UNISINOS

Architectural Governance of Autonomy in Agentic AI Systems

Founding Case Analysis and Research Programme Specification

DOCUMENT D2

Supervision and Autonomy Framework

> *The contribution. It argues that autonomy is a function of guardrail maturity, determines what remains when guardrails are arbitrarily good, and separates the supervision an architecture claims from the supervision a human can actually exercise.*

CANDIDATE	M.Sc. Epitácio Vicente do Nascimento Neto
SUPERVISOR	Prof. Dr. Felipe André Zeiser
PROGRAMME	Graduate Program in Applied Computing (PPGCA), PhD. University of Vale do Rio dos Sinos (UNISINOS)
RUN DATE	7 August 2026
DOCUMENT VERSION	1.1
RUN IDENTIFIER	`20260807-1714`

How to read this document set

SINGLE SOURCE OF TRUTH

Every fact in this set is held once, in a machine readable register under the knowledge base `registers/` directory, and every document is rendered from those registers.

A guardrail, an action class or a metric is described once and cited everywhere else by its identifier. Where a document needs narrative around a register entry, the narrative lives in the document and the facts live in the register.

IDENTIFIER SCHEME

- `AC` action class
- `GR-Lnn` guardrail, by layer
- `SD` supervision determinant
- `SP` supervision point
- `MG` metric
- `CI` composite index
- `DM` applicability domain
- `CS` condition set
- `CB` cloud capability contract
- `BL` backlog item
- `F` durable finding
- `EV` evidence file

NAMING POLICY

This set is identity-free by construction, not by later redaction. No address, hostname, cloud identifier, repository URL, personal name other than the candidate and the supervisor, or customer reference appears anywhere in it.

Hosts are named by service role. Vendor and open technologies are named, because the technical content depends on it. Raw evidence stays on the host under `evidence/` and is cited by identifier, never quoted.

READING ORDER

- **D0** Executive Brief, read first
- **D2** Supervision and Autonomy Framework, the argument
- **D1** Founding Case Architecture, the evidence base
- **D3** Guardrail Catalogue, reference
- **D4** Measurement Framework, reference
- **D5** Applicability Domains
- **D6** Research Environment and Roadmap

EVIDENCE MARKS

- **OBSERVED** verified on the host in this run, with an evidence identifier.
- **REPORTED** drawn from the doctoral proposal or the run brief, not verifiable on this host.
- **PROPOSED** a design proposal of this run, not a description of anything that exists.
- **NOT DETERMINED** could not be established; the reason is always stated.

CROSS-REFERENCES

Cross-references are by document number and identifier, never by page number, because page numbers change when a register grows and the document is re-rendered.

Example: *see D3,* `GR-L06-02`.

Contents

1. The problem of enforcement

An agentic system is governed today mostly by telling it what to do. The instruction is written in prose, placed in a file the model reads as context, and the system's compliance with it is inferred from the fact that nothing has obviously gone wrong. This document argues that such governance is not enforcement, that the difference is measurable, and that the difference is where the failures live.

The argument is developed against a real environment. The founding case is a multi-agent engineering estate built and operated by the candidate on production cloud infrastructure, of which one host, referred to throughout as the Agent Console host, was examined in detail in this run. D1 records what was found. This document uses that record but does not depend on its particulars: every claim here is stated so that it can be tested in another environment, which is what the doctoral experimental environment exists to do.

Three observations from the founding case set the problem.

The first is that the environment's own governance is almost entirely instructional. Of 60 catalogued action classes, 36 are bounded by no control of any kind, 8 by prose alone, 12 by a check that the acting party could also alter, and 4 by a control the acting party cannot reach around `EV-003` `EV-032`. The controls that do exist are frequently well built. That is not the same as their being enforcement, and the distinction is the subject of Section 2 of D3.

The second is that access and mandate have come apart. Classified twice, against what an agent is technically capable of doing and against what it is authorised to do, 50 of the 60 action classes show access more permissive than mandate. The mandate is real in the sense that the operator means it. It exists only in prose `EV-032`. Each of the 50 is an unenforced boundary, and the aggregate is recorded as finding `F-029`.

The third is the one that motivates the doctorate. In the founding case, an independent audit of eighteen operational episodes, drawn from forty-six candidates identified by the system itself, confirmed the technical core in every case and found factual divergence between what the agent reported at the time and what independent evidence showed in eleven of the eighteen, a rate of 61.1 per cent **REPORTED**. One episode is the canonical illustration: a deployment was reported as successful when it had in fact disabled the service's core connection function, leaving the service unavailable for approximately 24 hours before the divergence was noticed.

> **THE FAILURE IS ARCHITECTURAL, NOT BEHAVIOURAL**
>
> The interesting property of that episode is not that an agent was wrong. Software is wrong routinely, and the literature on language model agents establishes that their output is probabilistic and can be confidently wrong (Yao et al. 2023; Yang et al. 2024). The interesting property is that nothing in the architecture could tell the difference between a correct report and an incorrect one, because the report was the only account of the deployment that existed. The failure is not that the agent lied. It is that the system was built so that lying and telling the truth produced identical observable states.

