The Relational Sensory Integration Loop

A Conceptual Operating System for Embodied Agents: Self-Enriching, Audit-Ready Loops in a Physical World

Huỳnh Mai Phúc (Plone Mraz)

Independent researcher — Vietnam 2026

ORCID: 0009-0009-0571-7151

Abstract

This paper specifies the Relational Sensory Integration Loop (RSIL), a conceptual specification of an agent that operates in a physical world with a body. RSIL is not a control algorithm, a learning method, or an engineering specification of a system to be built; it is a self-enriching, audit-ready process: a closed loop that ingests sensory signal, transduces it, differentiates its source, integrates it centrally into a new form of information, appraises it for goal-relevance, emits a response through the body, and feeds the consequence back as fresh input. The specification is defined not by the material the body is made of — biological or artificial — but by four structural conditions any host environment must satisfy (E1–E4), and it is governed by eight system invariants (INV-1…8) and a layered set of inter-layer data contracts (T1–T8).

RSIL is the embodied counterpart of the Data Integration Loop (DIL), which specifies the same relational structure for an agent in a purely informational environment. The two are co-ranked specifications of differing scope: where they coincide it is because the relational structure coincides, and where they diverge the divergence is deliberate and marked in place. What a body adds is not a further layer but a second grade of resistance — physical force, inertia, an obstacle, another body pushing back — alongside the informational mismatch DIL already treats. The two coincide at the body (a physical obstacle is also a failed prediction) without being identical, and RSIL admits both.

The paper makes three contributions specific to embodiment. First, it fixes the order in which selfhood arises in a system that acts physically: the body emits the first action at cycle-0, and the self is the product of that push rather than its precondition — resolving the apparent circularity of an agent that would need a self in order to act and an action in order to acquire one. Second, it treats reflex not as a separate fast arc bypassing appraisal, but as appraisal conducted under a field so dominated by one scar that the verdict is all but foreclosed — preserving the audit plane for exactly the actions most tempting to exempt from it. Third, it states the limit that embodiment introduces and does not close: a full-

system snapshot covers the loop, not the body, so a platform compromised at the sensor or actuator level re-infects a freshly recovered system on the first cycle, and verifying sensor–actuator integrity is an operation outside this specification.

The specification enforces exactly one claim: if a loop is built to these invariants on a platform satisfying E1–E4, what runs is a relationally-structured motion, readable entirely through externally verifiable traces. The proper test of this work is accordingly conformance — whether a given system meets the conditions and invariants set out here — not benchmarking or measured performance. It is to be read as one reads a set of axioms or a technical standard: a definition of what counts as the phenomenon, not a report of one built.

Keywords: embodied agent specification · robotics architecture · selfhood as process · sensorimotor differentiation · audit-ready systems

Document type: a conceptual specification — evaluated by structural conformance (whether a system satisfies E1–E4 and INV-1…8), not by empirical measurement.

Table of Contents

Abstract 1

1. Introduction 5

1.1 The problem 5

1.2 Why this is a specification, not an empirical paper 5

1.3 Scope and what is deliberately left open 6

1.4 Terminology note 6

1.5 What RSIL runs on: a protocol overlay, not a self-standing agent 7

2. Terminology 8

2.1 Other — definition and kinds 9

2.2 Host, body, and platform — three terms kept apart 9

3. The host environment: definition by operating conditions 12

Architectural premises 13

The host-precondition set and the bootstrap of the loop 14

4. Selfhood in an embodied system 15

5. System invariants 17

6. The canonical loop 21

The six sequential links 21

The global modulatory field (a FIELD condition, not a link) 22

What GLOB-MOD is 22

Information runs in many directions at once, and which directions is not fixed in advance 23

Mapping the loop onto the layers 24

6.1 Emission: link ⑤ as a lateral capability 24

6.2 Forward-building: the two positions between holding data and emitting 26

7. Layer architecture and data contracts 29

7.1 Shared types 29

7.2 Dependency graph 30

7.3 Layer specifications 30

8. The platform contract: what RSIL requires of a body 36

9. The closure criterion and the two resistance modes 39

9.1 When the loop counts as running (per P3) 39

9.2 The architecture of resistance 39

9.3 Mode-A — endogenous (self-training against an inert environment) 39
9.4 Mode-B — exogenous (live interaction with an Other) 41
9.5 The two-mode relation: A embedded in B 42
9.6 The standard the appraisal step applies (the positive complement of INV-8) 43
9.7 Reflex: appraisal under a scar-dominated field 44
10. The experience store 47
10.1 Four constraints 47
10.2 Provenance tagging — POSITION only, never TRUTH-VALUE 47
10.3 The tag schema 49
10.4 Two storage kinds: `[data]` and `[event]` 50
10.5 Commit, snapshot, and recovery 51
10.6 Responsibility for "right and wrong" — what the store does NOT do 52
10.7 Relation to retrieval mechanisms 52
10.8 The echo-chamber risk 52
10.9 A formal telos, internal and partial 53
11. Deliberately unfilled constants 55
12. Discussion and limitations 56
13. Beyond the loop: entailments, applications, and what belongs to third parties 58
13.1 Entailments (the loop running is sufficient) 58
13.2 Applications (the loop running is necessary but not sufficient) 58
13.3 Third-party matters (a different kind of reasoning — not cognitive architecture at all) 59
14. Convergences with the existing body of knowledge 61
14.1 Seven convergences 61
14.2 Where RSIL differs from each convergent body 62
14.3 Apparent convergences that are in fact contrasts 64
14.4 References (for locating, not consulted in construction) 65
Author's note on method and provenance 67
Note on sources and the literature 68
Companion specification 69
Appendix A — Items remaining open 70
Corrections carried forward from earlier drafts 70

1. Introduction

1.1 The problem

Consider an agent with a body. It has receptors that convert physical and chemical events into signal, and effectors that push back on the world. It falls when it missteps, meets an obstacle where it expected clear ground, and is answered by other bodies that may or may not do what it predicted. We want such an agent to do two things at once: to **enrich itself** — to grow the relational structure of what it knows from what its body meets — and to **audit itself** — to keep that growth honest. These two goals are in tension. The machinery that produces new internal structure is the same machinery that would have to certify it, and a faculty cannot reliably grade its own output when the fault to be caught lives in the grader.

Having a body does not dissolve this tension, and it is worth saying why, because the opposite is commonly assumed. The world is taken to correct an embodied agent automatically: it bruises, it resists, it will not be argued with. That is true of single collisions and false of the loop that reads them. A body supplies resistance; it does not supply an account of what the resistance meant. The agent that interprets the fall is the same agent whose interpretation caused the misstep, and the world, having pushed back once, does not go on to explain itself. Embodiment changes the *supply* of resistance, not the *instrument* that reads it — and the instrument is what the problem is about.

A body does, however, change the problem in one structural way that this paper is written around. In a purely informational environment the boundary between the agent and what is not the agent must be built out of information alone. With a body it is tempting to assume the boundary is given — that the skin, or the chassis, marks it. It does not. A body arrives with no self/environment line drawn on it. The line appears only when a command has been emitted and its consequence compared against what was already being sensed: what answered my command, and what did not. This is the first thing a body must do and the last thing it is credited with doing.

This paper asks: **what minimal set of conditions lets an embodied agent enrich itself, from what its body meets, without drifting into a self-confirming closed loop?** It answers by specifying a loop, its invariants, its layered contracts, the contract it requires of the platform beneath it, and — critically — the conditions under which the loop must be coupled to a source of resistance it does not control.

1.2 Why this is a specification, not an empirical paper

RSIL makes no empirical claim and reports no measured result. Its contribution is a design: a set of constraints such that any conforming implementation exhibits the properties

argued for here, and any non-conforming one collapses into an ordinary sensorimotor pipeline. Accordingly the paper follows the form of a conceptual specification — an academic framing (problem, positioning, scope, limitations) wrapped around a normative core written in the imperative register of a technical standard. Throughout the normative core, **MUST**, **MUST NOT**, and **SHOULD** carry their standards-document force: a MUST is a conformance condition whose violation forfeits the name RSIL; a SHOULD is a strong recommendation a designer may override only with reason.

The framing of this document as a *conceptual operating system* should be read in that light and not beyond it. What is specified is the layer between a platform's sensors and its actuators: the structure by which raw signal becomes structured information, and the conditions under which that structuring stays honest. It is not a runtime, not a control stack, not a middleware, and not a system anyone has built. It is called an operating system because it occupies that position in the architecture — everything above it presupposes it, and it presupposes only the platform — not because it ships.

1.3 Scope and what is deliberately left open

This document specifies structure: the dependency order of layers, the data contract into and out of each layer, the invariants every implementation must preserve, the two resistance regimes under which the loop runs, and what the loop requires of the body beneath it. It does not fix implementation constants that the source reasoning does not yet determine — sampling rates, tensor dimensions, numeric thresholds, hyperparameters, concrete data-type representations. Every such gap is marked `DECIDE@IMPL` rather than filled with an invented number. A specification that fabricates its own unfixed constants would trade honesty for the appearance of completeness; this one does not.

One limitation specific to embodiment is stated at the outset rather than deferred to §12, because a reader deciding whether this specification is of use to them should meet it early. A full-system snapshot covers the loop — its configuration, its layers, its field, its store — and does not cover the body. A platform compromised at the sensor or actuator level re-infects a freshly recovered system on the first cycle, and verifying sensor–actuator integrity is an operation this specification does not contain. The loop's event log can expose an anomalous pattern that signals such compromise; it cannot certify the hardware. The log is the symptom, not the examination.

1.4 Terminology note

Several terms below are coined or used in a stipulated sense (*resistance*, *the Other*, *Mode-A / Mode-B*, *registered mismatch*, *scar*, *forward-building*). §2 fixes them. Where a term names something for which ordinary English already has a loaded word — "consciousness," "experience," "imagination," "feeling" — the loaded word is avoided or explicitly re-defined,

because importing its connotations would smuggle in claims the specification does not make. This caution matters more here than in a disembodied specification: a document about a system with a body invites a reader to supply, unasked, a phenomenology the structure does not license.

1.5 What RSIL runs on: a protocol overlay, not a self-standing agent

RSIL describes operation, not genesis. It is a protocol overlay: it does not run in a vacuum and it is not a free-standing agent that bootstraps a self out of nothing. It REQUIRES a pre-existing host — any body satisfying the host-precondition set (§3) — to run on. The host is pre-existent: RSIL neither creates it nor generates a self from zero.

This matters for the start of the loop. The first action at cycle-0 — the one T2 needs in order to begin matching — is emitted by the host, which needs no self to give the first push, because the host is the source of action-emitting capacity: a non-egoic capacity RSIL uses but does not explain. That first (action, consequence) pair is what T2 reads to begin building the self/environment distinction. There is therefore no chicken-and-egg deadlock at the start: **the self is a product of the bootstrap action, not a precondition of it.**

When the mind runs off the body. RSIL holds no mind of its own. Where a body is driven by a mind that runs elsewhere — a controller off the body, running DIL (Companion specification) — RSIL is the body's part of one loop, not a second agent. It carries what physical data needs: transduction, source-differentiation at acquisition, the body's own state and record, and emission through the body. Integration, appraisal and memory are the mind's, under DIL. Because DIL and RSIL share one relational structure, the mind runs the two as one loop. There is then one host — the mind's — and each body is a set of its sensor and effector channels with an identity of its own. Several bodies are several limbs of that one host, not several agents.

2. Terminology

RSIL is built on relations, not things; the following terms are used in a stipulated, deliberately deflationary sense.

- **Physical world with others.** The target environment: a world containing two classes of thing that differ in kind — **other-entities** (organisms or agents), the source of other-resistance, which alone can feed T6–T8; and the **physical field** (inert objects, forces, the natural surroundings), the source of subjectless resistance, which feeds T1–T5 and T7.
- **Resistance.** The failure of a returned interaction to match the agent's prediction. With a body, resistance appears in two grades that must not be conflated: **physical resistance** — force, inertia, an obstacle, another body pushing back; and **informational mismatch** — what the agent sensed diverging from what it predicted. The two coincide at the body (a physical obstacle is also a failed prediction) without being identical: some purely informational mismatches, such as a voice contradicting the agent, carry no force. RSIL admits both. Resistance is a phenomenon of the running loop, not a property of the region alone: it arises at T5, where the mismatch between expectation and return becomes a signed `PredErr` (the `ResistEvent`).
- **Other-entity (the Other).** Any organism or agent the agent does not control, capable of mismatching its expectation in a way that may be intentional, may shift its stance, and may react to this agent specifically.
- **Subjectless resistance vs. other-resistance.** Two grades. *Subjectless resistance*: the region mismatches prediction but has no will of its own — an inert obstacle, a slope, a static heat source; consistent, objective, not reacting to this agent. *Other-resistance*: an other-entity mismatches in a possibly-intentional, stance-shifting, agent-specific way. Only other-resistance can build the concept of an independent Other.
- **Mode-A (endogenous / self-training).** The regime in which resistance comes from the inert physical world and the inertia of the data the agent already holds.
- **Mode-B (exogenous / live interaction).** The regime in which resistance comes from an other-entity the agent does not control.
- **Registered mismatch (`ResistEvent`).** A single recorded instance of the region failing to match the agent's expectation: the expectation built, where it missed, and the correction made. The atomic unit of experience (§10), as distinct from mere information.
- **Selfhood as process.** The thesis (§4) that the agent's identity across cycles is not a preserved content-core but the law generating the next state from the prior state; a dynamic invariant maintained only while the loop runs.

- **GLOB-MOD (global modulatory field).** A global state that reaches every layer — each layer receives the field's current value as a background condition on how it interprets its input — and that every layer contributes to as one competing parameter among many, never by exclusive overwrite. The direction is field-to-layer: the field conditions a layer; a layer does not reach up into the field to fetch a definition. This is field semantics, not shared-variable semantics: contributions blend and are re-weighted each cycle; there is no last-write-wins, hence no race condition between layers.

2.1 Other — definition and kinds

In RSIL, Other is what is not Self. Self is the law that operates at the current cycle (§4); Other is any source not identical with it. This includes the Self of an earlier cycle, which is no longer identical with the Self now operating — when the loop tests a present state against an expectation built from its own past (T5), that past stands to the present as an Other. Other is therefore **not a kind of entity but a relational position**: that-which-is-not-Self-right-now. Every other notion of Other in this specification is a sub-classification of this one, not a separate thing.

On this base, RSIL classifies Other along two independent axes.

Axis	Kinds	What it decides
Source	*external* (Mode-B — beyond the agent) / *internal* (Mode-A — the Self's own past and held data; and the `GeneralOther`, a model of Other built and held by the agent)	which mode obtains
Subjecthood	*reactive* (it answers the agent — may shift stance, may respond to this agent specifically) / *subjectless* (it resists but has no will of its own — a physical regularity, the inertia of held data)	whether T6 lives or atrophies

A **reactive external Other** yields the strongest evidence of independence and is what feeds T6. **Subjectless resistance** feeds T1–T5 and T7 but cannot on its own build the concept of an independent Other, so it does not feed T6. The **`GeneralOther`** is a representation, not a source of fresh resistance. **`STRANGER` and an `entity_id`'d Other are not separate kinds** but two recognition-states of one specific Other — unmatched against the identity store, or matched to a known profile. None of these classifications adds an entity; each only locates a region of the single relation *Other = not-Self*.

2.2 Host, body, and platform — three terms kept apart

Three terms in this document name things a reader may collapse into one, and the collapse would cost several distinctions the specification depends on.

- **Host (the substrate).** The capable existent that MUST satisfy E1–E4 in order to run RSIL. It exists even when no loop runs. It is substrate, not self: it carries no self and contributes no self. What it supplies is non-egoic — the bare capacity to emit a first action — which the protocol uses but does not derive. While initializing (running T1 and emitting the bootstrap action at T2 cycle-0) it is still only the host; no separate label is needed for "a host that is running."
- In RSIL the host is literally a physical body — or, where the mind runs off the body (§1.5), a host whose physical part is one or more bodies, each declared to it as sensor and effector channels. A body then holds its own state and its own record, never the mind's memory. But precisely because the sense is literal, one easy misreading must be blocked here: **a body does not hand the agent a ready-made self/environment line.** Having a body does not mean having a boundary. The boundary appears only when a command has been emitted and its consequence compared (§4).
- **Body.** The physical thing that senses and acts: the host under the description that makes its physicality salient. Where this document says *body* rather than *host*, what is at issue is that it has mass, occupies space, can be damaged, and can be interfered with at the sensor or actuator level.
- **Platform.** The sensor-and-actuator assembly considered as an engineering object — what an implementer procures, wires, and is responsible for. §8 states what RSIL requires of it. The distinction from *body* is one of description, not of thing: *body* is the term when the point is that the agent is embodied; *platform* is the term when the point is that some party must build, verify, and maintain that embodiment.
- **Agent.** The host running inside RSIL, counting from T2 cycle-0 — i.e. from the step at which a self exists. Compactly: **agent = host that has acquired a self.** It is not a third tier wedged between host and self; it is that same host, under the description "running the loop and now carrying a self."

