed 2026-07-18.
- Catherine D’Ignazio and Lauren F. Klein. *Data Feminism*. The MIT Press, 2020b. ISBN 9780262044004. URL <https://mitpress.mit.edu/9780262044004/data-feminism/>. Accessed 2026-07-18. Used for power-aware classification, invisible labor, and the limits of data speaking for themselves; not an exhaustive ethics standard.
- Edsger W. Dijkstra. On the role of scientific thought. In *Selected Writings on Computing: A Personal Perspective*, pages 60–66. Springer-Verlag, 1982. Circulated as EWD447, dated 30 August 1974; the essay that names “separation of concerns” as studying one aspect at a time while knowing the others still hold. Accessed 2026-07-27.
- 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.
- Hubert L. Dreyfus and Stuart E. Dreyfus. *Mind over Machine: The Power of Human Intuition and Expertise in the Era of the Computer*. Free Press, New York, 1986. ISBN 9780029080610. URL <https://archive.org/details/mindovermachinep00drey>. Accessed 2026-07-28.
- 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.
- Wendy Nelson Espeland and Michael Sauder. Rankings and reactivity: How public measures recreate social worlds. *American Journal of Sociology*, 113(1):1–40, 2007. doi: 10.1086/517897. Empirical study of law-school rankings identifying two mechanisms of reactivity — self-fulfilling prophecy and commensuration — by which a public measure changes the conduct it claims to describe. Accessed 2026-07-28.
- Virginia Eubanks. *Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor*. St. Martin’s Press, 2018. ISBN 9781250074317. URL <https://us.macmillan.com/books/9781250074317/automatinginequality>. Accessed 2026-07-17.
- Richard P. Feynman. Cargo cult science. *Engineering and Science*, 37(7):10–13, 1974. URL <https://calteches.library.caltech.edu/51/2/CargoCult.htm>. Caltech commencement address. Accessed 2026-07-18.
- Viktor E. Frankl. *Man’s Search for Meaning*. Beacon Press, 1959. English translation; German original *Ein Psychologe erlebt das Konzentrationslager* (1946). On meaning as an orienting direction. Accessed 2026-07-18.
- Miranda Fricker. *Epistemic Injustice: Power and the Ethics of Knowing*. Oxford University Press, 2007a. 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.
- Miranda Fricker. *Epistemic Injustice: Power and the Ethics of Knowing*. Oxford University Press, 2007b. ISBN 9780198237907. URL <https://academic.oup.com/book/32817>. Accessed 2026-07-17.
- 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.
- Charles A. E. Goodhart. Problems of monetary management: The u.k. experience. In *Papers in Monetary Economics*, volume 1, pages 1–20. 1975. URL <https://www.econbiz.de/Record/problems-of-monetary-management-the-u-k-experience-goodhart-charles/10002525062>. Original monetary-policy context for the control-target warning later associated with Goodhart’s law. Accessed 2026-07-18.

- Steven N. Goodman, Daniele Fanelli, and John P. A. Ioannidis. What does research reproducibility mean? *Science Translational Medicine*, 8(341):341ps12, 2016. doi: 10.1126/scitranslmed.aaf5027. URL <https://pubmed.ncbi.nlm.nih.gov/27252173/>. Accessed 2026-07-18.
- Mary L. Gray and Siddharth Suri. *Ghost Work: How to Stop Silicon Valley from Building a New Global Underclass*. Houghton Mifflin Harcourt, 2019. ISBN 9781328566249. URL <https://marylgray.org/bio/on-demand/>. Accessed 2026-07-17. Used for hidden labor and human judgment at the boundary of automated systems.
- 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, 1988a. URL <https://www.jstor.org/stable/3178066>. Account of partial, located knowledge and a non-transcendent conception of objectivity. Accessed 2026-07-18.