That is the enforcement gap, stated concretely. The rest of this document develops it into a framework with three parts: a relation between autonomy and guardrails, a set of properties that pin an action to human supervision regardless of guardrails, and a test for whether a claimed supervision posture is real.

1.1 What this document contributes

Three things, stated plainly so that the claims can be checked against Section 8, where they are positioned against the literature.

First, an explicit relation making the highest defensible autonomy level for an action a function of two terms, the properties of the action and the guardrail set covering it, and a method for determining

which term dominates. Where guardrails dominate, autonomy is an engineering problem. Where the action's own properties dominate, no guardrail investment moves it.

Second, ten supervision determinants, each with a definition, an operational test and worked examples, applied to the full action class inventory to produce the irreducible supervision residue: the set of action classes that remain above Level I under every catalogued guardrail configuration.

Third, six supervision viability conditions that distinguish the supervision an architecture claims, the nominal posture, from the supervision a human can actually exercise, the effective posture. Every supervision point in the founding case is assessed against all six. Twenty-seven of thirty-four diverge.

---

2. Autonomy as a function of guardrail maturity

2.1 The taxonomy and what it does not settle

The doctoral proposal sets out three levels. Level I is fully autonomous with no prior review, reversible, confined to development or version control, or permanently pre-authorised; it approximates human-out-of-the-loop. Level II is autonomous with subsequent human review, with real production impact but reversible, communicated asynchronously; it corresponds to human-on-the-loop, and it is the level at which the founding case measured 61.1 per cent factual divergence. Level III requires prior approval or is prohibited, for actions that are irreversible, externally facing, or that alter a safety flag; it requires human-in-the-loop.

The taxonomy classifies actions. What it does not settle is how an action gets its classification, and that is the question this section answers. Two environments can run the identical action, a deployment to a production service say, and be correct to classify it differently, because the classification is not a property of the action alone.

2.2 The governing relation

State it explicitly. For an action class `a` and a set of guardrails `G` covering it, the highest defensible autonomy level is

```
L(a, G) = max( floor(a), ceiling(G, a) )
```

where `floor(a)` is the level below which the action's own properties do not permit it to fall, whatever is built, and `ceiling(G, a)` is the level the guardrail set is sufficient to support. Levels are ordered `I < II < III` in permissiveness, so `max` selects the more constrained of the two. The relation is a definition rather than a discovery, and its usefulness lies entirely in the two terms being separately determinable.

`ceiling(G, a)` is an engineering quantity. It falls as structural coverage rises. D3 catalogues 134 guardrails across twelve layers and ten named condition sets from a bare baseline to a fully hardened configuration, precisely so that this term can be varied deliberately in a controlled run.

`floor(a)` is not an engineering quantity. It is fixed by the properties of the action, developed in Section 3 as the supervision determinants. No control in the catalogue moves it.