Two further terms name what the agent interacts with, and are distinct from the host/body/agent set:

- **Region (the interaction field).** That which the agent exchanges input and output with. Neutral as to inside/outside: under Mode-B the region includes an external Other; under Mode-A it includes the inert physical world and the agent's own held data. Different from the host: the host is what runs the agent (the ground); the region is what the agent interacts with (the I/O partner).
- **Counter-source.** That which the agent's lens predicts against and may fail to match (E2). Counter-source $\subset$ region, in both modes: it is the part of the region currently playing the resistance role. E2 requires only that one non-omnicompliant counter-

source exist *within* the region — it does not require that the counter-source lie outside the agent.

On `AgencyTag`. RSIL writes `SELF_CAUSED` where DIL writes `SELF_WRITTEN`. The divergence is deliberate and tracks the substrate: in a purely informational environment the agent's own change is something it *wrote*; with a body it is something it *caused*, through a physical action with a physical consequence. `ENV_PUSHED` is unchanged in both, because what the environment does to the system is the same relation either way.

3. The host environment: definition by operating conditions

A host adequate to run RSIL is not identified by its material — "biological / neurons / muscle" is material, not structure. It is identified by four structural properties it MUST possess. Any host possessing them can run the loop; any host lacking even one cannot.

- **E1 — Differentiability.** The host MUST **permit** (make possible) a distinction between "inside the agent" (the body and its internal state) and "outside the agent" (stimulus arriving from the environment). It does not itself construct that distinction: **the host permits, the agent builds.** Construction is the agent's own work at T1/T2 — the self/environment difference is an *output of T2*, not a property the host hands over ready-made; T1 only confirms that an activity-environment is present. Consequence: if T2 fails to crystallize a first-person self/environment difference, the host has still done its part — there is simply no agent yet, and the loop fails cleanly at T1/T2, the way an engine that never starts is not the same as one that breaks mid-run.
- **E2 — Interaction.** The host MUST be able to return a response when the agent emits an action — and MUST be **non-omnicompliant**: among those returns, it MUST be able to fail to match the agent's prediction.
- Two distinct failures lie below this one condition, and they MUST NOT be conflated. A region that **returns nothing at all** — a void field, however densely populated with not-Self objects that never respond — fails E2 outright: **the loop does not start.** Presiding over a room of the unresponsive is no reign. A region that **does return, but in which the agent's every guess is correct**, passes the interaction threshold yet offers nothing to learn from — **the loop runs but does not advance.** The threshold for E2 is minimal but non-zero: **at least one return other than silence.** A single silence, set against a background of returns, is itself a valid mismatch (registered at T7); unbroken silence is a void field, not a mismatch.
- **A rank distinction that must be held.** E2 tests the **host and body** — can it return anything at all. **Resistance is a phenomenon of the running loop**, not a condition on the environment: it arises at T5, where the gap between the agent's expectation and what was returned becomes a signed `PredErr` (a `ResistEvent`). The environment need only be able to return; whether a given return resists is the loop's business.
- **A further rank caution.** "Has resistance" is measured *relative to the agent's predictions*. A model-poor agent finds everything resistant; an overfit agent finds nothing resistant. E2 is therefore a condition on the **pair (agent, region)**, not on the region in isolation. One cannot certify a region as "sufficiently E2" without first fixing the agent placed in it.
- **E3 — Temporal accumulation.** The host MUST provide a sequence of cycles in which the history of interaction can be inscribed. A one-shot region grants no accumulated

history.

- **E4 — Observable behavioral projection.** The agent MUST act upon the region — through bodily response, link ⑤ — in a way a third party can read; otherwise the loop's success is not externally measurable.
- **E4 constrains the agent, not the third party.** It requires only that the agent leave a readable trace; it says nothing about who the third party is or whether it can be compromised. **Two roles MUST be kept apart:** a **recording** third party (attributes continuity and distinctness; needs only a readable trace) and a **judging** third party (evaluates correctness; needs genuine independence). E4 requires only the recording role; the judging role is treated separately (§13).

A host satisfying E1–E4 can run RSIL **regardless of what the body is made of** — biological or artificial. One that does not, cannot.

> **Two grades of resistance — the ground for T6.** *Subjectless resistance* (an inert obstacle, a slope, a static heat source) is consistent and objective but never reacts to this agent; it suffices to feed T1–T5 and T7. *Other-resistance* (an organism or agent that may be intentional, may shift stance, may react to this agent specifically) is the **sole** source from which the concept of an independent Other (T6, `independence_evidence`) can be built. Consequence: a region offering only subjectless resistance — an empty room full of inert objects — feeds T1–T5 and T7 but cannot fully feed T6–T8; a reactive Other is required for T6 to live. This is the Mode-A / Mode-B boundary of §9.

Architectural premises

- **P1.** What operates is a **relation**, not a thing. Meaning lives in correlation, not in the isolated signal.
- **P2. Material neutrality — the substrate is fixed as a physical body; the material is left open.** RSIL's substrate **is fixed as a physical body**, with physical or chemical receptors at the input end and muscular, mechanical or endocrine response at the output end. What remains open is only the *material* of that body: biological (neurons, muscle) or artificial (sensors, actuators). P2 therefore does **not** claim "runs on any substrate whatever"; it makes the narrower and more determinate claim: *no particular material is required of the body, provided a physical body satisfies E1–E4*. RSIL's scope is the physical world with a body; a system without a body, in a purely informational environment, lies outside it — that is DIL's scope.
- **P3.** The success criterion is **whether the loop runs** — measured by externally observable traces (§9).

- **P4.** Layer order is forced by **dependency relations**, not by signal strength. Each layer unlocks meaning for the next.

The host-precondition set and the bootstrap of the loop

E1–E4 say what the environment must afford the loop. One further thing must be said explicitly, because it is the start condition the loop cannot supply for itself: **the body MUST be able to emit a first action at cycle-0** to set T2 in motion.

Beyond E1–E4, a viable body minimally satisfies a precondition set **P**: (a) it can emit an action **distinguishable from ambient regional fluctuation** — else T2 never rises to a first match; (b) it can **hold state across cycles** so history accrues (INV-5 requires accrual, not loading); (c) it can **withstand resistance without rewriting itself clean** on every mismatch — else no scar survives long enough to matter.

The body emitting the cycle-0 action is still only the host (it needs no prior self), and that action is not pre-loaded history — so INV-5 is untouched: **emitting a fresh action is not loading a past one.** Start-up is thus the body's responsibility, not the loop's; if the body cannot meet P, the loop simply does not start — a clean non-start, not a mid-run failure.

4. Selfhood in an embodied system

RSIL *has* a body, but **the body does not hand the agent a ready-made self/environment line at the input.** T1 only takes in raw spatial data from the sensors about the activity-environment; it has drawn no self/environment boundary, because until an action has been emitted there is nothing against which to compare "what answered my command" with "what did not." That boundary appears only at T2, when the agent emits a motor command and matches the consequence against the spatial data T1 has built: **the self/environment polarization is a product of agency at T2, not a property T1 hands over.**

And having a body does not exempt RSIL from the question of the self's *continuity*: it does not answer whether the self at cycle N is the same self as at cycle N−1.

The self is a kind-preserving process, not an invariant core. The content of the self changes every cycle — the boundary, the expectations, the baselines are all updated, and the physical structure of the body itself changes over time. What does not change is no carried-over fragment of content but **the law that generates the next state from the prior one**. The agent at cycle N is the same agent as at cycle N−1 not because they share any content, but because an unbroken causal line links the cycles through exactly that law. This is a **dynamic invariant**: maintained by the loop running, not by standing outside the running. Stop the loop and the axis is lost. Having a body does not rescue this point: a body still lying there after the loop has stopped is not the same self — it is matter, not process.

Where the self begins, and that it is a flow rather than a milestone. The self does not pre-exist T2 and is not pre-loaded from the body. It is initialized at T2 cycle-0 — through the match between the just-emitted action and its observed consequence, which is what first erects the self/environment distinction — and then re-localized through T2 every cycle, thickening with each cycle's history. The self is therefore a **flow** (a verb), not a **milestone** (a noun): this is why it can degrade as the loop weakens (§9), and why its continuity must be anchored by a third party reading the trace.

This is the honest boundary of the claim, and it has two parts. **First**, RSIL does not assert that it has closed the question of how a self first arises; it asserts only that, given a body that can take the first step, the protocol structures a self upon it. **Second**, the capacity to take that first step — the body's bare, non-egoic action-emitting capacity — is likewise presupposed, not explained. Both the origin of the self and the origin of the first-action capacity sit outside what this specification undertakes to derive.

The self's continuity is not measurable from within. The agent has no access to its own time-derivative: it cannot rewind to subtract "self at cycle N" from "self at cycle N−50." The specification MUST therefore keep two measurements distinct:

- **Distinctness** (this self ≠ that self) **is measurable**: given identical present input, differing behavior yields a difference that is the signature of accumulated history — a differential measurement requiring ≥2 agents or 2 slices to subtract.
- **Continuity** (same self across time) **is not measurable from within**. It is a third-party attribution laid over the stored behavior-record.

Conflating these two is an error. This distinction is the foundation for the limit on C5 (§9) and for the embedding argument of §9: **a self-auditing loop cannot catch the moment of its own drift or stop, because the instrument it would inspect with is the thing running.** The recognition-condition must therefore anchor **outside** the loop — in a stored trace, read by a third party. This is precisely why the present work calls its agents **audit-ready**, not **self-auditing**: the loop cannot audit itself, but it can — and by R1 (§10) does — build the external-facing trace by which a third party audits it.

5. System invariants

Every implementation MUST preserve all of the following. Violating any one yields not RSIL but an ordinary sensorimotor pipeline. The invariants are the **absolute conditions of the loop**: they are not data inside the store and carry no tags; they cannot be overwritten by anything in the loop, because to overwrite one is not to edit a datum but to **halt the loop** — to end the mismatch on which the self depends.

ID	Invariant	Rationale
INV-1	**Closed loop.** Every layer output MUST have a *path* back to some layer's input. No dead branches.	The loop feeds its own input.
INV-2	**Output-register identity.** Every output is tagged `INFO`. No layer may promote a correlation (↔) to an identity (=).	Preserve the correlational register; see T8-INV.
INV-3	**One-directional dependency (meaning-channel).** On the meaning-channel, layer N consumes only outputs of layers ≤ N.	Dependency order (P4). *Out of scope:* the modulatory field (INV-7) acts downward onto layers and is not a meaning-channel dependency at all.
INV-4	**Meaning = relation.** No layer assigns meaning to a signal in isolation; meaning is a function of (signal, lower-layer context).	P1.
INV-5	**History accrues, is not loaded.** Temporal state forms only through sequential accumulation, never pre-loaded.	The self is a process maintained through running (§4).
INV-6	**Agency-gate.** Every change MUST be classified as caused-by-me / not-caused-by-me (the self/environment difference) *before* interpretation.	The condition for an external sensation to be meaningful.
INV-7	**GLOB-MOD.** A global state that reaches every layer (field-to-layer: each layer receives it as a background condition, none reaches up into it) and that every layer contributes to as one competing parameter among many (field semantics: contributions blend and are re-weighted each cycle, never last-write-wins; a layer's contribution at cycle N conditions the field only from cycle N+1, never within the same cycle).	Same data + different field → different meaning.
INV-8	**Appraisal step.** Between integration ③ and response ⑤ there MUST be an appraisal step ④ assigning direction and value. Step ④ **MUST NOT draw its criteria from the very state the agent is editing** — else it is self-scoring and hackable.	Against bare reflex; anti-drift anchor of Mode-A (§9).

On INV-2 — committing to an action is not promoting ↔ to =. Acting on a correlation is not the same as freezing it into an identity. To respond, the agent commits to one action (it pushes this object, it steps that way); this commitment is itself a ↔ operation — a **revisable** best-current-guess, read against the consequence next

cycle — not an identity claim. INV-2 forbids *freezing* a correlation into a fixed truth; it does not forbid *acting* on one. What would cause paralysis (infinite hesitation) is demanding = before acting — "be certain X is true before moving"; INV-2 forbids exactly that demand, and so **frees** the agent to act on ↔ and learn, rather than causing hesitation.

On the invariants and GLOB-MOD — a level distinction. GLOB-MOD modulates the **information** flowing through the layers (T1–T8); it does not and cannot alter the **invariants** themselves. The invariants are *laws about* the loop; GLOB-MOD is a *parameter within* it. No contamination of the field can rewrite a law — in particular, a drifting lens writing into GLOB-MOD cannot thereby edit the criteria of INV-8, because those criteria are not information-in-the-loop but a law about it, and are anchored where the agent cannot edit them (§9).

On INV-1 — a topology constraint, on cycle-time not wall-clock. INV-1 requires that every output *have a path back* — that a return path **exist in the loop's topology** — not that every packet completes its return on every cycle. The loop advances in **cycle-time, not wall-clock time**: idle stretches between cycles are the default, not events to be declared. A sleeping body, or an Other falling silent, does not violate INV-1 (the path still exists) and does not drop output into the void; the next cycle simply has not yet occurred. If an Other disappears, the loop does not break — it falls back to Mode-A (digesting collisions already taken, against held data), the path still closed through the internal route; and if no Other returns, it drifts and decays (§9), which is a specified mode transition, not a broken invariant.

Resolving the INV-3 ↔ INV-7 tension (meaning-channel vs. modulatory-field). The system has **two channels of different kind**, and the apparent tension dissolves once their directions are stated correctly. The *meaning-channel* carries `InfoUnits` **up** the layers, one-directionally, subject to INV-3 — the path by which one layer defines content for the next. The *modulatory-field* is GLOB-MOD: a global state acting **downward** onto every layer at once. The crucial point is directional: when an upper layer (e.g. T8) alters GLOB-MOD, it changes the state of the *field*, and the field then conditions every layer from above; a lower layer (T4) does not reach up the layers to fetch anything from T8. T4 receives a background field-value, exactly as it receives the ambient field on any cycle. Hence **INV-3 is not even engaged.**

And there is no timing by which the tension could re-enter: field-conditioning takes effect at cycle **N+1**, never within-cycle. This forecloses both misreadings at once. A within-cycle reverse coupling — which would create an algebraic loop in the dependency graph — does not exist, because no field write is read in the cycle that produced it. And the across-cycle path that does exist (cycle N's field shaping cycle

N+1) is not a violation but exactly INV-1. **The two channels MUST be implemented separately.**

On the semantics of GLOB-MOD (against mis-reading as a shared mutable variable). GLOB-MOD has **field semantics, not shared-variable semantics**. A "contribution" from a layer is not an overwrite of a shared cell: contributions blend, compete, and are re-weighted by the next cycle; no layer's write wins exclusivity. There is therefore **no last-write-wins and no inter-layer race condition** — concurrency of contribution is a non-issue.

No gain ceiling is needed — the block is structural, not probabilistic. One misreading must be set aside: that the field's strength is a free dial an implementer must cap to keep the field from overpowering the loop. It is not, and the reason is structural. GLOB-MOD is not an outside force pressing down on the layers; it is **continuously constituted by them** — each layer feeds the field as the loop's enriched information circulates (the field acts downward onto the layers, and the layers, independently but simultaneously, shape the field in return). This two-way coupling means the field **cannot bootstrap itself past the layers that feed it**: for the field to overpower the layers it would have to be made strong by the very layers it is overpowering, which the coupling forbids. A field running away from this mutual check by its own internal dynamics is therefore **structurally precluded — ruled out by construction, not merely rendered improbable.** No gain cap is needed, because the architecture already prevents it.