- Donna Haraway. Situated knowledges: The science question in feminism and the privilege of partial perspective. *Feminist Studies*, 14(3):575–599, 1988b. doi: 10.2307/3178066. URL <https://www.jstor.org/stable/3178066>. Accessed 2026-07-18. Used for situated perspective and accountable partiality, not relativism or a substitute for evidence.
- 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.
- Elisa D. Harris, editor. *Governance of Dual-Use Technologies: Theory and Practice*. American Academy of Arts and Sciences, Cambridge, MA, 2016. URL <https://www.amacad.org/publication/governance-dual-use-technologies-theory-and-practice>. Accessed 2026-07-28. Comparative study of nuclear, biological, and cyber dual-use governance; used for the layered structure of control regimes, not as an assessment of this project.
- Andrew Hunt and David Thomas. *The Pragmatic Programmer: From Journeyman to Master*. Addison-Wesley, Boston, 1999. URL <https://www.oreilly.com/library/view/the-pragmatic-programmer/9780135956977/>. Accessed 2026-07-18.
- Steven J. Jackson. Rethinking repair. In *Media Technologies: Essays on Communication, Materiality, and Society*. MIT Press, 2014. doi: 10.7551/mitpress/9780262525374.003.0011. URL <https://academic.oup.com/mit-press-scholarship-online/book/14976/chapter-abstract/169335302>. On broken-world thinking and repair as a site of creativity, knowledge, power, and care. Accessed 2026-07-18.
- Sheila Jasanoff. Technologies of humility: Citizen participation in governing science. *Minerva*, 41(3):223–244, 2003. doi: 10.1023/A:1025557512320. URL <https://doi.org/10.1023/A:1025557512320>. Accessed 2026-07-18. Used for framing, vulnerability, distribution, and learning under uncertainty; not a local decision procedure.
- Anna Jobin, Marcello Ienca, and Effy Vayena. The global landscape of AI ethics guidelines. *Nature Machine Intelligence*, 1(9):389–399, 2019. doi: 10.1038/s42256-019-0088-2. URL <https://doi.org/10.1038/s42256-019-0088-2>.
- Hans Jonas. *The Imperative of Responsibility: In Search of an Ethics for the Technological Age*. University of Chicago Press, 1984. English edition; German original *Das Prinzip Verantwortung* (1979). On long-horizon duty toward future life. Accessed 2026-07-18.
- C. G. Jung. *Psychology and Alchemy*. Collected Works of C. G. Jung, Volume 12; Bollingen Series XX. Princeton University Press, 1953. German original *Psychologie und Alchemie* (1944). Reads nigredo, albedo, citrinitas, and rubedo as figures for psychological transformation rather than as laboratory chemistry. Cited here only for that symbolic register. Accessed 2026-07-27.
- Kautilya. *King, Governance, and Law in Ancient India: Kautilya’s Arthashastra*. Oxford University Press, 2013. doi: 10.1093/acprof:osobl/9780199891825.001.0001. URL <https://academic.oup.com/book/8486>. Annotated translation; accessed 2026-07-17.
- 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.
- Ibn Khaldun. *The Muqaddimah: An Introduction to History*. Princeton University Press (Rosenthal translation, 1958), 1377. Prolegomena to the history of the world; on how purposes and cohesion are carried by institutions. Accessed 2026-07-18.
- Donald E. Knuth. Literate programming. *The Computer Journal*, 27(2):97–111, 1984. doi: 10.1093/comjnl/27.2.97. URL <https://academic.oup.com/comjnl/article/27/2/97/343244>. Accessed 2026-07-18.
- Tahu Kukutai and John Taylor, editors. *Indigenous Data Sovereignty: Toward an Agenda*. Centre for Aboriginal Economic Policy Research. ANU Press, 2016. doi: 10.22459/CAEPR38.11.2016. URL <https://press.anu.edu.au/publications/series/caepr/indigenous-data-sovereignty>. Accessed 2026-07-18. Used for collective data authority and Indigenous governance of data, not as a universal consent shortcut or substitute for community authority.