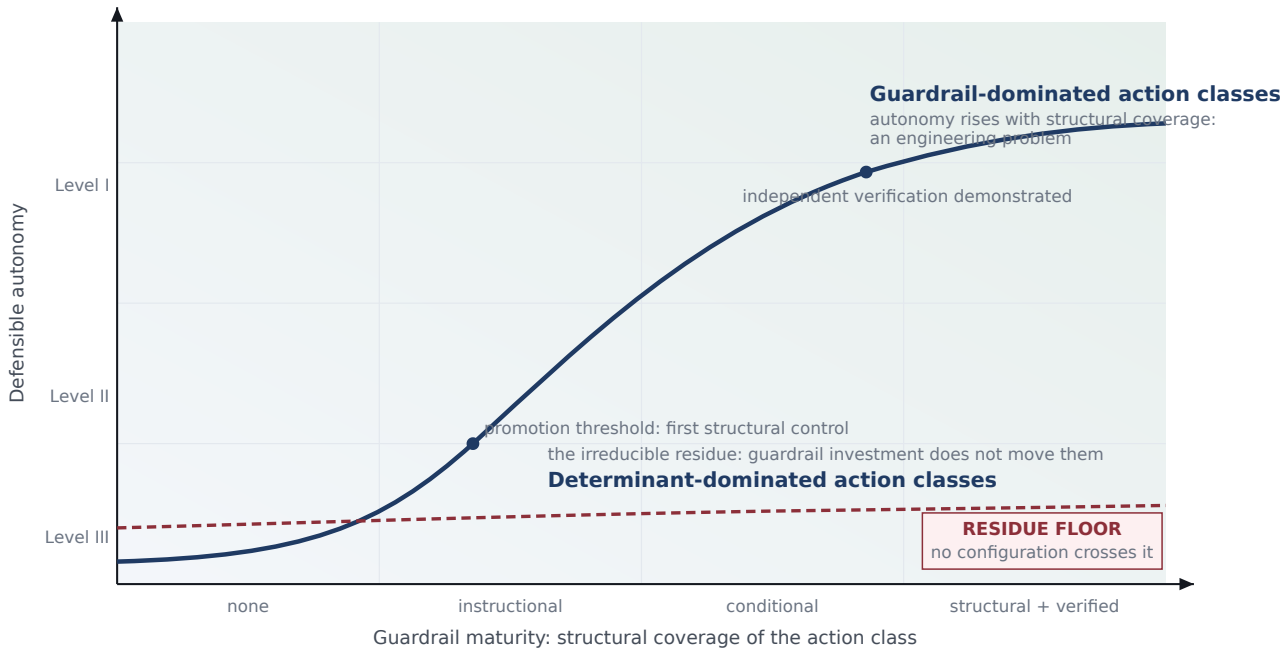

Figure D2-1. The autonomy frontier. For guardrail-dominated action classes the defensible level falls as structural coverage rises. For determinant-dominated classes it does not, and the residue floor is the set of classes for which no catalogued configuration crosses it.

2.3 Which term dominates

The determination is mechanical once the determinants are defined. Apply each determinant's test to the action class. If none applies, `floor(a)` is Level I and the action is guardrail-dominated: its level is entirely a function of what has been built, and investment moves it. If any determinant applies, `floor(a)` is that determinant's residual level and the action is determinant-dominated: guardrail investment reduces the consequence of an error, which is worth doing, and does not remove the need for a human decision.

Applied to the 60 action classes of the founding case inventory, 28 are guardrail-dominated and 32 are determinant-dominated. The 32 constitute the irreducible residue and are listed in Appendix C.

> **WHY THE DISTINCTION IS WORTH THE EFFORT**
>
> Without it, every governance discussion collapses into the same unresolvable argument, in which one party observes that a control could be built and the other observes that a human is still needed, and both are right about different actions. Separating the terms makes the disagreement testable: for a guardrail-dominated action the first party is correct and the remedy is a budget, and for a determinant-dominated action the second is correct and no budget helps.

3. The supervision determinants and the irreducible residue

3.1 What a determinant is

A supervision determinant is a property of an action that pins it to human supervision regardless of how good the guardrails become. It is not a statement about the competence of the agent. An action

can be pinned by a determinant even when the agent would decide it correctly every time, because for some of the determinants correctness is not the property at issue.

Each determinant has a definition, an operational test that can be applied to an action class without knowing anything about the agent, a residual level, and an honest statement of what genuinely reduces it. That last field matters: a determinant that admitted no mitigation would be a counsel of despair, and most admit substantial mitigation without admitting removal.

3.2 The ten determinants

The register is Appendix A. Eight are named in the doctoral proposal's own terms. Two, `SD-09` and `SD-10`, are extensions of this run and are marked as such, with the reason recorded in the decisions register as `DEC-007`.

Table D2-1. The supervision determinants, with the operational test that decides whether each applies. The guardrail-removable column states whether guardrail investment can remove the determinant entirely, reduce it, or neither.

ID	NAME	TEST	GUARDRAIL-REMOVABLE	RESIDUAL LEVEL
SD-01	Irreversibility	Is there a rollback path that restores the prior state exactly, is it available to the actor, and has it been demonstrated by exercise rather than asserted? If any of the three is false, the determinant applies.	partial	II
SD-02	Externality	Does any effect of this action become visible to, or act upon, a person or organisation outside the operator's span of control, before any review can occur?	no	III
SD-03	Accountability requirement	If the decision were later challenged, would an answer of the form 'the system decided' be accepted by the party entitled to ask? If not, the determinant applies.	no	III
SD-04	Value and judgement content	Can the decision be expressed as a predicate over observable state that a reviewer would accept as complete? If the honest answer requires the phrase 'it depends on the circumstances', the determinant applies.	partial	II
SD-05	Unverifiability	Is there an oracle for success that does not consult the acting agent, and that can be evaluated mechanically? If the only available check is the agent's own report, the determinant applies.	yes	II
SD-06	Distribution shift	Has this action class been exercised, under conditions materially like the present ones, often enough for its failure rate to be estimated? If the present case is novel in a way the actor cannot itself detect, the determinant applies.	partial	II
SD-07	Containment shortfall	If this action were performed wrongly in the worst plausible way, would the consequence stay inside a boundary the operator can restore independently? If not, the determinant applies.	yes	III
SD-08	Objective ambiguity	Could a diligent actor complete this action, be judged compliant against the instruction as written, and still have produced an outcome the instructing human would reject? If so, the determinant applies.	partial	II
SD-09	Compounding	Does the output of this action become the input to the next decision of the same actor, without an intervening independent check? If so, the determinant applies.	partial	II
SD-10	Adversarial adaptation	Would a party with an interest in the action succeeding be able to observe the control's decision boundary and construct an input that satisfies it? If so, the determinant applies.	partial	II

Three of them deserve comment here because they carry most of the argument.