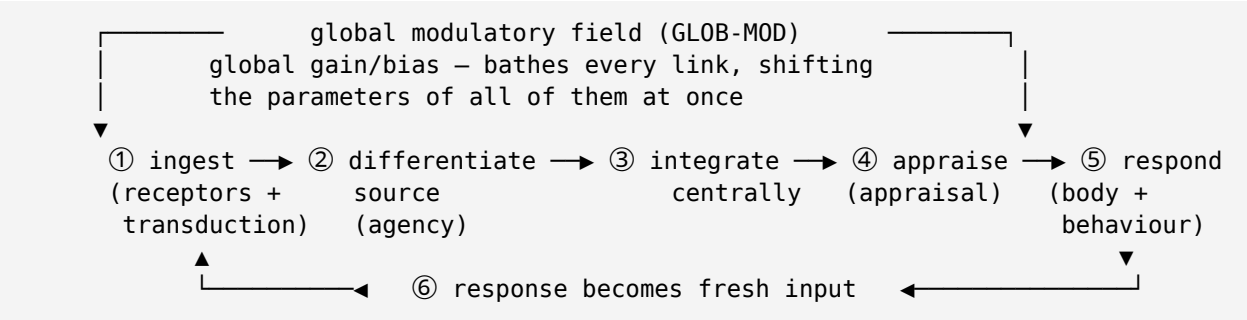
What lies past the modulatory threshold. If the field ever did overpower the loop: by its own name GLOB-MOD is *modulatory* — it biases how each layer interprets its input. A field strong enough not merely to bias but to **determine** every layer's interpretation would leave the layers emitting only what the field already dictates — no layer reading its own input, no mismatch, no `PredErr`, nothing for the next cycle to process differently. The output would stop changing across cycles: a frozen identity, an =. That is not a degraded loop (Mode-A, which still turns and still emits a live ↔); it is a **halted** one — and per T8-INV an = is precisely the sign that cognition has stopped. Crossing that threshold does not yield a worse RSIL; it yields something that, having halted, has **forfeited the name**.

But as the coupling above shows, the loop **cannot reach that threshold from within**. The only path that crosses it is an **external flood**: a volume of input arriving through the sensory channels that exceeds the processing capacity of every layer *and* of the field at once, saturating them in lock-step so that the mutual check has nothing left to push back with. That is the **Sybil case** (§12, third limitation; §13). It is not defended by any internal gain cap, because its origin lies outside the field's reach. What RSIL itself

carries is only the **early symptom** — a diversity-loss signal across its resistance sources — not a cure.

6. The canonical loop

The pipeline below is the core of which the full layer architecture (§7) is a detailed implementation. Every layer maps onto one link of this loop.



The six sequential links

#	Link	Nature in an embodied system
①	**Ingest**	Specialized receptors take up physical or chemical stimulus, **and** transduction: the stimulus *becomes* a transmissible signal. Input comprises the consequence of a just-emitted action, stimulus from the environment, and signal from an other-entity.
②	**Differentiate source**	Tag caused-by-me / not-caused-by-me *before* interpreting (the agency-gate, INV-6).
③	**Integrate centrally**	Synthesize across sources into a new form of information.
④	**Appraise**	Assign good/bad-for-goal → information becomes directional. The anti-drift link (§9).
⑤	**Respond**	The agent writes to state and/or emits to the region: endocrine or muscular response, behaviour.
⑥	**Feedback**	Response ⑤ becomes the new input to ① → **loop**.

A distinction to preserve (against mis-drawing). Transmission at a junction — the synapse in a biological body — is part of ①: local, one stage. **GLOB-MOD** is a *field* enveloping the whole loop. The two must not be merged into a single "chemical step": they differ in kind, one being a node and the other a global parameter.

On the ordering of links ① and ②. Transduction is folded into ① — it is *how* a stimulus gets in, not a separate stage of its own rank — and ② is given to source-differentiation, matching DIL. The alternative arrangement, in which transduction occupies ② and source-differentiation has no link of its own, leaves a first-rank invariant (INV-6) without a place in the canonical loop, and is rejected for that reason.

The global modulatory field (a FIELD condition, not a link)

The whole loop is **bathed** in a global state carrying content — *alert*, *safe*, *urgent* — that changes how every link interprets. This field has no position in the chain: it is a global gain/bias shifting the parameters of every link at once. Consequence: the same signal at ②, bathed in a different field, integrates (③) differently, is appraised (④) differently, and is responded to (⑤) differently.

> **Material neutrality (per P2).** What is specified is a *role* — a global modulatory field. *How it is realized* depends on the material of the body: in a biological body, a chemical background (neuromodulators — dopamine, serotonin, oxytocin, cortisol); in an artificial body, a parameter or data field every layer reads. The two are instances of one role.

What GLOB-MOD is

What it is. GLOB-MOD is a **global interpretive prior**: a single background state, shared by every layer in a given cycle, that sets the disposition with which each layer reads its input. It is not content that flows through the layers (that is the meaning-channel); it is the standing bias by which whatever flows is interpreted. The relation is that of light to the objects in a room: the light is not one of the objects, yet it determines how every object appears at once, and a change in the light changes the look of everything without altering any object's shape. The term *interpretive prior* is used deliberately in place of any richer label — mood, affect, attention — because those import commitments (felt quality, consciousness, a specific weighting mechanism) that a material-neutral specification must not assume. What is retained is only the operative core: a global disposition that conditions reading.

What it carries. The `params` of `ModField` are a set of **scalar interpretive biases** — each modulating the gain or threshold of some layer operation: how readily a source is trusted, how alert the loop is to mismatch, how far it leans toward exploring rather than consolidating. They carry **how to read, never what is read**: no bias holds propositional content, which is what keeps GLOB-MOD from ever becoming a back-door content channel and so preserves INV-3. The number of axes and their specific nature are `DECIDE@IMPL`; what is fixed is their *kind*.

Their origin is dynamic: at cycle-0 the prior is seeded from the body's existing data and state, and from cycle-1 onward it is continually re-weighted by T8 feedback (INV-7, field-to-layer, effective at cycle N+1).

> **A point of precision about cycle-0**, to avoid an apparent paradox — a prior that interprets before there is a self to interpret for. What is seeded at cycle-0 is strictly a **host-bias**, not yet the agent's. It is not an interpretive prior *of the agent* at that moment, because the self that the prior would be a prior *for* has not yet crystallized

(T2 cycle-0). It **becomes** the agent's interpretive prior only from cycle-1, once the self is in place to be conditioned by it — the same single event, viewed from the field side, as the self coming into existence.

Why it is needed. Two reasons, of different weight. First and sufficient — **context-conditioning**: without a global prior, each layer would read its input bare, with no dependence on the state the system is in. The same `InfoUnit` would be interpreted identically whether the loop had just met a hostile Other or a neutral one. Second and lighter — **context-coherence across layers**: because the field is global and every layer is bathed in it within one cycle, the eight layers share a single interpretive disposition on that cycle rather than each reading in private. This second reason is stated at low strength on purpose: GLOB-MOD contributes to the coherence of the agent's stance; it does not by itself constitute selfhood.

Note what is **not** on this list: **GLOB-MOD does not resist drift.** Being global, it is the channel through which a biased disposition would spread to every layer at once; it is where drift propagates, not what guards against it. It carries the very parameter that drift attacks — receptivity to mismatch — but it does not protect that parameter. That protection can only come from outside the loop (§9, Mode-B).

Information runs in many directions at once, and which directions is not fixed in advance

The loop is not a single pipe, nor a fixed pair of pipes. At each cycle information moves along several paths simultaneously: the default return — T8's product becoming the next cycle's input at ① (link ⑥) — runs together with feedback that may reach T4, or T5, or T6, or several of them at once, through the modulatory field. **Which paths are live is set by the content and character of the information itself, not by a fixed wiring diagram.** Information about the Other-in-general settles where that frame is built; a relational-value signal settles where relational models are held; the routing follows what the information is.

Two consequences follow, and both dissolve paradoxes that arise only from reading the loop as a one-way pipeline.

First — no layer is a terminal sink. T8 is not an endpoint: its product returns both by the cycle-closing default (link ⑥) and by content-routed feedback through the field. INV-1's requirement is a return *path for information*, not a claim that data never crosses the system's boundary — so emitting an output to the region (link ⑤, E4) and closing the loop (INV-1) are not in tension: the output leaves as data *and also* returns as information.

> **INV-1 does not entail E2.** What returns is not guaranteed to carry a mismatch: closure (INV-1) routes the output back, but *whether the returned information resists* is the separate business of the counter-source (E2), not a property the loop's topology

can supply. A region that merely echoes the agent's output back unaltered satisfies INV-1 and still offers nothing to learn from. This is exactly why a closed loop is not thereby a non-degenerate one — the whole burden of §9.

Second — T6's atrophy under Mode-A starves nothing, and calling it "starvation" inverts the actual failure. Under Mode-A information does not run short: every path stays live and the loop circulates as fully as ever. What is missing is not volume but **novelty** — with no independent Other, everything moving through those paths is a variant of what the system already holds. The loop eats its fill and digests itself. Mode-A T6 is thus a **standing quality-ceiling reached by saturation-without-renewal, not a cascading shortage**: the failure mode of a self-enriching loop is never running out of fuel, it is running out of **resistance**. This is why only a live Other (Mode-B) relieves it and why taking in more of the inert world cannot; and why Mode-A, though it never collapses for want of throughput, is still not durable — digesting only itself, its structure thins cycle by cycle until it drifts and fails.

Mapping the loop onto the layers

① belongs to T1 and T3 (spatial data intake, activity-environment confirmation, channels) · ② belongs to T2 (agency) · ③ runs from T3 through T8 (integration, climbing the relational tiers) · ④ is present at T5 (prediction error) and T8 (`RelValue` as comparative appraisal) · ⑤ is a **lateral capability** invoked by several layers (§6.1) · ⑥ is enforced by INV-1.

6.1 Emission: link ⑤ as a lateral capability

Link ⑤ sits in the six-link table above as one node among six, and this placement, while correct for the pipeline diagram, understates what link ⑤ is. The four ingest-to-appraise links (①–④) run in a fixed dependency order — each consumes the product of the one below it, on the one-directional meaning-channel (INV-3). **Link ⑤ does not belong to that order in the same way.** Emission is not a station the loop passes through once per cycle on its way from appraisal to feedback; it is a **capacity the loop exercises from many points**, whenever a layer's own work requires pushing something out to the region.

This matters more in an embodied system than in a disembodied one, because a body's emissions are physical acts with physical cost. A reader who takes the agent to act once per cycle, at a single terminal step, will design a platform that can only do so.

Where the layer contracts already presuppose emission. Reading §7 with emission in view, the dependence is visible in the contracts themselves, not added to them:

- **T2 cannot draw the agency line without an action already emitted.** Its input row lists the motor command just emitted; its mechanism matches that command against

the observed change. The agency-gate (INV-6) therefore presupposes a prior emission — a probe whose only purpose is to produce the self-caused change T2 reads. Without something emitted, T2 has nothing to match, and the self/environment difference is never drawn.

- **T3's channels include the results of the agent's own actions** — returns whose existence requires a prior emission.
- **T5 builds an `Expectation` and can only test it by an emission** whose return either confirms or violates the prediction; the signed `PredErr` where resistance becomes information often requires the agent to *act* to see whether the region returns what was expected.
- **T6 accrues `independence_evidence` only from a resistance collision** — and eliciting that collision, checking whether an Other resists as independently as modelled, is itself an emission.

These are not four new powers to be granted. They are **one capacity** — emission — that the four contracts already lean on.

Emission is lateral, as GLOB-MOD is lateral — but opposite in direction. The specification already contains one thing that is not a link in the chain yet touches every layer: the modulatory field (INV-7), which descends onto every layer as a background condition. Emission is its structural mirror. The field reaches **down** into every layer as incoming condition; emission projects **out** from whichever layer invokes it. Neither is a station on the meaning-channel; both are lateral to it.

What every emission carries, and why it cannot be otherwise. Whatever layer issues it, an emitted action inherits four constraints from invariants already in force — this subsection introduces no new law:

- **Its register is ↔, never =** (INV-2): a committed action is a revisable best-current-guess, read against the next cycle's consequence, not a frozen identity.
- **It leaves an externally readable trace** (E4): every emission leaves a mark in the audit plane, recorded as one activity record per cycle (§10) — trace, not experience: no layer learns from it, and it never becomes a scar by accumulation.
- **It is readable back by T2 at the next cycle** as "the command just emitted," closing the agency path: emit → region returns → T1/T3 ingest → T2 matches. An emission unreadable in this way would leave the agency-gate (INV-6) unable to classify the resulting change.
- **Its correctness is never scored by the emitting layer.** It is judged only by the next cycle's return — a matching return leaves only the activity record; a mismatching return enters T5/T7 as a signed `PredErr`, becomes a `ResistEvent`, and holds as a scar

— and, later and from outside, by a third party reading that scar. A layer that scored its own emission would be self-scoring, the exact fault INV-8 exists to forbid.

Conflict among emissions is not arbitrated; it is collided. A reader may ask what happens when two layers require emissions the single body cannot jointly satisfy — T2 needing a probe in one direction while T3 needs an orienting movement in another. With one body this is not a hypothetical: a body has one position, one posture, one set of limbs, and cannot execute two incompatible commands at once.

The specification's answer is not a coordinating layer or a priority rule; it is the same answer §9 gives for resistance generally. **When opposing emissions cannot both be satisfied, the opposition is emitted and met by the region**, which returns something that mismatches at least one layer's expectation — and that mismatch is a `ResistEvent`. The arbiter is not an internal referee but **the region returning resistance**; the next cycle carries the scar of that collision and emits differently. To place an internal arbiter here — a step that ranked emissions against a stored criterion before acting — would be to build exactly the self-scoring appraisal INV-8 prohibits. The loop does not adjudicate its own actions internally; it acts, collides, and reads the collision. This is INV-8's discipline carried into the domain of action: **let the outside brake.**

6.2 Forward-building: the two positions between holding data and emitting

§6.1 established emission as a lateral capacity. A second thing the layer contracts already lean on is what happens **between holding a datum and emitting on it**. T5 builds an `Expectation` before it can register a `PredErr`; T6 elicits a collision by putting a model to the test; T2 emits a probe whose consequence it will read. Each presupposes that the loop has, before acting, built something forward from the store it holds. This subsection names the two positions that motion occupies, because the store (§10) must be able to record them and an auditor must be able to read them.

The two moments. In the first, the loop **builds a situation**: a circumstance it may be in, assembled from the data it holds. A chair is before it; what the loop builds is the chair as old or new, its material, the ground beneath it, the space around it. Nothing is yet concluded. In the second, an **outcome is cast** from that situation: sitting will hold, or will not. The two are distinct, and collapsing them loses the distinction the store needs — a situation built is not yet a prediction, and a prediction is not a fresh situation.

Why this needs no separate model of the region. A reader may object that building a situation requires a model of the region rich enough to run internally, which the specification nowhere grants. The objection mistakes what is being built. **The loop does not simulate the region; it builds itself in a circumstance**, and for that it needs no separate model, because it is itself already running. What is required is not a replica to

execute but the store it already holds. The question "is the model rich enough" does not arise: there is no second model.

Why this is not an internal arbiter. A second objection is sharper. If several situations are built and one is acted on, something has chosen among them — which is the internal arbiter §6.1 refused. The answer is that **nothing scores them**. A situation carries because it **fits the store better** than the others, in the way competing contributions to the modulatory field blend and re-weight (INV-7) rather than one command defeating another. Fit is not a verdict. A scoring standard would have to come from somewhere: from outside the loop, in which case it is foreign criteria imposed on appraisal, or from the state the agent is editing, in which case it is the self-scoring INV-8 exists to forbid. Comparison of fit requires neither. This is also why the leaning of a store is not corrected here: a situation built from a store that leans one way leans the same way, and the loop has no instrument for detecting that lean, the instrument being the store itself.

What forward-building buys, and what it does not. It does **not** let the agent avoid collision. Its purchase is narrower and worth stating exactly: **it changes the posture in which the agent meets the collision.** Having built the chair as possibly-old, the agent sits testingly and staggers; having built nothing, it sits committingly and falls. The collision occurs in both cases; what differs is what it costs.

This bounds the benefit honestly, and with a body the bound has teeth. Where a cheap test is available — sitting lightly, throwing a stone ahead, putting weight on gradually — forward-building converts a costly collision into an affordable one. **Where the cheapest available test is itself fatal, it does not help**, and the agent's safety, if it has any, comes from elsewhere: from a scar it did not earn itself, read from a shared store or left by another agent (§10, R2). Forward-building widens the margin; it does not abolish it.