- Bartolomé de Las Casas. *A Short Account of the Destruction of the Indies*. Penguin Classics, 1552. ISBN 9780140445626. URL <https://www.penguin.co.uk/books/35187/a-short-account-of-the-destruction-of-the-indies-by-bartolome-de-las-casas-ed-and-trans-by-nigel-griffin-intro-anthony-pagden/9780140445626>. Spanish primary polemic composed 1542 and published 1552; English edition accessed 2026-07-17. Used as a situated colonial-era source, not Indigenous testimony or a universal authority.
- 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.
- Niccolo Machiavelli. *The Prince*. 1513. URL <https://www.gutenberg.org/ebooks/1232>. Primary text; accessed 2026-07-17.
- Alasdair MacIntyre. *After Virtue: A Study in Moral Theory*. University of Notre Dame Press, 1981. On practices, goods internal to a practice, and the narrative unity of a life toward a telos. Accessed 2026-07-18.
- Muhsin Mahdi. FARABI vi. political philosophy. *Encyclopaedia Iranica*, 2000. URL <https://www.iranicaonline.org/articles/farabi-vi/>. Accessed 2026-07-17.
- 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.
- David Manheim and Scott Garrabrant. Categorizing variants of goodhart’s law. arXiv preprint arXiv:1803.04585, 2018. URL <https://arxiv.org/abs/1803.04585>. Separates four mechanisms by which optimizing a proxy collapses its relationship to the goal — regressional, extremal, causal, and adversarial — and notes that Campbell’s law arguably has scholarly precedence over the later Goodhart formulations. Accessed 2026-07-28.
- 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.
- Robert K. Merton. The self-fulfilling prophecy. *The Antioch Review*, 8(2):193–210, 1948. doi: 10.2307/4609267. On how a definition of a situation can evoke behavior that makes the definition come true. Accessed 2026-07-18.
- Robert K. Merton. The normative structure of science. In Norman W. Storer, editor, *The Sociology of Science: Theoretical and Empirical Investigations*, pages 267–278. University of Chicago Press, Chicago, 1973. URL <https://press.uchicago.edu/ucp/books/book/chicago/S/bo28173694.html>. Essay originally published 1942. Accessed 2026-07-18.
- 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.
- MITRE. MITRE ATT&CK enterprise techniques. MITRE ATT&CK knowledge base, 2026. URL <https://attack.mitre.org/techniques/>. Accessed 2026-07-17. Threat-model vocabulary, not attribution or telemetry.
- Shakir Mohamed, Marie-Therese Png, and William Isaac. Decolonial AI: Decolonial theory as sociotechnical foresight in artificial intelligence. *Philosophy and Technology*, 33(4):659–684, 2020. doi: 10.1007/s13347-020-00405-8. URL <https://doi.org/10.1007/s13347-020-00405-8>.
- Geert Molenberghs, Caroline Beunckens, Cristina Sotto, 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.

- Marcus R. Munafò, Brian A. Nosek, Dorothy V. M. Bishop, et al. A manifesto for reproducible science. *Nature Human Behaviour*, 1:0021, 2017. doi: 10.1038/s41562-016-0021. URL <https://www.nature.com/articles/s41562-016-0021>. Accessed 2026-07-18.
- National Academies of Sciences, Engineering, and Medicine. *Reproducibility and Replicability in Science*. National Academies Press, Washington, DC, 2019. doi: 10.17226/25303. URL <https://nap.nationalacademies.org/catalog/25303/reproducibility-and-replicability-in-science>. Consensus Study Report. Accessed 2026-07-18.
- National Research Council. Biotechnology research in an age of terrorism. Technical report, The National Academies Press, Washington, DC, 2004. URL <https://nap.nationalacademies.org/catalog/10827/biotechnology-research-in-an-age-of-terrorism>. Known as the Fink report, after committee chair Gerald R. Fink. Accessed 2026-07-28. Used for the proposition that researcher-level judgment is a governance layer, not for its biosecurity subject matter.
- Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(1):119–158, 2004a. 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.
- Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 79(1):119–158, 2004b. URL <https://digitalcommons.law.uw.edu/wlr/vol79/iss1/10/>.
- Safiya Umoja Noble. *Algorithms of Oppression: How Search Engines Reinforce Racism*. New York University Press, 2018. ISBN 9781479837243. URL <https://nyupress.org/9781479837243/algorithms-of-oppression/>. Accessed 2026-07-17.
- Brian A. Nosek, George Alter, George C. Banks, et al. Promoting an open research culture. *Science*, 348(6242):1422–1425, 2015. doi: 10.1126/science.aab2374. URL <https://pubmed.ncbi.nlm.nih.gov/26113702/>. Accessed 2026-07-18.
- Martha C. Nussbaum. *Creating Capabilities: The Human Development Approach*. The Belknap Press of Harvard University Press, 2011. On the central human capabilities and a threshold conception of flourishing. Accessed 2026-07-18.
- Nāgārjuna. *The Fundamental Wisdom of the Middle Way: Nāgārjuna’s Mūlamadhyamakārikā*. 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.
- OECD. Oecd principles on artificial intelligence. Organisation for Economic Co-operation and Development, 2019. URL <https://www.oecd.org/en/topics/ai-principles.html>. Updated page accessed 2026-07-17; principles adopted 2019.
- Open Science Collaboration. Estimating the reproducibility of psychological science. *Science*, 349(6251):aac4716, 2015. doi: 10.1126/science.aac4716. URL <https://www.science.org/doi/10.1126/science.aac4716>. Accessed 2026-07-28.
- Naomi Oreskes 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.
- Naomi Oreskes, Kristin Shrader-Frechette, and Kenneth Belitz. Verification, validation, and confirmation of numerical models in the earth sciences. *Science*, 263(5147):641–646, 1994. doi: 10.1126/science.263.5147.641. URL <https://doi.org/10.1126/science.263.5147.641>. Accessed 2026-07-18.
- Elinor Ostrom. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, 1990a. doi: 10.1017/CBO9780511807763. URL <https://www.cambridge.org/core/books/governing-the-commons/7AB7AE11BADA84409C34815CC288CD79>.
- Elinor Ostrom. *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, 1990b. doi: 10.1017/CBO9780511807763. URL <https://www.cambridge.org/core/books/governing-the-commons/7AB7AE11BADA84409C34815CC288CD79>. Accessed 2026-07-18.
- D. L. Parnas. On the criteria to be used in decomposing systems into modules. *Communications of the ACM*, 15(12):1053–1058, 1972. doi: 10.1145/361598.361623. URL <https://doi.org/10.1145/361598.361623>. Argues that modules should be organized around design decisions each hides from the others, rather than around steps in a flowchart. Accessed 2026-07-27.
- D. L. Parnas. Designing software for ease of extension and contraction. *IEEE Transactions on Software Engineering*, SE-5(2): 128–138, 1979. doi: 10.1109/TSE.1979.234169. URL <https://doi.org/10.1109/TSE.1979.234169>. Treats extension and contraction as one problem, and locates the cost of both in the “uses” relation rather than in the size of the edit. Bibliographic record verified 2026-07-28.
- D. L. Parnas, P. C. Clements, and D. M. Weiss. The modular structure of complex systems. *IEEE Transactions on Software Engineering*, SE-11(3):259–266, 1985. doi: 10.1109/TSE.1985.232209. URL <https://doi.org/10.1109/TSE.1985.232209>. Introduces the module guide: a document that records the decomposition and each module’s secret, kept because the structure is not recoverable from the code. Bibliographic record verified 2026-07-28.