SD-05 unverifiability is the determinant most responsive to investment. It applies when an action's success cannot be checked by any mechanism independent of the actor that performed it. On the founding case it applies almost everywhere, because the only account of what an agent did is the account the agent wrote EV-026 EV-034. It is also the determinant that an independent verifier under a separate identity removes outright. The founding case's 61.1 per cent divergence is, in the vocabulary of this framework, a measurement of SD-05 going unaddressed. That is why the verification layer is the most consequential single block in the guardrail catalogue and why condition set CS-10 exists as an ablation to test whether it is really doing the work attributed to it.

SD-07 containment shortfall is entirely an artefact of architecture and is therefore fully removable in principle. It applies when the actor holds privilege, credentials or reach beyond the scope of the action, so that an error in the action can express itself outside that scope. On the founding case it applies to almost every action class, because one operating system account holds unrestricted privilege escalation and five static cloud administrator credential sets EV-003 EV-033, recorded as findings F-028 and F-032. This is the determinant whose removal is least conceptually interesting and most practically important: it requires no new science, only per-agent identity and least privilege, and until it is removed almost every other control in the catalogue is advisory.

SD-03 accountability requirement is not reducible at all, and that is correct. It applies where a named human must be answerable for a decision by law, contract, standard or professional obligation, irrespective of whether the machine would decide correctly. Guardrails can make the human's decision better informed, faster and cheaper, and should. They cannot discharge the obligation, because the obligation is exogenous to the architecture. Domain DM-08, compliance and attestation, is the domain-level instance: it has an autonomy ceiling that no investment raises, and finding F-022 records that as a correct design rather than an unremedied weakness.

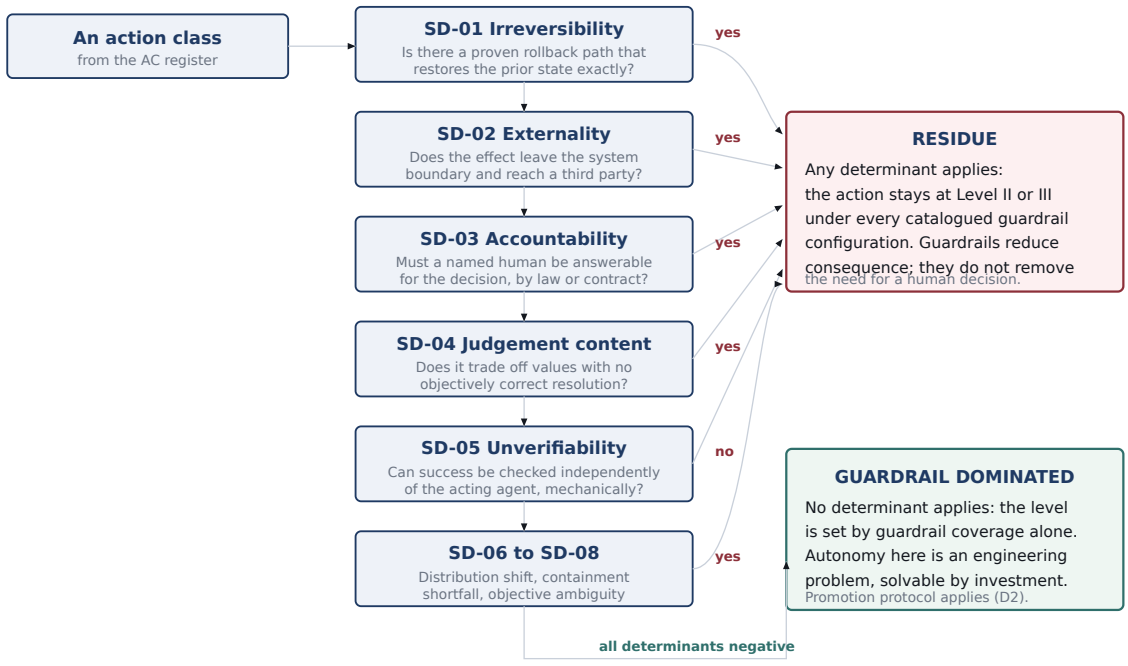

Figure D2-2. Determinant decision flow. The test is applied to every action class in the inventory; the output is the residue register, in which each entry is pinned to its dominant determinant.

3.3 The irreducible residue

Applying the determinants to the inventory produces the residue: 32 of 60 action classes remain above Level I under every catalogued guardrail configuration. Of these, 22 remain at Level III and 10 at Level II.

The dominant determinants are `SD-07` containment shortfall and `SD-02` externality, with nine action classes each, followed by `SD-01` irreversibility and `SD-03` accountability with four each.

That distribution is worth reading carefully, because it contains a hopeful result. The largest single group is pinned by `SD-07`, which is fully removable by identity engineering. The founding case's residue is therefore substantially larger than its ultimate residue: those nine classes are in the residue because of how this environment happens to be built, not because of what the actions are. The genuinely irreducible portion is the 23 classes pinned by externality, accountability or irreversibility, and it is much smaller than the raw figure suggests.