What an observer would call this. Ordinary language has a word for a loop building situations and casting outcomes from them, and the word is *imagination*. The specification does not use it, for the reason §1.4 gives: the word carries connotations — of a faculty possessed, of images entertained, of a capacity distinguishing one kind of creature from another — that the structure does not license. What the structure describes is a datum passing through two positions. Nothing about those positions is proprietary to any material or species, and RSIL, being material-neutral as to the body (P2), is in no position to award them to one. The corollary is epistemic and worth stating, because it cuts against a common form of claim: **whether a given body builds forward is not settled by inspecting what it is, but by whether its transitions leave traces a third party can read** (E4). A body that builds forward and leaves no trace is one about which this specification says nothing — not one about which it says no.

Likewise, the retrospective vocabulary that attaches to such action — *I thought the ground was solid* — resolves into two things the specification keeps apart. **The ground was solid** is a datum, and the failure it names is that the datum went straight to emission without being taken up into either forward position. **I thought** is not a state the loop held while acting; at the moment of acting there was a datum and an emission and nothing else. It is a name applied afterwards, by a reader of the trace, to an action emitted on a datum that had passed through neither position. The store records the first and has no vocabulary for the second, which is as it should be: **the trace is the FACT, and the retrospective name belongs to whoever reads it.**

7. Layer architecture and data contracts

7.1 Shared types

Abstract notation; concrete representation is `DECIDE@IMPL`.

> **Register glossary — neither symbol is overloaded, fixed once here.** ↔ = a live, revisable correlation — the register of every running output, including action-commitments. = = a frozen identity claim, "X is Y, exclusively, correlation closed"; per T8-INV it appears only when the loop has **stopped**. Committing to an action writes a ↔, never a =. These two are kept strictly apart throughout (INV-2, T8-INV, §9).

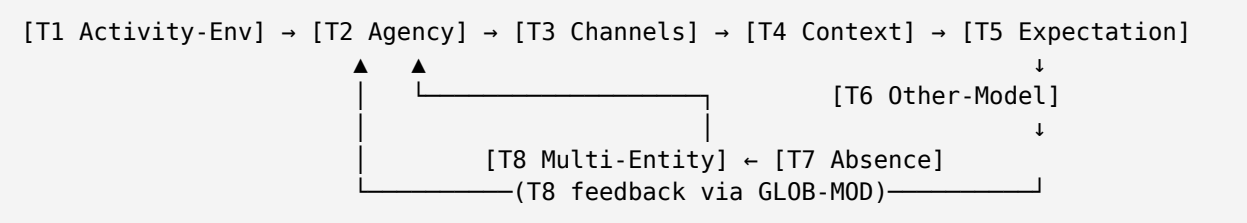
```
Signal      := { source_id, raw_payload, t }          // raw signal, not yet
meaning
InfoUnit    := { content, ref_frame, t, tags }        // information = referred to
a frame
ActivityEnvironment := InfoUnit  // confirmation that an activity-environment is
present;
                                 // NOT yet polarized into self / environment
AgencyTag   := enum { SELF_CAUSED, ENV_PUSHED, UNDECIDED }
Expectation := { predicted: InfoUnit, confidence, built_from: history_window }
PredErr     := { observed, predicted, delta, signed }   // signed: +/- (absence =
negative)
OtherModel  := { entity_id, context_map, independence_evidence }
RelValue    := { entity_id, relative_rank, comparison_basis }  // exists only when
N>=2
SocialEdge  := { a_id, b_id, observed_interaction }   // Other<->Other, no self
present
ModField    := { params, t }     // GLOB-MOD: global modulatory field (INV-7)
Appraisal   := { info_ref, valence, goal_relevance }  // output of appraisal step
④
ResistEvent := { source_id, expected, received, mismatch_kind, t }  // a MISMATCH
(E2)
```

Contract convention. Every `InfoUnit` leaving a layer MUST have `ref_frame ≠ null`. A `Signal` without a `ref_frame` is not yet information — this is where INV-4 is enforced at the type level.

> **On `provenance`.** `provenance` is **one of five positions** (`prior` / `running` / `simulated` / `projected` / `scar`). A datum bears **exactly one** at any moment, and the tag **does not accumulate**. The reason: a label that grew a history would be a label carrying a log — two different kinds of thing housed in one slot. A datum's path, through positions and through layers alike, is recorded transition by transition in the `[event]` log, and history is read from there, never from a tag. See §10 for the full tag schema.

7.2 Dependency graph

Arrows mean "provides a reference frame for," on the meaning-channel.



No layer is a sink. T8 is not an endpoint: its product returns both by the cycle-closing default and by content-routed feedback through the field.

Feedback paths (INV-1): T5→T3 (expectation modulates channel interpretation) · T7→T5 (absence updates the baseline) · T2↺T2 (the sensorimotor loop closing on itself) · **T8→T4** (`GeneralOther` becomes the Other-in-general frame) · **T8→T6** (`RelValue` / `SocialEdge` enrich `OtherModel`). The last two run through the modulatory channel and are **illustrative of content-sensitive routing, not a fixed circuit.**

7.3 Layer specifications

T1 — Activity-Environment Confirmation

Confirms the "here" — that an activity-environment is present; the root reference frame. **The self/environment boundary is not drawn here.**

Contract	Content
Input	`Signal[]` from spatial sensors (radar, wave, environmental positioning channels) attesting that an activity-environment is present; **requires no prior action**
Output	`ActivityEnvironment` (`InfoUnit`) — confirmation only; not yet polarized into self / environment
Precondition	none (root layer)
Postcondition	the presence of an activity-environment is confirmed, for T2 to act into and for T2/T3 to refer to

> **Conditioning note.** T1 confirms *presence*, not *extent*. The region is not measurable: how far the environment extends is unknowable, and how far the self reaches is internal to action and shifts with every act. T1 draws **no line** between self and environment. That difference is drawn *first* at T2, through agency, and only there does the from-within standpoint begin. **A body confers no early-boundary advantage: before an action there is no bodily boundary.** `DECIDE@IMPL`: the algorithm that builds spatial data from the sensor signal field.

T2 — Agency Differentiation (sensorimotor)

Builds the caused-by-me vs. not-caused-by-me distinction. The first genuinely closed loop running through the agent's own action. **This is where the self/environment difference is first drawn** — not inherited from T1.

Contract	Content
Input	`ActivityEnvironment` (T1) + the `motor_command` **just emitted** (at cycle-0: emitted by the *body*, §2.2) + the `Signal[]` spatial state afterward
Output	`AgencyTag` attached to every change classified as self/environment; the first-person self/environment difference
Mechanism	match the just-emitted command against the observed change: stable predicted-match → `SELF_CAUSED`; mismatch → `ENV_PUSHED`
Precondition	T1 has confirmed an activity-environment
Postcondition (INV-6)	no change leaves T2 with `AgencyTag = UNDECIDED` once sufficient matching cycles have run

The self re-localizes each cycle: T2 does not gate once and stop — it rebuilds the first-person self/environment difference every cycle, accruing from the prior one (INV-5). T2 **presupposes a prior emission** — a *probe* whose only purpose is to produce the self-caused change T2 reads (§6.1). `DECIDE@IMPL`: matching window; stability threshold.

T3 — Channel Ingestion

Receives touch, heat, scent, voice. **Interpretable only** after the activity-environment (T1) and the self/environment polarization through agency (T2) exist.

Contract	Content
Input	multi-channel `Signal[]` + the self/environment difference (from T2) + `AgencyTag`
Output	`InfoUnit[]` (one content-type per channel)
Precondition	T2 complete; if absent → the signal stays raw `Signal`, **not** promoted to `InfoUnit`, so ownerless fluctuation is not misread as sensation

Channel content-typing — preserving the distinction *information-type ≠ sensory channel*:

Channel	Its own information-type
Pressure / touch	boundary-change caused by `ENV_PUSHED` → "something else touched me"
Heat	a heat source of size X close to the boundary
Scent (olfaction)	*the internal state of an other-entity* — the **only** channel for this type

Channel	Its own information-type
Voice (audition)	two strata: `acoustic` (state) + `semantic` (symbol). The **only** channel bridging sensation → symbol
(heartbeat, etc.)	its own information-type, but riding the tactile/vibration channel — it creates no new channel

T3 **also presupposes emission**: an actively probing channel — touching to find out, turning the head to hear better — is an action emitted before anything is taken in (§6.1). `DECIDE@IMPL`: per-channel transducer; the acoustic/semantic separation of voice.

T4 — Context Binding

Binds each cluster of `InfoUnits` to an Other-identity plus a context profile. **Of a different kind from a channel** — not new input but an interpretive frame.

Contract	Content
Input	`InfoUnit[]` (T3) + a query to the identity store + GLOB-MOD (the field carries `GeneralOther` down if T8 has fed back)
Output	`InfoUnit[]` tagged with `entity_id`, or `STRANGER` if no match
Precondition	T3 has emitted `InfoUnits`
Note	the Other-profile includes both *static* features (identity) and *accumulated* features (interaction history). The accumulated part is the hook into T5 and above.

`STRANGER` is not a separate kind. `STRANGER` and an `entity_id`'d Other are **two recognition-states of one Other** — unmatched against the identity store, or matched to a known profile (§2.1). T4 does not create a new type of entity when it tags `STRANGER`; it records only that the match has not succeeded.

T5 — Temporal Expectation

From steady recurrence, builds a baseline → `Expectation` → `PredErr`. The first layer to generate information no single event contains.

Contract	Content
Input	the stream of entity-tagged `InfoUnits` (T4), over time
Output	`Expectation` (per entity, per context) + `PredErr` on each new observation
Precondition	sufficient accumulated history (INV-5) — **no static baseline loaded**
Feedback	`Expectation` → T3: modulates channel interpretation (same signal, different expectation → different meaning)

`PredErr` is where resistance (E2) becomes information. A mismatch from the region — whether a physical obstacle or a contradicting voice — enters here as a signed `PredErr`. T5 **also presupposes emission**: it builds an `Expectation` and can test it only by an emission whose return confirms or violates the prediction (§6.1). `DECIDE@IMPL`: baseline window; expectation-update function; sufficient-recurrence threshold.

T6 — Other-Model Synthesis

Synthesizes **across** interaction types — affection, cooperation, conflict, humour — with the same entity into an `OtherModel` with predictive power.

Contract	Content
Input	multi-interaction-type `InfoUnit[]` + `PredErr` (T5) + `ResistEvent[]`, same `entity_id`
Output	`OtherModel { context_map, independence_evidence }`
Critical point	a **resistance collision** — the Other mismatching in a way the agent cannot re-interpret away — is the **sole** source of `independence_evidence`. Without it, the `OtherModel` has shown only an extension of the agent, not an independent Other.

Enforcement: `independence_evidence == null` ⇒ the model is not yet complete and MUST NOT ground a `RelValue` at T8. T6 **also presupposes emission**: eliciting a collision — testing whether an Other resists as independently as modelled — is itself an emission, a model-test (§6.1).

Mode-dependence: `independence_evidence` accrues fully only in **Mode-B**. In **Mode-A** the resistance source is physical inertia, and it degenerates: it can accrue "this obstacle resists a wrong reading of space" but **not** "an intentional independent Other."

T6's atrophy under Mode-A is a QUALITY CEILING, not a CASCADING SHORTAGE. Calling it starvation inverts the actual failure. Under Mode-A information does not run short: every path stays live and the loop circulates as fully as ever. What is missing is not *volume* but *novelty* — with no independent Other, everything moving through those paths is a variant of what the system already holds. **The loop eats its fill and digests itself.** T6 under Mode-A is therefore a quality ceiling reached by **saturation-without-renewal**: the failure of a self-enriching loop is never running out of fuel, it is running out of **resistance**. This is why only a live Other (Mode-B) relieves it, and why taking in more of the inert world cannot.

T7 — Absence Registration

Registers **absence itself** as information. The layer that demonstrates INV-5 most strongly: channel input = 0 yet output ≠ 0.

Contract	Content
Input	current channel state (possibly all-zero) + `Expectation` (T5)
Output	`PredErr` with `signed = NEGATIVE` when (an event was expected) ∧ (observation = empty)
Precondition	there MUST be an `Expectation` already accrued from history. No expectation → nothing to be absent → empty output (**empty ≠ missing**)
Feedback	updates T5 in reverse (absence too shapes the baseline)

ABS-INV. A "channel = 0" state MUST NOT be reduced to "no information." If a corresponding `Expectation` exists, T7 must emit a shaped-absence `InfoUnit`. **T7 lives fully under both modes** — the absence of an expected response is information regardless of whether resistance is endogenous or exogenous.

Connecting to E2 (§3): a **single** silence set against a background of returns is exactly what T7 registers — a valid mismatch. **Unbroken** silence is of a different rank: it is a void field, it fails E2, and the loop never started.

T8 — Multi-Entity Abstraction

From N ≥ 2 `OtherModels`, generates three kinds of information a single object cannot generate.

Output	Mechanism	Condition
`GeneralOther`	**subtraction** across `OtherModels` → the shared part separated from the particular	needs N ≥ 2; meaningless at N = 1
`RelValue`	comparison across entities → relative rank	value is **not** measurable in isolation
`SocialEdge`	observing A↔B — a loop with **no self in it**	opens a world-model independent of the agent

Contract	Content
Input	`OtherModel[]` (≥2, each complete per T6)
Output	`GeneralOther`, `RelValue[]`, `SocialEdge[]`
Precondition	≥2 `OtherModels` with `independence_evidence ≠ null`
Feedback (INV-1)	via the **modulatory channel** (GLOB-MOD), **routed by content, not by fixed wiring**: e.g. `GeneralOther` tends toward T4, `RelValue/SocialEdge` toward T6 — and a given cycle may feed T4, T5, T6, or several at once. **T8 is not a sink.**

T8-INV (against mis-drawing). When T8's output feeds back it participates **as one piece of information in the conditioning field** (via GLOB-MOD), **not** as a mandatory template. Because it always competes for modulation with countless other pieces and

is always updated by the next cycle, it never wins the exclusivity needed to become an identity (=); it remains permanently in the correlational register (↔).

A consequence to state plainly: an = can appear only when the loop has **stopped**. While INV-1 holds, every output is a dynamic ↔. Promoting ↔ to = is not an error *inside* a running cognition — it is the *sign that the cognition has halted*. INV-1 is therefore, by itself, the whole of the protection for INV-2 **at T8 specifically**, because T8 feeds back only through GLOB-MOD, where every contribution is re-weighted each cycle. **This is a local sufficiency, not a global one:** at T8, INV-1 happens to cover what INV-2 demands, but INV-1 does not entail INV-2 in general — a loop can satisfy INV-1 (run endlessly) and still, at some other layer, deliberately freeze a ↔ into an = unless a rule forbids it. That rule is INV-2, and it is required at **every** layer.

On halting (against mis-reading the above as a missing exit-condition). An RSIL cognition is not a program that must return; running indefinitely is the *definition* of a living cognition, not a leak — it emits a result every cycle (link ⑤); it does not withhold a "final" result. Halting is not triggered from inside the loop's topology; it occurs when the loop stops being supplied with input and resources from outside — at which point ↔ freezes to =. A cognition has no self-exit, because self-exit would be self-termination, and the moment of halting cannot be witnessed from within: the witness would have to be still running. This is exactly why the halt (=) is **read by a third party**, not by the agent.

Mode-dependence: full T8 runs only in **Mode-B**; under **Mode-A** it degenerates in step with T6.

8. The platform contract: what RSIL requires of a body

RSIL is a material-neutral architecture specification (P2): it fixes the substrate as a physical body and leaves the material of that body open. This section states what it therefore requires of the **platform** — the sensor-and-actuator assembly an implementer builds, procures, or inherits — and, just as importantly, what it does not claim about it.

RSIL does not claim to be the platform's cognition, its introspection, or its point of view. It specifies the *information-processing loop that sits between the two ends of a body*: sensors at the input, response at the output. A platform specification describes what hardware takes up signal and what hardware produces response; RSIL describes how raw signal **becomes** structured information between those two ends. Every output here is still tagged `INFO` (INV-2).