- Roger D. Peng. Reproducible research in computational science. *Science*, 334(6060):1226–1227, 2011. doi: 10.1126/science.1213847. URL <https://www.science.org/doi/10.1126/science.1213847>. Accessed 2026-07-18.
- 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.
- Hans E. Plesser. Reproducibility vs. replicability: A brief history of a confused terminology. *Frontiers in Neuroinformatics*, 11:76, 2018. doi: 10.3389/fninf.2017.00076. URL <https://www.frontiersin.org/journals/neuroinformatics/articles/10.3389/fninf.2017.00076/full>. Accessed 2026-07-28.
- Michael Polanyi. *Personal Knowledge: Towards a Post-Critical Philosophy*. University of Chicago Press, Chicago, 1958. URL <https://press.uchicago.edu/ucp/books/book/chicago/P/bo19722848.html>. Accessed 2026-07-18.
- Michael Polanyi. *The Tacit Dimension*. Doubleday; reprinted University of Chicago Press, 1966. On tacit knowing: *we can know more than we can tell*. Accessed 2026-07-18.
- Karl R. Popper. *The Logic of Scientific Discovery*. Hutchinson & Co., London, 1959. URL <https://www.routledge.com/The-Logic-of-Scientific-Discovery/Popper/p/book/9780415278447>. English translation of Logik der Forschung (1934). Accessed 2026-07-18.
- Michael Power. *The Audit Society: Rituals of Verification*. Oxford University Press, Oxford, 1997. ISBN 9780198289470. Accessed 2026-07-22.
- 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.
- Ingrid Robeyns. *Wellbeing, Freedom and Social Justice: The Capability Approach Re-Examined*. Open Book Publishers, Cambridge, UK, 2017. doi: 10.11647/OBP.0130. URL <https://www.openbookpublishers.com/books/10.11647/obp.0130>. Open-access re-examination distinguishing the capability approach as an open framework from the specific theories built inside it; the framework does not itself supply a metric. Accessed 2026-07-28.
- Scott Rose, Oliver Borchert, Stu Mitchell, and Sean Connelly. Zero trust architecture. Technical Report NIST SP 800-207, National Institute of Standards and Technology, 2020. URL <https://csrc.nist.gov/pubs/sp/800/207/final>.
- 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.
- Gilbert Ryle. *The Concept of Mind*. Hutchinson’s University Library, London, 1949. URL <https://archive.org/details/conceptofmind0000ryle>. Accessed 2026-07-28.
- Geir Kjetil Sandve, Anton Nekrutenko, James Taylor, and Eivind Hovig. Ten simple rules for reproducible computational research. *PLOS Computational Biology*, 9(10):e1003285, 2013. doi: 10.1371/journal.pcbi.1003285. URL <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003285>. Accessed 2026-07-18.
- Donald A. Schön. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, New York, 1983. ISBN 9780465068760. URL <https://www.hachettebookgroup.com/titles/donald-a-schon/the-reflective-practitioner/9780465068784/>. Accessed 2026-07-28.
- James C. Scott. *Seeing Like a State: How Certain Schemes to Improve the Human Condition Have Failed*. Yale University Press, 1998. URL <https://yalebooks.yale.edu/book/9780300246759/seeing-like-a-state/>. Accessed 2026-07-17.
- Andrew D. Selbst, Danah Boyd, Sorelle A. Friedler, Suresh Venkatasubramanian, and Janet Vertesi. Fairness and abstraction in sociotechnical systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 59–68. ACM, 2019. doi: 10.1145/3287560.3287598. URL <https://doi.org/10.1145/3287560.3287598>.
- Michael A. Sells. *Mystical Languages of Unsayng*. 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.

- Amartya Sen. Equality of what? In Sterling M. McMurrin, editor, *The Tanner Lectures on Human Values, Volume 1*. Cambridge University Press, 1980. URL <https://tannerlectures.org/lectures/equality-of-what/>. Tanner Lecture on Human Values delivered at Stanford University, 22 May 1979. Argues that utilitarian, total-utility, and Rawlsian equality each answer the wrong question, and introduces basic capability equality. Accessed 2026-07-28.