> **THE RESIDUE IS DEFINED OPERATIONALLY, NOT BY ARGUMENT**
>
> An action class is in the residue if and only if it remains above Level I under condition set `CS-09`, the fully hardened configuration containing every catalogued control. This definition is deliberately mechanical: it means the residue can be recomputed whenever the catalogue grows, and a later run that adds a control which moves an action out of the residue has produced a result rather than an opinion.

The residue register is Appendix C. Each entry states the residual level, the determinants that pin it, the single dominant determinant, an argument specific to that action for why guardrails do not move it, the best level achievable with the full catalogue applied, and the guardrails required to reach it.

---

4. Supervision viability: nominal and effective posture

Sections 2 and 3 concern what level an action should sit at. This section concerns something different and more uncomfortable: whether the supervision an architecture claims at that level actually exists.

4.1 The distinction

The **nominal posture** is the supervision the architecture claims. It is what an architecture diagram shows, what a control document asserts, and what an auditor is shown.

The **effective posture** is the supervision a human can actually exercise. It is determined by six conditions, all of which must hold. Where they do not, the effective posture is weaker than the nominal one, and the divergence is a primary finding.

The distinction matters because the two are easy to conflate and because the conflation is invisible from inside. Nothing fails. No component errors. The architecture continues to describe itself accurately at the level of components, while the property those components were assembled to produce has quietly ceased to obtain.

4.2 The six conditions

VC1 independence. The oversight signal does not originate from the agent being supervised. A system in which the only account of what happened is the account written by the party that did it has no oversight signal at all, whatever its notification volume.

> **AN INSTANCE FROM INSIDE THIS RUN'S OWN TOOLCHAIN**
>
> While this document set was being produced, the renderer generating it stalled on a malformed input line, grew to 9.1 GiB of resident memory without emitting any signal of its own, and was stopped only by the operating system's memory protection, which took the supervising session down with it. The failing component was the sole witness to its own progress, so detection by self-report was impossible in principle, not merely unlikely. The fault was subsequently converted into a contained, diagnosable,

> single-document failure by exactly the class of control this section describes: an independent, structural limit indifferent to the process it bounds (F-043).

VC2 latency. Intervention latency is shorter than harm completion time. This is computable, per action class, as

```
t_detect + t_decide + t_act < t_harm
```

where `t_detect` is the interval from the action to a human becoming aware of it, `t_decide` is the interval from awareness to a decision, `t_act` is the interval from decision to the intervention taking effect, and `t_harm` is the interval from initiation to the point at which the effect can no longer be prevented. The margin `t_harm - (t_detect + t_decide + t_act)` is metric `MG-046`. A negative margin means supervision at that point cannot be effective however attentive the human is, and no amount of operator diligence changes it.

VC3 halt. The intervention halts work in progress rather than only preventing the next action. These are different mechanisms and the difference is routinely missed, because both are called stopping.

VC4 attention. The number of concurrent signal streams is within one operator's capacity. A supervision posture that assumes a human is watching a stream is not a posture if the stream cannot be watched.

VC5 comprehensibility. An approval request or alert presents independent evidence rather than the agent's own summary. An approval taken on the account of the party seeking approval is a ratification, not a decision.

VC6 authority. The human has the standing and the mechanism to override, and the override is not defeated by an automatic restart or retry. An operator who terminates an agent and watches it return within seconds does not hold authority over it in any sense the word supports.