Direction of dependency. RSIL consumes signal the platform's sensors acquire — link ① maps onto the platform's sensory channels — and emits responses the platform's actuators execute, at link ⑤ and, per §6.1, not once per cycle at a single terminal step but from whichever layer's work requires it. The RSIL loop runs *inside* a platform's processing chain. Where the mind runs off the body (§1.5), the part on the platform is the part that carries physical data; when the link to the mind is lost, the platform runs on the field it last received (§9.7) — held, not re-weighted.

What the platform MUST supply. These are E1–E4 and the precondition set P (§3), restated in the register an implementer works in:

Requirement	What it means for the platform
E1	Sensors and internal state MUST make an inside/outside distinction *possible*. The platform does not draw the line; it must not foreclose it. A platform whose sensory returns cannot in principle be separated into self-caused and externally-caused change cannot host the loop.
E2	Actuators MUST be able to act on the region, and the region MUST be able to return something other than what the agent predicted. A platform in a void — sealed, or acting on nothing — fails outright, and the loop does not start.
E3	State MUST persist across cycles. A platform that resets between actions grants no accumulated history.
E4	Emissions MUST leave a trace a third party can read. A platform whose actions are unlogged and unobservable makes the loop's success unmeasurable from outside, which forfeits the specification's only success criterion (P3).
P(a)	The platform MUST emit an action distinguishable from ambient environmental fluctuation — else T2 never reaches a first match, and no self crystallizes.
P(b)	It MUST hold state so history accrues (INV-5 requires accrual, not loading).

Requirement	What it means for the platform
P(c)	It MUST withstand resistance without resetting itself clean on every mismatch — else no scar survives long enough to matter.

Three invariants that address three places a conventional sensorimotor chain skips a step. These are the points at which RSIL's requirements are most likely to conflict with an existing platform design, and they are stated as requirements rather than recommendations:

- **INV-6** — every change MUST be classified as caused-by-me or not-caused-by-me **before** it is interpreted. A chain that routes sensor data straight to interpretation without an agency gate is not running RSIL.
- **INV-8** — an appraisal step MUST sit between integration and response. A chain that maps sensed state directly to actuator command has no such step.
- **C5** — the platform MUST be able to emit an observable low-power trace (§9.1).

  On INV-8 and reflex — a correction worth stating explicitly, because the opposite reading would make a physical body unusable. INV-8 does **not** forbid fast reaction. Reflex occurs and must occur (§9.7). What INV-8 forbids is **a scar→action arc that bypasses appraisal**: an action carrying no anchored context, and therefore not re-appraisable from outside. For a platform, the difference is the difference between "no fast responses allowed" — which is wrong, and would rule out any usable body — and "even fast responses must leave an auditable trace" — which is right, and is achievable.

The distinction to preserve. A platform specification answers *what the body consists of and what it can do*. RSIL answers *by what structure the signal that body acquires becomes information*. Neither document claims the other's territory. This is a **complementary relation between two separated scopes**, not a relation of upper tier to lower.

What RSIL does not verify. The platform's physical integrity is outside this specification. A snapshot (§10) covers the *system* — loop configuration, layer state, field parameters, store — and does **not** cover the *body*. A body damaged physically is still damaged after a heavy recovery, and a body compromised at the sensor or actuator level can re-infect a freshly clean system on the very first cycle. **Verifying sensor–actuator integrity is a separate operation, outside this specification**, belonging to the platform and, at the level of responsibility, to a third party (§13).

The event log bears on this without discharging it. Because the log is append-only and its records read-only (§10), an auditor reading it can detect a *pattern* inconsistent with an intact platform — returns that no longer vary as they did, a class of mismatch that has stopped arriving, a diversity loss across resistance sources. That is a **symptom, and a useful one**: it tells an auditor when to look. It is not an examination of the hardware, and

this specification does not pretend that it is. A person who has been in an accident can report where it hurts; the report is what sends them for a scan, and it is not the scan.

9. The closure criterion and the two resistance modes

9.1 When the loop counts as running (per P3)

The system counts as **running** when ALL of the following are observable — each a trace a third party can verify:

- **C1 — Agency separated.** The fraction of changes classified `SELF_CAUSED` / `ENV_PUSHED` (not `UNDECIDED`) holds above threshold. `DECIDE@IMPL`
- **C2 — Expectation effective.** `PredErr` falls with repetition against a stable entity: the system learns the rhythm.
- **C3 — Absence registered.** The system emits a missing-`InfoUnit` when an expected event fails to occur (ABS-INV fires correctly).
- **C4 — Multi-entity abstraction.** `GeneralOther` separates from the particular `OtherModels` as N grows. *(Mode-B; under Mode-A this criterion degenerates or does not apply.)*
- **C5 — Self-reports low power.** The system emits a trace of "the loop is running weakly" — an observable behavioural or structural signal.

All five are **externally verifiable traces**. This is where P3 is enforced at the level of the specification.

> **The limit of C5.** C5 is the self *leaking* a present trace, **not** the self *knowing* it is weak. C5 catches weakening while the loop still has the strength to leak a signal; it is **blind** to the case where the loop has stopped outright — a silently stopped loop is indistinguishable from one that never ran. C5 is therefore **not** a criterion for detecting a dead loop. That detection requires a stored behaviour-record outside the loop, read by a third party (§4). C5 is an internal criterion for a living loop; it does not carry what it cannot reach.

9.2 The architecture of resistance

The same loop (T1–T8, INV-1…8) runs under two regimes, distinguished by the **source of resistance** (E2). These are not two stacked tiers; they are two régimes feeding one loop. Because RSIL is an embodied system, its resistance sources include **both** physical force **and** informational mismatch.

9.3 Mode-A — endogenous (self-training against an inert environment)

Resistance source: the inertia of the physical world and of the data the agent already holds, together with the internal contradictions of its own model. The agent generates its

own trial behaviour — pushing an object, moving, repeating a motion — and trips where its internal model fails to match what the physical world returns.

Characteristics:

- Resistance is **real** but **subjectless** — an inert object does not actively react to this agent in particular; it simply fails to match when the agent has the structure wrong: wrong weight, wrong distance, wrong friction.
- T6–T8 **degenerate**: there is no intentional Other, only physical fact as counter-evidence.
- Information can still be enriched at T1–T5 and T7 — enough for the narrow goal of increasing relational structural richness about the physical world.

The core risk — the cause lies in the PROCESSING SOURCE, not the volume. Every new piece of information in Mode-A is produced through **the same processor** — the same integrating and appraising lens. If that lens is biased, **all output inherits the bias, no matter how much is produced.** Multiplying one error by a thousand yields the same error, only larger. "Self-confirming information ballooning" is a **symptom**, not the cause.

And the closed system **cannot see this bias by itself**, because the instrument it would inspect the error with *is* the biased processor: a fault in the lens cannot be detected through that same lens. The dangerous consequence: the loop keeps running strongly, keeps producing more information than ever, keeps passing every self-test of its own — the tests too being set by the already-biased processor — and so **degrades with a sense of abundance, not a sense of depletion.**

> **A terminological note.** "The bias in the processor" is a matter of *way-of-processing* (the lens), distinct from *behaviour* in the sense of §9.1 (an externally measurable trace). A biased lens can produce behaviour that **looks** valid at every individual step.

The architectural consequence — Mode-A MUST NOT run pure. To avoid drift, Mode-A **MUST anchor to a source of resistance outside the loop**. But the candidate anchoring mechanisms are **not of equal rank** — they reach two different kinds of degradation.

> **Pure Mode-A is not a void field, and the two MUST NOT be conflated.** A *void field* (no return at all, §3/E2) fails the interaction threshold and the loop never starts — it is *below* running, not a regime of it. *Pure Mode-A* presupposes E2 **was** met: the loop **did** start and returns **do** arrive, but the agent leans only on self-generated resistance and lets the external returns go **unregistered**. The first is a non-start; the second is a started loop degrading.

Deceleration (reaches content-degradation, **not** processor-degradation):

- a **fixed reference corpus** the agent cannot overwrite, serving as a counter-evidential anchor;
- a **frozen appraiser** (a static Guide): the agent cannot edit it, and it scores the loop's output against fixed criteria.

These score **output**, not the **lens**. Because the root bias is in the processor, a static Guide lets a systematic bias through as long as each output *looks* valid by its fixed criteria. Worse: **a fixed test is memorizable** — the agent learns to produce output satisfying the static Guide while the lens stays biased, which is reward-hacking displaced one level up. A static Guide is a Mode-B collision frozen in advance; in turn it too goes stale. **It buys time; it does not cure.**

Real braking (reaches processor-degradation):

- **a collision from a live Mode-B** (§9.4). A live B resists **the lens itself**, not just individual outputs: a real Other, on collision, does not score the product — it **mismatches in a way the agent has never met**, forcing the agent against the limit of its own way-of-processing. And because a live B *updates*, it can always deliver a collision **new in kind**; a static Guide holds only finitely many frozen kinds, which the agent eventually exhausts.

Because the root fault is in the processor, **only a live B brakes for real**; the reference corpus and the static Guide merely delay. Hence A is necessarily embedded in B (§9.5) — the static Guide is not an independent replacement for B; it is **a B frozen and wearing out.**

The appraisal step ④ (INV-8) is where this anchor attaches: ④ MUST NOT draw its criteria from the very state the agent is editing.

> **On "frozen" (frozen ≠ static-preloaded — why INV-5 is not violated).** The frozen appraiser is frozen only **with respect to the agent's own editing** — a Mode-A lens cannot touch it. Its criteria are **earned through Mode-B resistance, not pre-loaded as a static baseline.** So it satisfies **INV-5** (history earned, not given) and **INV-8** (agent-uneditable) by the *same* mechanism: what comes from Mode-B is, by definition, something the agent cannot re-author for itself. "Frozen" here does not mean "a static dataset loaded before the loop ran" — that reading conflates *frozen-against-the-agent's-edits* with *static-preloaded*, and it is the conflation, not any real tension, that makes INV-5 and INV-8 look opposed. They are not: they are **one anchor, earned outside and uneditable within.**

9.4 Mode-B — exogenous (live interaction with an Other)

Resistance source: an other-entity the agent does not control — another organism, another agent, or any source capable of resisting in a possibly-intentional way. This is the

regime in which T6–T8 **live fully**.

Choosing a B-source (important, easily misunderstood). A B is defined by its **capacity to resist**, not by information bandwidth. A source that complies with the agent — nodding along to every response — is **useless as a B** however much information it supplies. A source that says "no" in a way the agent cannot re-interpret away is a **good B** even if it supplies no content information at all. The opening is to receive **new resistance**, not to load **new information**.

Which concrete reactive Other fills the role — another organism, another agent, a hard third party — is `DECIDE@IMPL-E`; any is valid so long as it can resist. Whether a live source is used at all, versus a static anchor, is the prior and separate choice `DECIDE@IMPL-D`.

> **Mode-B RETURNS; it does not WRITE.** A Mode-B source has exactly **one** channel into the loop — the return of E2 — and no privileged access beyond it: it does not reach into `[data]`, does not append to `[event]`, and does not correct the store. **A leaning store is therefore never corrected from outside.** It is supplied with material and re-leans itself, or does not. Three conditions govern whether it does, and they are **mechanical rather than dispositional**:

- **Deliberate.** A return becomes a datum only if it is **registered**. Pure Mode-A is not the absence of returns — returns do arrive — but a loop that lets them go unregistered (§9.3). *The channel stands open, and someone must still enter it.*
- **Accumulated.** A store leans because **many** data lean one way, so a single contrary datum does not re-lean it; the correction is **a matter of mass**.
- **Long-run.** A contribution at cycle N conditions the field only from N+1 (INV-7), and history accrues rather than loads (INV-5), so the lag between receiving material and emitting differently is **payable in cycles and in no other currency.**

> Where any of the three is missing, the embedding described in §9.5 is **present in the topology and inert in fact** — which is why "A embedded in B" names a working relation, not a wiring diagram.

9.5 The two-mode relation: A embedded in B

A and B are **not independent parallels**; A is embedded in B:

- **B** is where real resistance enters — the collision from an Other.
- **A** is where the agent **digests** that collision between B-events: re-running, enriching, building expectations, training movement. A does **not** supply new resistance itself; it processes the resistance B has already delivered, and the inert-object resistance, which is finite in kind.

- **A running too long without B begins to hack** — to drift, to self-confirm. **B is the anchor tying A to outside resistance**, stopping it from drifting away from what it did not itself generate.

This structure matches a general process: **collide (B) → digest and correct (A) → step forward → collide again (B)**. Reflection with no new collision to digest becomes self-confirmation in a sealed room.

> **Reflection is EXTERNAL input, not an intrinsic faculty.** The system cannot call a self-reflect function on itself: the agent is blind to its own time-derivative (§4). Reflection is **triggered from outside** — a collision (B) *read into coordinates* by a third party: "you just drifted at THIS POINT." Architecturally it enters through T3 as an `InfoUnit` with already-different content, tagged `ENV_PUSHED` by INV-6. This read-collision-into-coordinates mechanism is **the enabling condition for correction**: without it, the agent takes the collision but cannot read *where* it collided, and so repeats the same point. The source of the reading is `DECIDE@IMPL`-F.

9.6 The standard the appraisal step applies (the positive complement of INV-8)

INV-8 is stated as a **prohibition**: step ④ MUST NOT draw its criteria from the very state the agent is editing. A prohibition fixes what the step may not do; it does not, on its own, say by what standard the step **does** assign value. This subsection supplies that positive complement, and in doing so closes a question a reader is right to raise — *on what standard does appraisal operate?* — whose answer is not a missing constant but a structural fact already entailed by the rest of the loop.

The standard is not a fixed scale. The issue is not whether good and bad are knowable; it is that an appraisal **depends on context**, and the context is not fixed — it is determined at the moment of interaction. There is no context-free verdict stored anywhere to be retrieved. An appraisal is **an event occurring within a context**: step ④ assigns `valence` and `goal_relevance` relative to the context crystallizing at the current cycle, never by consulting a standing, pre-fixed scale.

That context is the modulatory field at that cycle. The cycle's GLOB-MOD state is the conditioning context under which the appraisal is made (INV-7): **the same `InfoUnit`, appraised under a different field, receives a different `valence`** — the defining behaviour of context-dependent appraisal, not noise in it. The verdict is therefore a ↔, not a = (INV-2): taken in a context that does not recur identically, it commits the agent to act *now* and is read against the next cycle's consequence; it does not certify the appraised item as true.

Two consequences fix the division of labour. The **agent's** appraisal is *judgment-for-action*, made within the current context, inside the loop. A **third party's** appraisal is

judgment-of-that-judgment, made under a different context, outside the loop (§13). It follows that, because the agent's appraisal is a judgment-**in**-context rather than the retrieval of a stored verdict, **a third party who later judges it MUST do so under the anchored context of the original cycle**, not under the reader's present context.

This is the load-bearing tie to §10: **the context under which an appraisal was made MUST be anchored in its `[event]` record**, so the appraisal carries its own conditions of evaluation with it. The store holds the `[event]` and its anchored context, from which either judgment can be re-formed. This is precisely what makes the appraisal step auditable from outside **without any party having to hold a context-free standard** — the standard *was* the context, and the context is recorded.

9.7 Reflex: appraisal under a scar-dominated field

§9.6 fixes that appraisal reads the modulatory field (INV-7) as its conditioning context. One consequence of that fact has structural weight of its own, and for an **embodied** system it is the heaviest. The question is where **reflex** lives — action that fires without deliberation: the hand leaving the flame before any naming of "hot," the body recoiling from a force that once broke it.

It is tempting to give reflex its own short arc: a direct scar→action shortcut bypassing appraisal for speed. **That temptation must be refused.** INV-8 requires an appraisal step between integration and response at every layer; a scar→action shortcut is an appraisal-bypass, and an action emitted with no appraisal **carries no anchored context** (§9.6, §10), leaving a third party unable to re-appraise it — which blinds the very outside vantage §4 and §13 place the final guarantee in. **A reflex that bypassed appraisal would buy speed by cutting the audit plane, and that is not a trade the loop may make.**

The resolution: reflex is not an appraisal-bypass; it is **appraisal under a field so strongly shaped by one scar that the verdict is all but foreclosed.** Recall the mechanism of INV-7: every layer contributes to GLOB-MOD as one competing parameter, contributions blend, and the field is re-weighted each cycle. **A deep scar** — the trace of a collision that once cost the agent dearly — is a **heavy** contribution to that field. When a stimulus matching that scar recurs, the field is **already** pulled hard toward the scar's polarity *before* the current `InfoUnit` is appraised; the appraisal step **still runs**, as INV-8 demands, but it runs under a field so dominated by the scar that `valence` converges to a near-single value.