- Amartya Sen. *Development as Freedom*. Alfred A. Knopf; Oxford University Press, 1999. On the capability approach: value measured in real freedoms to be and to do. Accessed 2026-07-18.
- Amartya Sen. Capabilities, lists, and public reason: Continuing the conversation. *Feminist Economics*, 10(3):77–80, 2004. doi: 10.1080/1354570042000315163. Sen’s reply on why he declines to fix a canonical list of capabilities: the relevant capabilities depend on the purpose of the assessment and must be settled by public reasoning. Accessed 2026-07-28.
- Richard Sennett. *The Craftsman*. Yale University Press, New Haven, 2008. ISBN 9780300119091. URL <https://yalebooks.yale.edu/book/9780300119091/the-craftsman/>. Accessed 2026-07-28.
- Herbert A. Simon. Rational choice and the structure of the environment. *Psychological Review*, 63(2):129–138, 1956. doi: 10.1037/h0042769. On satisficing and bounded rationality: agents seek *good enough* within a structured environment. Accessed 2026-07-18.
- Audra Simpson. *Mohawk Interruptus: Political Life Across the Borders of Settler States*. Duke University Press, 2014a. 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.
- Audra Simpson. *Mohawk Interruptus: Political Life Across the Borders of Settler States*. Duke University Press, Durham, NC, 2014b. ISBN 9780822356554. URL <https://www.dukeupress.edu/mohawk-interruptus>. Accessed 2026-07-28. Source of the concept of refusal as a positive political stance rather than a deficit; read in its Kahnawà:ke and settler-colonial context and explicitly not transferred as a template for a practitioner declining paid work.
- SLSA Community. Slsa specification version 1.2. Linux Foundation community specification, 2026. URL <https://slsa.dev/spec/v1.2/>. Approved specification accessed 2026-07-17; implementation context, not a project attestation.
- Linda Tuhiwai Smith. *Decolonizing Methodologies: Research and Indigenous Peoples*. Zed Books, 1999. URL <https://www.royalsociety.org.nz/150th-anniversary/tetakarangi/decolonizing-methodologieslinda-tuhiwai-smith-1999>. Accessed 2026-07-17.
- Murugiah Souppaya, Karen Scarfone, and Donna Dodson. Secure software development framework (SSDF) version 1.1. Technical Report NIST SP 800-218, National Institute of Standards and Technology, 2022. URL <https://csrc.nist.gov/pubs/sp/800/218/final>. Implementation context; accessed 2026-07-17; not evidence of Red Line compliance.
- Susan Leigh Star. The structure of ill-structured solutions: Boundary objects and heterogeneous distributed problem solving. In Les Gasser and Michael N. Huhns, editors, *Distributed Artificial Intelligence, Volume II*, pages 37–54. Pitman, London, 1989. The companion 1989 statement of boundary objects, written for distributed problem solving and giving the typology: repositories, ideal types, coincident boundaries, and standardised forms. Bibliographic record verified 2026-07-28.
- Susan Leigh Star. This is not a boundary object: Reflections on the origin of a concept. *Science, Technology, & Human Values*, 35(5):601–617, 2010. doi: 10.1177/0162243910377624. URL <https://doi.org/10.1177/0162243910377624>. Star’s own account of how the concept was taken up, including her objection to its use for any object that happens to sit between groups. Accessed 2026-07-27.
- Susan Leigh Star and James R. Griesemer. Institutional ecology, translations and boundary objects: Amateurs and professionals in berkeley’s museum of vertebrate zoology, 1907–39. *Social Studies of Science*, 19(3):387–420, 1989. doi: 10.1177/030631289019003001. URL <https://doi.org/10.1177/030631289019003001>. Introduces boundary objects: artifacts robust enough to be shared across communities while remaining locally interpretable. Accessed 2026-07-27.