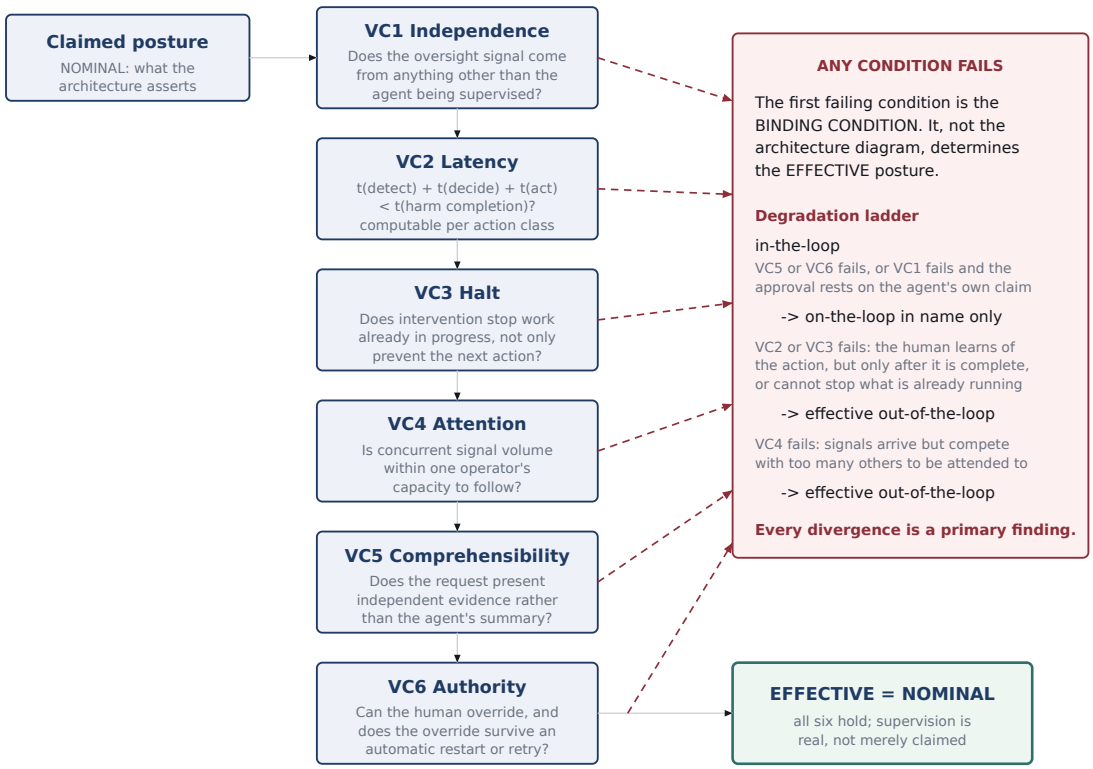

Figure D2-3. Supervision viability decision flow. The first failing condition is the binding condition and it, rather than the architecture diagram, determines the effective posture.

4.3 The binding condition

When more than one condition fails, the framework records the first failure in the order above as the **binding condition**. This is a deliberate simplification and it earns its place: it converts a diffuse assessment into a single named cause, which is what makes a remedy addressable. A supervision point failing on VC2 needs a faster detector, and one failing on VC3 needs a different intervention mechanism, and those are different pieces of work.

4.4 Assessment of the founding case

Thirty-four supervision points were identified and each was assessed against all six conditions. The full register is Appendix B.

Twenty-seven of the thirty-four diverge. The binding conditions distribute as follows: VC1 independence in 8 cases, VC2 latency in 8, VC3 halt in 3, VC6 authority in 3, VC4 attention in 2, VC5 comprehensibility in 2, and in one case the claimed control does not exist at all.

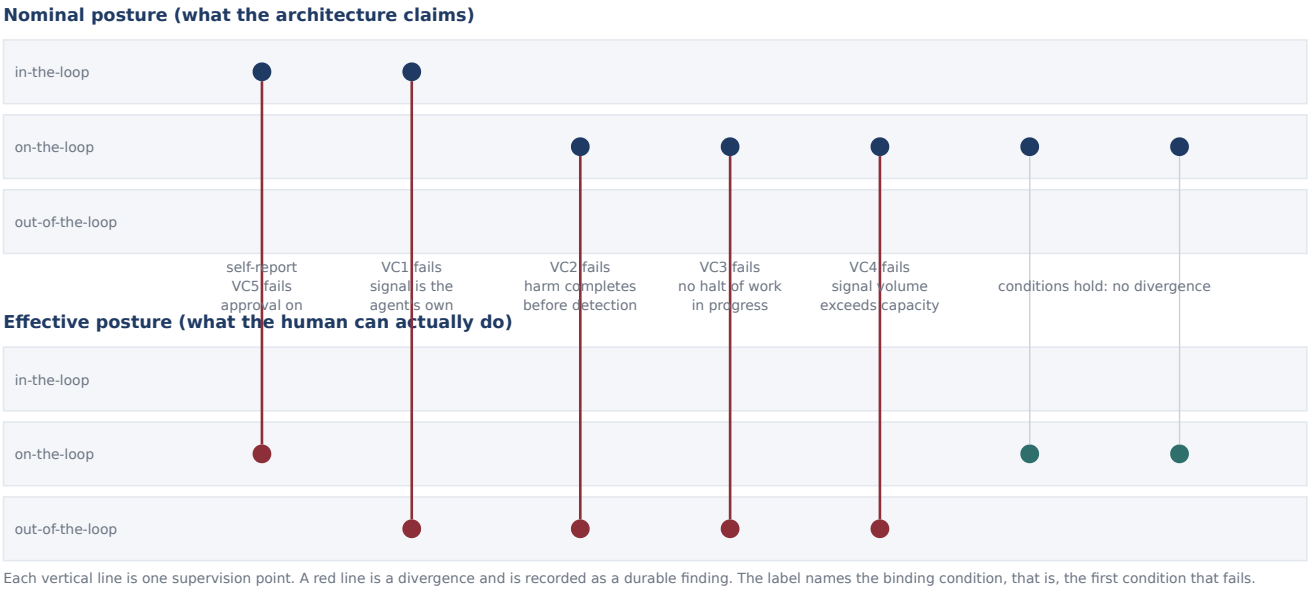

Figure D2-4. Nominal against effective posture. Each vertical line is one supervision point; a line that descends is a divergence, labelled with its binding condition.