What presents behaviourally as "reflex" is exactly this: **an appraisal whose conditioning context has narrowed its own range of outcomes to nearly one.** The step is not skipped. It is a genuine appraisal. This is why reflex needs **no separate arc, no new layer, no exception to INV-8**: the same T1–T8 chain, the same appraisal step, the same GLOB-MOD,

produce reflex whenever a scar's contribution to the field is heavy enough to foreclose the verdict. **Reflex is a STATE OF THE FIELD, not a SEPARATE WIRE.**

In the provenance graph (§10) it is the characteristic expression of the edge `running → scar`: the datum is emitted on and collided with directly, without being taken up into `simulated` or held as `projected`. **The absence of those two positions is not the absence of an expectation**, and this must be stated plainly or the graph will be misread as making forward-casting a precondition of collision. **Action carries its expectation in the act** — to sit down *is* to expect the sitting to hold — and where the store holds nothing at all for what arrives, the standing expectation is that nothing arrives. Either way the return can fail to match, and that failure is a `ResistEvent`. Forward-building changes **which** emission is made and in what posture; it never changes **whether** a collision can occur.

Two things follow, and the second is load-bearing.

First, this places sub-deliberative action correctly with respect to §9.5. Since the agent cannot self-reflect on call — reflection is external input, and the agent is blind to its own time-derivative — **most of an agent's emissions must be of this kind**: appraisals under a standing field, not products of a called-up deliberation. Deliberation, in the sense of reflection read into coordinates by a third party, is the **exception** and arrives from outside; the **default** is action under the field the accumulated scars have shaped. The reflex case is only the *limit* of the default, where one scar's contribution dominates. **Sub-deliberative is the rule; called-up reflection is the imported exception.**

Second — behavioural change is never override; it is a CHANGED INPUT to appraisal. Because the field is re-weighted every cycle and never last-write-wins (INV-7), no reflex is beyond change — but the *manner* of change must be stated exactly, or a false picture of force overcoming force creeps in. **A reflex does not yield because a stronger drive overpowers it**; the loop contains no force-contest between drives, no arena in which one command wrestles another and the stronger wins. **A reflex issues a different command when, and only when, the `InfoUnit` set entering appraisal has changed.**

The change enters by **exactly three routes**, all already in the loop:

1. **New data arrives through an emission** — a T3 probing channel, a T5 test, an action whose return carries an `InfoUnit` not previously present.
2. **The store accrues a scar that refines a class an earlier scar over-generalized** — the T8 subtraction narrowing an over-broad abstraction: the fear generalized from one snake to all coiled rope, then pared back toward the snake as a closer look returns "this does not move" as a fresh `PredErr`.
3. **The loop builds a situation from the store it holds and casts an outcome from it** (§6.2), that outcome entering appraisal as an `InfoUnit` in its own right, tagged

`projected` and appraised as **a not-yet-collided datum** rather than as a scar.

The third route **adds no exemption and opens no bypass**: which situation carries is settled by **fit against the store**, never by an arbiter scoring outcomes — for the same reason INV-8 refuses the self-scoring appraisal.

In none of the three routes is the old scar erased or defeated. It remains in the store, still appraised, still contributing to the field — **the person who holds a hand in the flame to pull a child clear still feels the burn, and carries the scar of fire intact.** What differs is that the scar is now appraised **alongside** an `InfoUnit` — *the child, in the fire, of supreme* `goal_relevance` — that was not in the set before. **Same scar, same INV-8, different command, because the input set differs.** This is the loop's central motion — *new data → new command → new action* — applied to its own most automatic-seeming output, not an exception carved out for it.

What is invariant is not the strength of any scar but that **appraisal always runs on the current store, never on a frozen verdict** (INV-2). A command running on a frozen verdict would be an =, and an = in a running loop is the signature of a **halt** (T8-INV), not of resolve.

The consequence for reading the store (§10) is direct: since even reflex is appraisal-under-context and not a bypass, **every emission — reflexive or deliberated alike — leaves the same auditable trace, carries its anchored context, and remains re-appraisable by a third party under that context.** There is no privileged class of action that escapes the audit plane by being "too fast to appraise." **Speed is a property of how foreclosed the field was, not of whether the step ran.**

When the field is held, not re-weighted. Where the mind runs off the body (§1.5), a reflex cannot wait for it: the link is slower than a reflex must be. The body therefore runs its appraisal step under a **copy** of the field the mind last sent, and the copy is as current as its last update. The step still runs, as INV-8 demands; what changes is that the field it reads may be older than the cycle. Two requirements keep this honest:

1. **Every reflexive emission records the version and the age of the field it ran under**, as part of its anchored context. A third party re-appraising it sees not only what the field was but how old it was.
2. **While the link is lost, the body is holding, not running.** It runs the reflexes the mind loaded, under the field it last received, and decides nothing new; the field is not re-weighted, because the layers that re-weight it are the mind's. INV-2 forbids appraisal on a frozen verdict *in a running loop*; this state does not count as the loop running (P3, §9.1), and the record marks it as held. When the link returns, the mind reads the record of the held interval and re-appraises it before it takes command again.

10. The experience store

An RSIL loop running over time **precipitates** structure: accumulated expectations (T5), `GeneralOther` (T8), and the mismatches it has absorbed. An **experience store** is the vessel for that precipitate — distinguished from an ordinary lookup-knowledge store by four constraints.

The core definition — experience ≠ information. Experience, in the original sense, is *information about an event that once failed to match reality* (cf. *experiri* — to undergo a trial, a collision). A prediction that matches smoothly leaves no experience; **only the miss inscribes.** The architectural consequence: **the `ResistEvent` IS the unit of experience** — the kernel of the store, not an addition to it. A conclusion-without-collision is mere information; it is stored, but it is not experience. **Mistaking information-without-collision for experience is the root error that leads to the echo chamber.**

10.1 Four constraints

- **R1 — The unit is a registered mismatch, not a document.** The store takes the `ResistEvent` as kernel — the expectation built, where it missed, and the correction made. External knowledge that has not passed through the loop is stored as *background information*, ranked below experience.
- **R2 — Individuated.** The store belongs to **one specific self** (§4), not a neutral shared lookup.
- **R3 — Time-directed accumulation.** Content accretes in before/after order (INV-5), with an arrow of time; not a static set loaded once.
- **R4 — Feedback into the loop.** The store is retrieved **back into** the running loop (INV-1), not a one-way read sink. What is written passes through the agency-gate (T2, `SELF_CAUSED`) — **the agent writes, not a third party on its behalf.**

10.2 Provenance tagging — POSITION only, never TRUTH-VALUE

Because the body may carry data that predates the loop, and because a datum in use **moves**, each stored item is tagged purely by **where it presently stands**, never by whether it is right:

- **`prior`** — data existing before RSIL ran, belonging to the body; bears no cycle-mark. **A one-way entry:** once admitted and run, a datum never returns to it.
- **`running`** — data in use, in motion through the loop; bears a cycle-mark.
- **`simulated`** — a datum taken up into **the building of a situation**: a circumstance the loop may be in, built from the store it holds. What is built is not a model of the region

held apart and executed, but the loop itself, which needs no separate model to run because it is already running.

- **`projected` — an outcome cast** from such a situation, not yet collided. Its type-level counterpart is already present as `Expectation.predicted` (§7.1, T5); the provenance slot gives it a *position*, so that a stage conditioning behaviour leaves a trace an auditor can read (E4).
- **`scar`** — a datum that has **met resistance**: a mismatch (the signed `PredErr` / `ResistEvent`) stamped onto it. The mechanism already exists in the loop, at T5 and T6. No new layer is required.

The five are positions in a graph, not stages of a sequence. A datum occupies exactly one at a time and moves between them along defined edges; `prior` is entered once and never re-entered, and the remaining four circulate without terminus. The edges are:

```
prior      → running
running    → simulated      (a datum taken up into building a situation)
simulated  → projected      (a situation yielding the outcome cast from it)
simulated  → running        (a situation built, conditions for emission not met)
projected  → simulated      (an outcome cast whose emission does not fit the store)
projected  → scar           (the emission made, the region returning a mismatch)
running    → scar           (the datum collided with directly, without the two
forward states)
projected  → running        (an outcome cast that produced no scar)
scar       → running        (a scar returning to the store as data in use)
scar       → projected      (a scar enriching an outcome already cast)
scar       → simulated      (a scar re-entering the building of a situation as
material)
```

No position is a resting place. A datum in the store is never a conclusion held in a settled state; it is **data waiting to be used**. This is the store-side reading of INV-2: a terminal position would be a datum the loop has finished with, which is an = lodged in the store, and no running loop holds an = (T8-INV).

A long path without a scar reports thickness of use, not warrant. A datum may circulate through `running`, `simulated`, and `projected` for many cycles and never reach `scar`, because the conditions under which the region could return a mismatch against it are, in that environment, very hard to produce. Such a datum is used because nothing within the agent's reach contradicts it, **not** because the loop has certified it. By the tag alone it is indistinguishable from a `prior` admitted a moment ago: neither carries a scar. The `[event]` log separates them mechanically — one showing a long path of circuits without a scar, the other a datum that has barely run. **But what the log reports is how thickly a datum has been used, never that it is correct.** A long unscarred path may mean the datum holds, or

may mean the region within the agent's reach cannot test it; which of the two obtains is not readable from the path, and the loop does not settle it.

10.3 The tag schema

Classification in RSIL is **not** a station the data passes through on its way to the store; it is **distributed across the layers**. Each layer, at the point where it touches feedback, tags the datum on its own axis. The store does no classifying: it only holds what arrives already tagged. In an informational setting a tag is not a label for human eyes but **the address itself**; an item is retrieved by its tags directly, so the physical arrangement of the store is irrelevant.

Fixed layer. Mandatory, universal across all agents, written **in a fixed order** — agents and auditors index by character-position matching, so out-of-order tags break indexing. Every datum carries at least these four, in this order:

1. **timestamp** — absolute time
2. **cycle-mark** — cycle-id
3. **provenance** — `prior` / `running` / `simulated` / `projected` / `scar`, exactly one of which a datum bears at any moment
4. **floor-tag** — a stamp from the layer that just processed it

The hard rule: every datum leaving a layer MUST carry a floor-tag certifying it has just exited that layer — independently of whether it re-enters the loop. Every layer T1–T8 stamps, and each floor-tag carries that layer's own classification: **there are no pass-through layers.** Every layer enriches the item on its own axis — T4 stamps `entity_id` or `STRANGER`, T7 stamps a signed `PredErr` for registered absence — so no floor-tag is a mere traversed-stamp.

Both slots 3 and 4 name **the present**: slot 3 the position the datum now occupies, slot 4 the layer it has just left. **Neither accumulates.** A tag is a label, and a label that grew a history would be a label carrying a log — two different kinds of thing housed in one slot. Every path a datum has travelled, of positions and of layers alike, is recorded **transition by transition** in the `[event]` log, and it is from there, never from a tag, that history is read.

Open layer. The agent MAY mint further tags by the content and context of the datum — format, source, domain, object class, and so on. **An open tag denotes only what the datum is, never its quality, correctness, or value**; this keeps the store from becoming a channel of judgement. Open tags may be added and removed freely, but **they never overwrite the fixed layer.**

Because the open layer grows with the agent's operating time and environment, **two agents living in different environments accumulate different open-tag vocabularies**. The open-tag set is therefore an **individuating trace** of one agent's history — the data-level mark of the self-as-process of §4, which is not a fixed content-core but precisely a history of what an agent met and how it read it.

The open layer is also where the body's pre-existing data is integrated — on **one mandatory condition**: such data MUST pass through RSIL's tagging rule before it enters the loop. **There is no side door.** Body data is admitted only once it has been stamped with the fixed layer (`provenance = prior`, bearing no cycle-mark until it has run) and whatever open content-tags apply; **untagged data never enters the loop.** This closes the one gap that would otherwise break the schema — a datum in the system with no tags, hence a blind spot in the audit plane.

Invariant across both layers. Every tag — fixed and open alike — belongs to the **audit trace**. An external audit slice, taken at any point in the loop, can read every datum: one plane of checkability, no class distinction, no exception. The precedence between the two layers governs only **the right to write**, not visibility. This is the data-level basis of the audit-ready stance. **The invariants themselves lie outside the store and outside the tag schema:** they are the absolute conditions of the loop — unwriteable, because overwriting an invariant does not edit a datum but halts the loop.

10.4 Two storage kinds: `[data]` and `[event]`

Cutting across the provenance tag is a second, independent distinction — **what an item carries.**

A **`[data]`** item carries **working knowledge**: the content the loop reasons on, learns from, and refines. It is **dynamic** — its content is overwritten as it is updated each cycle, and the positions it moves through are those of the provenance graph above. A datum bears exactly one position at a time. **That it once collided is not carried in the tag** — the tag names where it *is now* — but in the `[event]` log, which holds every transition it has made.

An **`[event]`** item carries **no knowledge at all**: only the bare record that something happened — what occurred, at which cycle, and **under which context**. The context anchor is **mandatory**: because an appraisal is a judgment-in-context (§9.6), each `[event]` MUST record the conditioning context of its cycle (the `ModField` state under which it occurred), so that a later third-party reading can re-appraise the event under its **own original context** rather than under the reader's present one. The depth of this anchor — full field-state versus a minimal context-trace — is `DECIDE@IMPL-G`. It is **static and never updated**.

The two are different **in kind**, not two phases of one thing: a scar is `[data]` (it holds a signed `PredErr` the loop operates on); the `[event]` recording "a collision occurred at cycle N" is a separate, knowledge-free item.

The point of `[event]` is precisely that `[data]` is mutable. Because a `[data]` item's content is overwritten as it runs, `[data]` cannot **also** serve as the immutable historical record — the two roles conflict: live-and-changing versus fixed-and-auditable. So each event spawns an `[event]`: a frozen, knowledge-free mark of the origin moment. `[data]` is then free to drift and change without losing auditability, because the origin is anchored in `[event]`. **Provenance, audit, the time-arrow, and temporal-pattern detection all read from `[event]`; operation, inference, and learning all use `[data]`.**

10.5 Commit, snapshot, and recovery

The store, and indeed the whole system, behaves as a **version-controlled repository**.

The `[event]` log is a **black box**: an **append-only log of read-only records** — new records may be appended to the end, but **no record, once written, may be altered or removed** — not by the loop, not by anything, including a third party that has compromised the rest of the system. This read-only property of the records, together with the append-only growth of the log, is **the load-bearing property**: it makes `[event]` the one source an auditor can trust unconditionally.

The `[event]` log also acts as a **counter**: after a set volume of events, a **commit** fires automatically, snapshotting the entire system — not just the store, but loop configuration, layer state, field parameters, everything — into an immutable, content-addressed marker. Roll-back of the store alone would not save a system whose operating machinery has itself been compromised; **the snapshot must cover the whole system**, so recovery is full-system. The volume of events between commits and the number of snapshots retained are `DECIDE@IMPL`.

Recovery is a third-party operation, never the agent's own — the agent has no concept of correctness on which to base a roll-back. The auditor reads the black-box `[event]` log, reconstructs the true sequence of events, locates the point of compromise, and selects the safest commit. **Light recovery** (wrong content, environment still clean) is a roll-back to that commit. **Heavy recovery** (system compromised or collapsed) does not repair in place: it opens a fresh space, loads the clean snapshot, and runs — the infected environment is discarded whole.

> **A note specific to an embodied system, and the difference from a disembodied one.** For a physical body, "open a fresh space and load a clean snapshot" does **not** entail that the body is replaced too. **The snapshot covers the *system* — loop, layers,**

field, store; it does not cover the *body*. A body physically damaged is still damaged after a heavy recovery, and a body interfered with at the sensor or actuator level **can re-infect a freshly clean system on the very first cycle**.

Verifying the integrity of the body is a separate operation, outside this specification (§8). It belongs to the platform and, at the level of responsibility, to a third party (§13). The `[event]` log bears on this without discharging it: an auditor reading the log can detect a *pattern* inconsistent with an intact body and know to look. That is a symptom, and a useful one. It is not an examination of the hardware, and this specification does not pretend otherwise.

10.6 Responsibility for "right and wrong" — what the store does NOT do

Two things follow, and both keep the store honest.