- Marilyn Strathern. ‘improving ratings’: audit in the british university system. *European Review*, 5(3):305–321, 1997a. doi: 10.1002/(SICI)1234-981X(199707)5:3<305::AID-EURO184>3.0.CO;2-4. On audit culture: how an instrument built to describe a practice can begin to reorganize the practice around itself. Accessed 2026-07-27.
- Marilyn Strathern. ‘Improving ratings’: Audit in the British university system. *European Review*, 5(3):305–321, 1997b. doi: 10.1017/S1062798700002660. URL <https://www.cambridge.org/core/journals/european-review/article/abs/improving-ratings-audit-in-the-british-university-system/FC2EE640C0C44E3DB87C29FB666E9AAB>. Accessed 2026-07-22.
- Lucy A. Suchman. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press, Cambridge, 1987. ISBN 0521331374. URL https://books.google.com/books?id=AJ_eBJtHxmsC. Accessed 2026-07-18.
- 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.

- Sunzi. The art of war. Classical Chinese military text; English translation consulted through the Chinese Text Project, 1910. URL <https://ctext.org/art-of-war/laying-plans/ens>. Accessed 2026-07-17. Used as a situated source on information, deception, and strategic judgment, not as an ethical endorsement.
- Elham Tabassi. Artificial intelligence risk management framework (AI RMF 1.0). Technical Report NIST AI 100-1, National Institute of Standards and Technology, 2023. URL <https://www.nist.gov/itl/ai-risk-management-framework>.
- 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.
- The Wassenaar Arrangement on Export Controls for Conventional Arms and Dual-Use Goods and Technologies. List of dual-use goods and technologies and munitions list. Multilateral export-control regime; control lists as amended by the December 2025 Plenary, 2025. URL <https://www.wassenaar.org/control-lists/>. Accessed 2026-07-28. Used as the reference case for a capability-keyed institutional control list, including the definitional over-breadth of the 2013 intrusion-software entry; not a legal authority over this project and not a claim that any listed category applies to it.
- 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.
- Alex Turner. A red line and oversight framework for government AI contracts. The Pond, July 2026. URL <https://turntrout.com/red-line-framework>. Accessed 2026-07-17. Source framework adapted by this project from organization-to-government scale to a single practitioner governing personal development work.
- 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.
- UNESCO. Recommendation on the ethics of artificial intelligence. United Nations Educational, Scientific and Cultural Organization, 2021. URL <https://www.unesco.org/en/legal-affairs/recommendation-ethics-artificial-intelligence?hub=1063>. Adopted 23 November 2021; accessed 2026-07-17.
- Greg Wilson, Jennifer Bryan, Karen Cranston, Justin Kitzes, Lex Nederbragt, and Tracy K. Teal. Good enough practices in scientific computing. *PLOS Computational Biology*, 13(6):e1005510, 2017. doi: 10.1371/journal.pcbi.1005510. URL <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005510>. Accessed 2026-07-18.
- Langdon Winner. Do artifacts have politics? *Daedalus*, 109(1):121–136, 1980. URL <https://www.jstor.org/stable/20024652>. On technical artifacts and systems embodying forms of power and authority. Accessed 2026-07-18.
- 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.
- Mary Wollstonecraft. *A Vindication of the Rights of Woman*. 1792. URL <https://www.gutenberg.org/ebooks/3420>. Primary text; accessed 2026-07-17.
- Jonathan Zong and J. Nathan Matias. Data refusal from below: A framework for understanding, evaluating, and envisioning refusal as design. *ACM Journal on Responsible Computing*, 1(1):1–23, 2024. doi: 10.1145/3630107. URL <https://doi.org/10.1145/3630107>. Accessed 2026-07-28. Used for the four facets of refusal — autonomy, time, power, cost — written from the standpoint of those who refuse; not a claim that a practitioner’s refusal is refusal from below.