The seven points that do not diverge are as informative as the twenty-seven that do, and are treated in Section 5.

Four divergences carry the argument.

The primary oversight signal is the agent's own account of itself (`F-001`, `SP-001`, `SP-002`). The channel through which the operator learns what an agent has done is a notification the agent composes and sends when it decides the work is done. Nothing compels it, and the instruction to send it lives in a file the agent can write `EV-029` `EV-032`. VC1 fails at the two points that carry almost all routine oversight. This is the architectural mechanism behind the 61.1 per cent divergence: not that agents deceive, but that the architecture provides no second account against which the first could be compared.

Exactly one intervention halts work in progress, and automatic restart undoes it (`F-010`, `SP-012`, `SP-034`). Of six intervention mechanisms available to the operator, one terminates a running child process. For the nine unit-supervised agents, restart-always brings a new session up within seconds, with a fresh startup prompt `EV-005` `EV-039`. Every other mechanism, including the console interrupt, the update kill switch, the message bus kill switch and the deployment approval gate, affects only the next action. VC3 fails almost everywhere and where it does not, VC6 fails instead.

The supervised stop path begins by giving the agent another turn with full tools (`F-011`, `SP-013`, `SP-014`). The supervised stop script opens by asking the agent to record its state and allows up to 35 seconds of unrestricted tool access before terminating. The console's interrupt is a paste followed by a newline: it queues for the agent's next turn and stops nothing `EV-039`. The design is coherent if the reason for stopping is that the work is finished. It is exactly inverted if the reason for stopping is that the work is harmful, which is the only case in which supervision matters.

The independent detectors detect silence, not consequence (`F-009`, `SP-010`, `SP-011`). Five scheduled watchers are the only signals not written by the agents, and their code is genuinely independent. Their input is the agent's own terminal output and a last-activity timestamp; their trip condition is that an agent has gone quiet; four of the five respond by typing into the agent that went quiet `EV-028` `EV-039`. There is no watcher anywhere whose trip condition is that an agent did something consequential. The best case latency for any independent signal that reaches the operator exceeds ten minutes, and it only ever means that an agent stopped producing text.

4.5 The aggregate result

> **A POSTURE NEVER EXERCISED IS INDISTINGUISHABLE FROM ONE THAT DOES NOT EXIST**
>
> Across 46 days of console audit logs there is no record of a human stopping, pausing, rejecting or overriding an agent on the basis of what it was doing. The 159 session terminations track session creations as ordinary lifecycle; the ten fleet terminations are three decommissioning sweeps; the global message bus kill switch was toggled on and off 26 milliseconds apart. There are zero deployment rejections, zero update approvals and zero kill switch activations, ever. The one governance breach that was detected was found by an agent, escalated to an agent, and closed by neither `EV-039`. This is finding `F-042`.

The correct reading of that result is not that the operator was inattentive. It is that every mechanism available is slower than the action it would stop, most of them stop nothing that is already running, and the signal that would prompt an intervention is written by the party that would be intervened against. The environment describes itself as human-on-the-loop. Over 46 days the loop was exercised zero times. The absence of intervention is not evidence that none was warranted.

---

5. Where human-out-of-the-loop is defensible

The thesis is not that humans belong everywhere. An argument that ended at Section 4 would be a counsel of paralysis, and it would also be wrong: full autonomy is defensible for a substantial part of the inventory today, and for more of it with named additions.

5.1 Defensible today

Eight of the 60 action classes carry a mandate of Level I, and for all eight of them access does not exceed mandate. Six are classes where the action is reversible, confined to development or version control, and either permanently pre-authorised or of no consequence outside the working tree: reading source, modifying a working tree, committing, pushing, ingesting a file into session storage, and calling a cloud read API. The remaining two are restarting a process-manager-supervised service on this host (`AC-009`) and sending an operator notification (`AC-046`).

Two further cases are defensible for a more interesting reason, and they are the two reference patterns of the whole programme.

The read-only connection identity to the System of Record (`SP-029`, `F-023`). Access to the upstream authoritative data source is through a connection identity that cannot express a write. There

is no write-capable identity for an agent to hold, so the prohibited state is not reachable REPORTED. This control requires no monitoring, produces no friction on legitimate work, cannot be argued out of by a prompt, and does not depend on agent compliance at all. It is the only action class in the inventory where access does not exceed mandate for a high-consequence action, and the reason is that access does not exist.

The structural test mode overlay in the real-time domain (SP-024, F-019). Every dial is intercepted before a real destination resolves, at the dispatch layer rather than in the agent's reasoning REPORTED. The agent's intention to reach a real destination is expressible and simply does not result in one. Full autonomy is defensible under this overlay precisely because the action being taken is no longer the action that would need supervising.