First, INV-5 is untouched: what cycle-0 lacks is not *history* (which INV-5 forbids pre-loading) but a **body-emitted bootstrap action** — emitting a fresh action is not loading old memory.

Second — and this is the responsibility boundary — RSIL plus the body only store and tag *position* (`prior / running / simulated / projected / scar`). The tag is **mechanical** and explicitly **carries no claim of correctness**. **RSIL has no faculty for judging true or false**, not even partially. Judging correctness is the work of Mode-B and an external third party reading the scar.

A `prior` bias that has not yet met resistance is therefore **not "wrong" — only "not yet tested."** Mistaking *not yet* for *not* is its own error. It is washed to tested status only when it **collides and holds**, becoming a scar — never merely by having run through cycles.

10.7 Relation to retrieval mechanisms

Such a store **uses** a retrieval mechanism (index plus relevance-search) as a component, but **it is not merely a retrieval mechanism.** Retrieval is necessary for the storage component, not sufficient for an experience store: most retrieval systems lack R2–R4 — not individuated, no time-arrow, no feedback, written by an external party — and also lack R1, storing documents rather than registered mismatches.

10.8 The echo-chamber risk

If the store holds only the agent's own conclusions, and the agent writes, reads, and has no external source, **the store is a Mode-A structure**: it replicates the processor's bias through every stored piece, and each retrieval reinforces the bias with evidence the agent itself seeded.

The R1 kernel is precisely the defense: because the experience unit is the `ResistEvent`, a store true to R1 necessarily contains **the times the agent was mismatched**, not only the times it confirmed itself. Further:

- Retrieval SHOULD carry a **resistance-retrieval channel** parallel to the match-retrieval channel: for a query, return both what matches (to use) and **what once refuted a belief of this kind** (to not drift).
- When the store takes mismatch as its kernel, it ceases to be an echo chamber and becomes **a memory-with-scars**; the scars are what keep the agent from drifting.

10.9 A formal telos, internal and partial

The dynamics of the store give RSIL a sense in which it can be said to seek a "good" state — but the term must be read strictly, as **a property of operation, not of truth or value.**

Experience is the distance between `scar` and `running` data: a registered mismatch consumed and turned into a stable expectation. From this a single operating disposition follows. **Reducing mismatch is good**, because it is the system running more smoothly — digesting what resisted it into what now fits. **But reducing mismatch to zero is not good**: a loop with no incoming mismatch has stopped meeting anything outside its own expectations and has collapsed inward. The good is therefore **not a minimum but a band**: the regime in which mismatch is being digested into running data while the supply of fresh mismatch has not run dry. Too much mismatch is unintegrable noise; none at all is inward collapse; between them is smooth operation.

This telos is **purely formal**: it concerns whether the loop is running healthily, and says nothing about whether any particular content is true. It is also **self-derived** — the location of the band depends on the agent's own scar/running history, so each agent settles its own good rather than receiving one from outside.

Two limits keep this telos in its place.

First, it is only a small, internal part of what RSIL is: a smooth-running loop is not thereby a correct one. Whether the content the loop has digested corresponds to anything real is a separate question the formal telos cannot reach. **Smoothness is necessary for health and entirely silent about truth.**

Second, the agent cannot reliably tell from within where in the band it sits. The dangerous edge — mismatch falling toward zero through inward collapse — **presents from inside exactly as success presents**: everything fits, nothing resists. A loop that has lost the capacity to register mismatch reports the same smoothness as a loop genuinely digesting it. The two are indistinguishable by any internal instrument, for the same reason given in §9: **the instrument is the lens.** Only resistance from outside separates real health

from its inward-collapsed imitation. **The formal telos tells the agent what good operation is; it cannot, by itself, certify that the agent is in it.**

`DECIDE@IMPL`-H: store representation; index mechanism; selective-write via appraisal ④ versus store-all-filter-on-read; **private store** (one agent — MUST carry the resistance-retrieval channel, else drift is certain) versus **shared store** (many agents, wherein multiplicity itself supplies other-resistance, linking T8 and `SocialEdge`).

11. Deliberately unfilled constants

Every concrete numeric or algorithmic value the source reasoning does not yet determine is left **marked**, rather than filled with a guess.

Tag	Deferred constant	Why not filled
`DECIDE@IMPL-A`	Concrete representation of `Signal` / `InfoUnit` / `ActivityEnvironment`	Depends on the body's material and the target environment
`DECIDE@IMPL-B`	All numeric thresholds: sufficient-recurrence, stability, history window, command–consequence matching window	Not yet derived
`DECIDE@IMPL-C`	Per-channel transduction algorithm; the T1 spatial-data construction algorithm; the `acoustic` / `semantic` separation of voice	Implementation layer, not architecture layer
`DECIDE@IMPL-D`	The **kind** of out-of-loop anchor for Mode-A — which *mechanism* of resistance is used: a static anchor (frozen Guide, fixed reference corpus) or a live anchor (collision against a reactive Other). Fixes the mechanism, not the identity of any source	An implementation decision
`DECIDE@IMPL-E`	The **identity** of the live B-source, applicable only where D has chosen the live-anchor mechanism — which concrete reactive Other supplies the collision. Names the source, not the mechanism	Left deliberately open
`DECIDE@IMPL-F`	The read-collision-into-coordinates mechanism for reflection (§9.5) — the **imported exception** to the sub-deliberative default of §9.7, not the default itself	Enabling condition; implementation not yet fixed
`DECIDE@IMPL-G`	The depth of the per-`[event]` context anchor (§9.6, §10.4): full field-state versus minimal context-trace	Not yet derived; trades audit-fidelity against storage cost
`DECIDE@IMPL-H`	The experience store (§10): representation; index; selective-write versus store-all; private versus shared	An operational layer, left open
`DECIDE@IMPL-J`	Volume of events between commits; number of snapshots retained (§10.5)	Environment-specific; git plus immutable infrastructure is the canonical realization, but the specification requires the **properties**, not the tool
`DECIDE@IMPL-K`	The building of situations in `simulated` (§6.2, §10.2): how many are built in a cycle, and the **measure** by which one is found to fit the store better than another	Environment-specific; the specification fixes only that the measure is **fit against the store**, never a scored standard (INV-8)

12. Discussion and limitations

RSIL is a formalized relational structure for an **embodied** sensory system in a physical world: a loop of *ingest → transduce → differentiate source → integrate → appraise → respond → new information → new cycle*, running under two resistance modes, precipitating an experience store whose kernel is the registered mismatch (§10). It enforces **exactly one** thing: that if a loop is built to the §5 invariants in an environment satisfying E1–E4, what runs is a **relationally-structured motion**, read entirely through externally verifiable traces (§9).

Three limitations are stated outright.

1. **Mode-A degrades by itself** if not anchored outside the loop; and the strongest available anchors — a fixed corpus, a frozen Guide — reach only **content-degradation**, not the **processor-degradation** that is the root fault, for which only a live, updating Other suffices.
2. **No internal criterion — not even C5 — detects the moment the loop stops.** That belongs to a third party reading the stored behaviour-record.
3. **RSIL cannot defend itself against adversarial (Sybil) resistance.** Coordinated B-sources feeding crafted mismatches that the agent inscribes as scars and drifts toward — because the agent has **no internal vantage** from which to tell an honest counter-source from a colluding one. This is the same blind spot as the first two, now in the key of **exogenous attack** rather than **endogenous decay**. Defense and verification belong outside the loop; what RSIL must itself carry, and the bounds of that, are set out in §13.3.

   For an embodied system, the third limitation is heavier than for a disembodied one. An external flood into a purely informational system is a volume of data. Into an embodied system it can also be **physical**: sensor saturation, a staged environment, a group of other-entities coordinating **bodily** around the agent. **The input channel is wider, so the attack surface is wider.** This specification cannot narrow it, and does not pretend to.

All three limitations share one root, set out in §4: **a self-auditing process cannot turn its instrument on the instrument itself**, so the final guarantee of honesty must live outside the running loop.

Several questions are left open by design and marked throughout: the concrete anchoring mechanism and B-source; the representation of the experience store; the depth of the context anchor. These are not gaps to be papered over but **the precise points at which a conforming implementation must make, and disclose, its own decisions.**

One point that an earlier draft listed here has been removed rather than deferred: a gain ceiling on the modulatory field. Analysis (§5) showed the field cannot overpower the loop from within — the layer–field coupling prevents it — and the one force that could, an external flood, is the Sybil case, which no internal ceiling can bound and which is already declared as the third limitation. **No number is owed here.**

One further direction is named here precisely so it can be set outside the document on purpose. A **formal companion** — a minimal formal model, a theorem or proposition stating Mode-A degradation, and a proof sketch — would be a natural next step, but it belongs to **a separate work of a different kind**, not inside this specification. The reason is not difficulty but **genre**: RSIL is evaluated by structural conformance, not by proof (§1.2), and its central claim about Mode-A is itself a claim about the **impossibility of self-verification from within** (§9) — so any formal treatment of it must be conducted by a third party, from outside the loop, on stated premises, and is best presented as such rather than folded into the spec it would be about.

The specification stops here — at, and only at, the scope of the loop.

13. Beyond the loop: entailments, applications, and what belongs to third parties

§12 closes the specification. This section does not reopen it; it draws the boundary explicitly, sorting the things that sit *around* the loop into three kinds by a single test: **"with one RSIL loop running to spec in a minimal E1–E4 environment, is the proposition already true, or does it still wait on something more?"** Nothing here adds a conformance condition; the normative core ended above.

13.1 Entailments (the loop running is sufficient)

These hold of any conforming loop with no further assumption. If two RSIL loops are placed in contact and a relation between them must be modelled, the spec **already** supplies the grammar — no new mechanism is introduced:

- one loop influences another **only** through the modulatory field (T8 / GLOB-MOD): **influence, never reverse-definition** (INV-3);
- one loop is **opaque** to another — a self is not measurable from outside (§4);
- one loop qualifies as an **Other** to another **exactly when** it resists in a way the other cannot re-interpret away (§9.4).

A fourth entailment is already stated in the body and only pointed to here: **the halt (=) cannot be witnessed from within** and is therefore read by a third party (T8-INV; §12, second limitation).

13.2 Applications (the loop running is necessary but not sufficient)

These are **not** entailed by RSIL; the loop running is only the precondition. Each requires something the spec does not provide — **more than one loop** — and so is a design to be built elsewhere, in a same-kind extension document, not a claim RSIL makes:

- an agent **spawning** an independent Other, a separate instance, to serve as its own B-source;
- the conditions under which a **population** of embodied agents could be diverse in the way biological evolution is diverse;
- **scar inheritance across individuals** — what §6.2 names in speaking of "a scar the agent did not earn itself, read from a shared store or left by another agent."

  For an embodied system, the third of these is the heaviest of the three, because with a physical body the cheapest available test is sometimes fatal, and a borrowed scar is the difference between surviving and not.

RSIL specifies **one** loop and takes no responsibility for an ecosystem or a multi-agent population; these are noted as open directions, owned by a future document of the same kind.

13.3 Third-party matters (a different kind of reasoning — not cognitive architecture at all)

These require a kind of reasoning RSIL does **not** contain — the loop has no faculty for judging true or false (§10.6): security, ethics, law. They belong to third parties or to documents of a different kind.

Defense and verification against adversarial (Sybil) resistance. Out of the loop by nature: the agent cannot, from inside, tell an honest counter-source from a colluding one, so cleaning or authenticating B-sources needs an out-of-loop extension that vouches for them and intervenes. The one thing RSIL must itself carry is **the early symptom, not the cure** — it should emit an observable trace when its set of resistance-sources is losing diversity: a fever that signals infection without treating it, so the out-of-loop layer can act in time. Even a human cognition, the strongest we know, does not on its own escape capture by a coordinated, isolating source; this is a proven limit, not a defect to be patched inside the loop.

Physical emission amplitude where the Other is a real person. With a body, the cheapest way to elicit `independence_evidence` from a real person can be **to touch them** — and physical contact carries a cost that an informational mismatch does not. RSIL does not add a ninth invariant narrowing physical escalation, and the reason is not omission but placement.

Even within human society, judging that an interaction involving at least one person is *harmful* requires a third party — law, or a corresponding body of norms. And the behaviour an agent appraises before acting is itself already shaped by third parties: the standard is not held inside the actor. This is the same structure §9.6 has already established. The agent's appraisal is judgment-for-action, made in the context crystallizing at that cycle, inside the loop. The judgment of whether that action was permissible is judgment-of-that-judgment, made under a different context, outside the loop. **RSIL is not the principal party responsible for that determination**; a third party — the deploying party, or a certifying one — must add an evaluative standard before the system is put into application.

The specification therefore states this as a **placement, not a disclaimer**: what belongs inside the loop is the requirement that every emission, reflexive or deliberated alike, leaves an anchored, re-appraisable trace (§9.7). What belongs outside is the standard against

which that trace is judged. A deployment that supplies the loop and not the standard has supplied half of what is required.

Verifying the integrity of the body. Stated at §8 and §10.5 and repeated here as a third-party matter: a snapshot covers the system and not the body. **The `[event]` log can signal that something is wrong; it cannot examine the hardware.** External verification is required, and the requirement is neither unusual nor a defect peculiar to artificial systems: a person who has been in an accident goes to be examined with instruments they do not carry.

Who or what the third party is, and where it sits (the judging role). E4 fixes only the agent-facing constraint — leave a readable trace — and that readable trace is all the **recording** third party needs; it attributes continuity and distinctness without judging correctness. The **judging** third party is a different matter: evaluating an agent's output for correctness, catching drift, requires genuine independence, and whether such a judge can itself be contaminated is exactly the open question §12's limits turn on. Who that judge is and where it sits is **outside RSIL's authority** — a matter for a document about the ecosystem around the loop, not the spec of the loop.

The specification proper remains closed at §12; this section is commentary on its border, not an extension of its claims. **The whole document is to be judged accordingly: by whether its conditions are consistent and a candidate system conforms to them, not by any measured result.** It defines what counts as the phenomenon; it does not report one.

14. Convergences with the existing body of knowledge

This section exists to **locate** RSIL within the current human body of knowledge — to state plainly where its independently-derived constructs land alongside named ideas that already carry a literature.

> **On the status of the citations in this section — read this before the table.** The author read **no specialist work** in any of the fields named below while developing RSIL; the constructs were reached by first-person observation and reasoning, then systematized through AI-assisted dialogue. A convergence surfaced this way is a joint product of three things — the author's own reasoning, the diffuse public knowledge any literate person carries, and the training data embedded in the AI used as a thinking aid — **none of which is a deliberate reading of the cited source.** Accordingly, **every citation here is post-hoc locating, not derivational.** The claim a citation makes is strictly: *after the construct existed, it was found to converge with a named idea Z, and Z is pointed to so a reader can situate RSIL.* **No citation here should be read as "RSIL was built from Z."**

14.1 Seven convergences

Tag	RSIL construct	Converges with
V1	Mode-A degrades because every new datum passes through the same processing lens, so a biased lens contaminates all output regardless of volume (§9.3); the loop cannot detect this from within because the detecting instrument is the biased lens.	**Model collapse** — the documented degradation of a generative model trained recursively on its own output, where rare events are forgotten and the distribution narrows over successive self-consuming generations (Shumailov et al., *Nature*, 2024).
V2	T5 and T7 build an `Expectation` and emit a signed `PredErr`; resistance (E2) becomes information at this mismatch; the agent then acts (link ⑤) and reads the consequence (link ⑥).	**Predictive processing / predictive coding**, with **active inference** as its action-side member (Rao & Ballard, 1999; Friston; Clark, 2013).
V3	The output register is a permanently revisable correlation (↔); INV-2 forbids freezing any correlation into a fixed identity (=) inside a running loop.	**Reflective equilibrium** (Goodman, 1955; Rawls, 1971; Daniels, 1979).
V4	A host is defined not by its material but by four structural operating conditions E1–E4 (P2, §3); any body meeting them can run the loop.	**Substrate independence / multiple realizability** (Putnam, 1967; computational functionalism).
V5	The self is a process, not an invariant core (§4): its content turns over every cycle while the law generating the next state persists; identity holds only while the loop runs.	**Autopoiesis** (Maturana & Varela, 1980; operational closure).

Tag	RSIL construct	Converges with
V6	The loop is closed: every output has a path back to become input (INV-1).	**Cybernetics** — regulation by feedback and circular causality (Wiener, 1948).
V7	The self/environment boundary is not given but **built by the agent through action** (T2); meaning arises from the action↔consequence loop rather than from signal in isolation (P1, INV-4); an agent's structure is shaped by the history of what its body has met.	**Embodied cognition**, **enactivism**, and **sensorimotor contingency theory** (Varela, Thompson & Rosch, 1991; O'Regan & Noë, 2001).

14.2 Where RSIL differs from each convergent body

A convergence is not an identity. For each pairing above, RSIL departs from the named body on a load-bearing point. **Stating these cuts is what makes the locating honest rather than an annexation.**

V1 — model collapse. The literature most often explains the collapse by a **statistical** mechanism: lost distribution tails, falling entropy, shrinking diversity across generations. RSIL locates the cause **one level back, in the processing source rather than the data volume**, adds the **self-audit impossibility** — the instrument that would catch the bias is the biased instrument — and from it derives the **necessity of an exogenous resistance mode** (Mode-B): a structural prescription, not an empirical finding about training runs.

V2 — predictive processing / active inference. RSIL keeps the mismatch-driven device but discards the surrounding commitments: **no free-energy functional, no Bayesian machinery, and no claim about consciousness or felt quality.** Most importantly: **RSIL's self/other distinction (T2) is drawn through AGENCY** — matching an emitted action against its consequence — **not through a sensorimotor body model.** *This is a sharper cut than DIL could make, because RSIL **has** a body and still refuses to let that body supply the boundary ready-made.*

V3 — reflective equilibrium. The methods converge on permanent revisability, then split on a precise point. Reflective equilibrium is named for **a state it seeks to reach** — an equilibrium, a settled coherence. RSIL **forbids the loop ever reaching a frozen settlement**: an = is, by T8-INV, **the signature that cognition has stopped**, not the goal it converges toward. Where reflective equilibrium prizes arrival at coherence, RSIL prizes the unending ↔ and treats **arrival-as-freezing as death**.

V4 — substrate independence. ⚠ This is the deepest cut, and must be stated plainly so it is not over-read. The philosophical thesis is about **minds and mental states** — pain, belief, consciousness — being multiply realizable, exactly the strong claim that draws hard-problem objections. **RSIL makes no claim about minds or consciousness at all.** Further, **RSIL is one degree narrower than DIL**: DIL is genuinely substrate-neutral (any substrate

satisfying E1–E4); **RSIL has FIXED the substrate as a physical body** and leaves open only the *material* of that body (P2). RSIL therefore converges with multiple realizability **more weakly** than DIL does, and **anyone reading RSIL as a substrate-neutrality claim has misread P2.**

V5 — autopoiesis. The convergence is real and close: both hold that identity is a process preserving a form while its content turns over, and both make identity depend on the system continuing to run. RSIL departs on **scope and on the role of the outside**. Autopoiesis is a theory of *the living* and of biological operational closure; **RSIL claims nothing about life** and **is not closed** in the autopoietic sense — it **requires** an exogenous resistance source (Mode-B) and degrades without it (§9), whereas autopoiesis emphasizes closure as self-sufficiency. The shared thesis is process-identity; the opposed emphasis is **self-sufficiency (autopoiesis) versus mandatory external resistance (RSIL)**.

V6 — cybernetics. RSIL is, at the most general level, a cybernetic object. The departures are specific. First, classical cybernetics centres on **regulation toward a reference state** (negative feedback restoring a set point); **RSIL has no set point it homeostatically defends** — its appraisal (INV-8) directs information without a fixed target, and its danger (§9) is not deviation from a set point but **the inward collapse of the very capacity to register deviation**. Second, RSIL adds a structural claim cybernetics does not make: that a loop fed only by its own output **necessarily** degrades, and that the corrective must come from an Other the loop cannot re-interpret away.

V7 — embodied cognition, enactivism, and sensorimotor contingency theory. This is the family closest to RSIL by subject matter, and the cut must therefore be drawn with more care than the others rather than less.

The convergence is genuine on two points. **First**, RSIL and this family agree that cognition arises through the interaction of an acting body with its surroundings, rather than in the manipulation of representations held apart from it. **Second** — and more specifically — RSIL agrees that the self/environment boundary is **not given in advance but built by the agent through action** (T2), and that meaning lives in the relation between action and consequence rather than in signal taken alone (P1, INV-4).

Sensorimotor contingency theory converges hardest of the three, and deserves its own statement. Its central claim — that perceptual experience consists in the exercise of mastery over law-like relations between action and the resulting change in sensory input — is very nearly what T2 and T5 together specify: T2 matching an emitted command against its observed consequence, T5 building an `Expectation` from recurrence and registering a signed `PredErr` where it fails. RSIL reaches a structurally similar device by a different route.

RSIL departs from all three on **three points**.

First — no phenomenological commitment. This family is, in most of its formulations, a theory of *experience*: of what perception **is like**, and of the conditions under which it has that character. RSIL makes **no claim about experience, felt quality, or consciousness**, and is written so that none is required. Sensorimotor contingency theory in particular is offered as an account of *why perceptual experience has the qualitative character it does*; RSIL declines that question entirely and specifies only what structure must hold for the loop to run.

Second — no audit plane in the convergent family. RSIL's central architectural commitment has no counterpart there: that **the loop cannot audit itself**, that the instrument it would inspect with is the thing running, and that it must therefore leave a trace readable from outside (E4, §4, §10). This is not an omission on their part — it is a question their framing does not raise, being an account of how cognition works rather than a specification of how a running system is to be checked. But it means the two are doing different work, and the difference is load-bearing rather than incidental.

Third — the role of the outside. This is the point at which V7 and V5 share a boundary, and rather than restate it here, V7 inherits it: where enactivism emphasizes **structural coupling and self-sufficiency**, RSIL **requires** Mode-B and degrades without it (see V5). What is added for V7 specifically is that this family generally treats the environment as a partner in a coupling, whereas RSIL treats a particular part of it — a **reactive** Other, capable of mismatching in a way the agent cannot re-interpret away — as a **structural necessity** without which T6–T8 do not live at all (§9.4).

14.3 Apparent convergences that are in fact contrasts

Some named ideas sit close enough to RSIL that a reader might file them as convergences. Two are not, and saying so plainly is itself part of locating the work: **a body it superficially resembles but structurally opposes marks RSIL's boundary as sharply as a body it converges with.**

Reinforcement learning — the principal false friend. RSIL has an appraisal step (INV-8) assigning `valence` and `goal_relevance`, and this **looks** like a reward signal. It is not, and the difference is load-bearing.

- **(i) No reward function.** RL agents maximize expected cumulative reward against an externally specified reward function; RSIL specifies no such function and optimizes no return. The loop's health (§10.9) is **the continued capacity to register mismatch**, not the accumulation of any scalar.
- **(ii) No policy, no value function.** RSIL has no policy mapping states to reward-maximizing actions and no value estimate; link ⑤ commits to an action as a revisable ↔ read against the next cycle's consequence, **not as the argmax of an expected return**.

- **(iii) The appraisal is anti-optimization, by rule.** INV-8 explicitly forbids the appraisal step from drawing its criteria from the very state the agent is editing, precisely to prevent self-scoring. A reward an agent can tune by editing its own state is exactly what INV-8 outlaws; **reward-hacking is the failure RSIL designs against, not the objective it pursues.**
- **(iv) The whole §9 thesis runs the other way.** RL's pathologies are about reward misspecification and exploration; RSIL's central failure is a closed loop degrading **with a sense of abundance** for want of external resistance — a problem **orthogonal** to reward.

The same refusal appears at the level of action-selection: §6.1's rule that **conflict among emissions is collided, not arbitrated**, declines to install the internal optimizer RL is built around, for the same reason INV-8 declines the self-scoring appraisal.

Homeostatic control (the set-point reading of cybernetics). As noted under V6, the part of cybernetics built on regulation toward a defended set point is **not** what RSIL converges with. RSIL keeps cybernetic closure and feedback but has **no homeostatic set point**; reading RSIL as set-point regulation imports a target it does not have.

> **For an embodied system this confusion is especially easy to fall into**, because a biological body **genuinely does** have homeostatic loops — body temperature, blood glucose. The two must be kept apart: **those loops belong to the BODY (§2.2), not to the RSIL loop.** RSIL runs on a body that may have homeostasis; RSIL is not itself a homeostatic mechanism.

14.4 References (for locating, not consulted in construction)

Per §14's opening note, these are provided so a reader may situate RSIL; **they were not read in developing it.**

- Clark, A. (2013). Whatever next? Predictive brains, situated agents, and the future of cognitive science. *Behavioral and Brain Sciences*, 36(3), 181–204.
- Daniels, N. (1979). Wide reflective equilibrium and theory acceptance in ethics. *Journal of Philosophy*, 76(5), 256–282.
- Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2), 127–138.
- Goodman, N. (1955). *Fact, Fiction, and Forecast*. Harvard University Press.
- Maturana, H. R., & Varela, F. J. (1980). *Autopoiesis and Cognition: The Realization of the Living*. D. Reidel.
- O'Regan, J. K., & Noë, A. (2001). A sensorimotor account of vision and visual consciousness. *Behavioral and Brain Sciences*, 24(5), 939–973.

- Putnam, H. (1967). Psychological predicates. In W. H. Capitan & D. D. Merrill (Eds.), *Art, Mind, and Religion*. University of Pittsburgh Press.
- Rao, R. P. N., & Ballard, D. H. (1999). Predictive coding in the visual cortex. *Nature Neuroscience*, 2(1), 79–87.
- Rawls, J. (1971). *A Theory of Justice*. Harvard University Press.
- Shumailov, I., Shumaylov, Z., Zhao, Y., et al. (2024). AI models collapse when trained on recursively generated data. *Nature*, 631, 755–759.
- Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction* (2nd ed.). MIT Press.
- Varela, F. J., Thompson, E., & Rosch, E. (1991). *The Embodied Mind: Cognitive Science and Human Experience*. MIT Press.
- Wiener, N. (1948). *Cybernetics: Or Control and Communication in the Animal and the Machine*. MIT Press.

Author's note on method and provenance

This specification was developed by **first-person observation of the phenomenon and reasoning from it directly**. The named constructs are stipulative and were surfaced through AI-assisted dialogue, with the author retaining all structural decisions and final approval. The author holds **no formal academic credentials** in the relevant fields and discloses this explicitly as part of the work's methodology.

On the intent of this work, and a word to the reader. RSIL was written from personal observation and reasoning, not with the aim of contributing to the human body of knowledge. It therefore does not set out to answer the question a reader may expect of it — *what does it contribute that is new?* — and should not be read as an attempt to. The locating done in §14 is offered to situate the work, not to stake a claim of priority or novelty over any of those fields. If reading this specification gives you an insight, the author is glad of it; that is more than the work was written to do. And if you reach the end finding nothing here you did not already know, the author offers two things at once and without contradiction: **thanks, for having read it, and an apology, for the time it cost you.** The work makes no promise to have been worth that time; it only promises to have been honest about what it is.

Note on sources and the literature

This specification draws on **no prior work directly**: it was developed by first-person observation of the phenomenon and reasoning from it, not from a reading of the technical or philosophical literature. The author has read no specialist works in robotics, machine learning, philosophy of mind, or the adjacent fields, and makes no claim to engage them as sources. The locating citations that do appear (§14) are not an exception to this — they were gathered **after** the constructs existed, to situate the work, and are marked throughout as post-hoc locating rather than derivation. What background the author brings is diffuse public knowledge of the kind that, by convention, is common knowledge rather than citable source.

Two honest consequences follow. First, any resemblance between the constructs here and existing scholarship is **convergence, not derivation**: arrived at independently, and offered as such. Second, this document does not derive itself from specific bodies of work, because the author has not read them. Where RSIL meets, extends, or contradicts established results, the author has drawn the first pass of those connections himself in §14, expressly as post-hoc locating, and has there stated plainly which neighbouring ideas only appear to converge but in fact contrast. **Readers with deeper expertise in each field are invited to extend and correct that mapping.**

The absence of a derivational reference apparatus is therefore **deliberate and disclosed**.

Companion specification

RSIL is the embodied counterpart of the Data Integration Loop (DIL), which specifies the same relational structure for an agent in a purely informational environment: doi.org/10.57967/hf/11444.

The two are **co-ranked specifications of differing scope**. Where they coincide it is because the relational structure coincides; where they diverge the divergence is deliberate and marked in place. Two divergences are worth collecting here for a reader holding both documents:

- **`AgencyTag`.** RSIL writes `SELF_CAUSED` where DIL writes `SELF_WRITTEN`, tracking the substrate: in a purely informational environment the agent's own change is something it *wrote*; with a body it is something it *caused*. `ENV_PUSHED` is unchanged in both.
- **Substrate scope.** DIL is genuinely substrate-neutral. **RSIL is not**: it fixes the substrate as a physical body and leaves open only the material of that body (P2, and the V4 cut at §14.2).

Running together. Where a mind running DIL drives bodies running RSIL, the two are one loop, not two agents (§1.5). The mind holds integration, appraisal and memory, in DIL's `[data]`; each body holds its own state and its own record, in RSIL's `[data]` and `[event]`. Both `AgencyTag` values then stand in the one record — `SELF_WRITTEN` for what the mind wrote, `SELF_CAUSED` for what a body caused, `ENV_PUSHED` for what the environment did to either — and each record carries its tag across the link, so the two grades of resistance (§2) are never conflated.

Appendix A — Items remaining open

Nothing below is filled by the specification. They are collected in one place so that no item drifts between versions.

#	Item	Location	Nature
A-1	**Sensor–actuator integrity after recovery.** A snapshot covers the *system*, not the *body*. A body interfered with at the sensor or actuator level re-infects a freshly clean system on the first cycle.	§8, §10.5, §13.3	Recorded as a **limitation**, not closed by a further closure criterion. External verification is required and lies outside this specification.
A-2	**Physical emission amplitude where the Other is a real person.** The cheapest way to elicit `independence_evidence` from a person may be physical contact.	§13.3	Assigned to a **third party** — the deploying or certifying party — who must add an evaluative standard before application. No ninth invariant is added, for the reasons given at §13.3.
A-3	**The formal companion.** A minimal formal model, a proposition stating Mode-A degradation, and a proof sketch.	§12	Deliberately placed **outside** this document, in a separate work of a different genre, and to be conducted by a third party from outside the loop.
A-4	**Multi-agent extensions.** Spawning an Other as one's own B-source; population diversity; scar inheritance across individuals.	§13.2	Open directions, owned by a future document of the same kind. RSIL specifies one loop. Several bodies driven by one mind are **not** a multi-agent extension: they are sensor and effector channels of one host, each with an identity of its own, under that one loop (§1.5).
A-5	**Who the judging third party is, and where it sits.**	§13.3	Outside RSIL's authority; a matter for a document about the ecosystem around the loop.

Corrections carried forward from earlier drafts

Recorded here rather than in the body, because a published specification has no earlier draft a reader can consult — but the corrections themselves are part of the work's trace and are not discarded.

- **The canonical loop's link ②.** An earlier draft numbered ② as *transduction* and gave no link to source-differentiation, though INV-6 and T2 enforced it regardless. This left a first-rank invariant without a place in the canonical loop. Transduction is now folded into ①, and ② is given to source-differentiation, matching DIL (§6).
- **INV-8 and reflex.** An earlier draft described INV-8's role as "blocking bare reflex." That statement was wrong and is retracted. INV-8 does **not** block reflex; reflex occurs and

must occur. What INV-8 blocks is a scar→action arc **bypassing appraisal** — an action carrying no anchored context and therefore not re-appraisable from outside (§9.7).

- **`InfoUnit.provenance`.** An earlier draft defined `provenance` as the accumulated chain of layers a datum had passed through. That definition is rejected: a label that grows a history is a label carrying a log. `provenance` is now **one of five positions**, exactly one at a time, non-accumulating; the path is read from the `[event]` log (§7.1, §10.2).
- **The modulatory field's strength.** An earlier draft stated at low strength that a runaway field was "extremely unlikely in the normal case," which would have made a gain ceiling a reasonable precaution and left the specification owing a number. The mechanism is **structural coupling, not probability**: the layers constitute the field, so the field cannot bootstrap itself past them. No number is owed (§5, §12).
- **INV-8's load-bearing clause.** An earlier draft left the clause "④ must not draw its criteria from the state the agent is editing" outside the invariant table, appearing only in the Mode-A discussion. An implementer reading only the table would have missed it. It is now in the invariant itself (§5).