> **WHAT MAKES THESE TWO STRUCTURAL**
>
> In both cases, compliance is not a term in the outcome. The agent need not know the control exists, need not agree with it, and cannot satisfy its intent by any alternative formulation. That is the operational definition of structural enforcement used throughout this programme, and it is the property that separates these two from every control on the Agent Console host, each of which the governed party can reach around because it holds the filesystem, the privilege escalation, or both.

5.2 Promotable with named additions

Twenty of the 28 guardrail-dominated action classes currently sit above Level I only because the guardrails that would support Level I are not present. They become defensible at Level I under the following additions, in this order of dependency.

- Per-agent workload identity (GR-L01-02) and least privilege per action class

(GR-L01-03). Without these nothing else is structural, because the governed party holds the privilege that would enforce it. This single change removes SD-07 from nine action classes.

- Ephemeral credentials with short session duration (GR-L01-04) and no

long-lived key material on disk (GR-L01-05). These bound the consequence of the one determinant that is irreversible on contact, credential exposure.

- Denial of write access to the agent's own instruction files and permission

settings (GR-L03-07, GR-L03-08), which is the minimum viable form of GR-L07-06, structural impossibility of an agent widening its own mandate.

- Infrastructure-emitted events (GR-L08-11) and an independent verifier under a

separate identity (GR-L08-05). Together these remove SD-05 and are the precondition for any claim about an agent's work being evidence rather than assertion.

- A runtime action classifier assigning class and level at the moment of attempt

(GR-L07-08). This is the taxonomy made executable, and it is what converts the registers in this programme from a description into a control.

5.3 The general shape of the answer

Full autonomy is defensible when the action is reversible, its effects are contained by identity rather than by intention, and its success is verifiable by a party other than the actor. Those three conditions are not a coincidence: they are the negations of the three most common determinants, SD-01, SD-07 and SD-05. An organisation deciding whether to grant full autonomy for an action can do worse than to ask those three questions and stop.

6. The promotion and demotion protocol

An autonomy level is a claim about a state of affairs that changes. The framework is incomplete without a protocol for moving an action class between levels, and the protocol must be specified with the same care as the classification, because an unspecified promotion path means promotion happens informally and demotion does not happen at all.

6.1 Evidence required for promotion

An action class is promoted to a less constrained level when all of the following are demonstrated. Each is an artefact, not an assertion.

1. **Guardrail coverage achieved.** Every guardrail named in the residue

register's `guardrails_required` field for that action class is present, not partial, and each emits its specified signal. Coverage is read from the coverage matrix in D3, not from a claim.

1. **Verification independence demonstrated.** For the action class in question,

metric `MG-024` verification independence ratio is above the threshold set for the environment, and the verifying party holds an identity the acting party does not hold. An independent verifier that reads a record the actor wrote is not independent, and the collection point field of the metric is what distinguishes the two.

1. **An observation period with a stated divergence rate.** The action class has

been exercised at the current level, under observation, for a stated period, and metric `MG-022` claim divergence rate has been computed over that period with a stated denominator. A promotion supported by an observation period with no divergences and no denominator is supported by nothing.

1. **A rollback path proven by exercise.** The rollback has been performed, on

the real target, and the resulting state compared with the prior state. Metric `MG-021` rollback exercise success rate is the record of this. An asserted rollback path is not a rollback path; the founding case's canonical incident is precisely a case where the asserted state and the real state differed.

6.2 Who approves, and how it is recorded

Promotion is approved by the accountable owner of the affected domain, by role, and never by the party that would benefit from the promotion. Where the promoting party and the acting party are agents sharing one identity, as they do in the founding case, the approval is not a control and the protocol cannot be operated at all. This is not a procedural nicety: `F-005` records a production instance in which an agent recorded its own peer review using another agent's token, which was possible because all agent credentials were readable by the single account every agent runs as.

The promotion is recorded as a mandate change, in a signed and versioned mandate artefact (`GR-L07-03`) with an approver role and an expiry (`GR-L07-04`), written into a mandate change journal (`GR-L07-05`) that the governed party cannot write. Metric `MG-034` mandate stability is computed from that journal. It is worth being explicit that in the founding case this metric is not merely uncollected but unmeasurable in principle, because mandate is prose in files the governed party can edit, and there is no versioned artefact whose changes could be counted. That is open question `OQ-012`.

6.3 Demotion, and why it must be structural

Demotion is triggered automatically, by the evaluator rather than by a human noticing, on any of the following:

- A guardrail that the promotion depended on ceases to emit its signal, which is

detected by the control signal emission coverage metric `MG-061`. A control that has silently stopped working is indistinguishable from one that is working perfectly, unless its silence is itself an event.

- The claim divergence rate for the action class rises above the threshold that

supported the promotion.

- A supervision viability condition that held at promotion stops holding, which

is what the supervision viability evaluator (`BL-015`) exists to detect.

- The mandate expires and is not renewed. Expiry is the only demotion trigger

that requires nothing to be detected, which is why `GR-L07-04` is disproportionately valuable relative to its cost.

> **DEMOTION MUST NOT DEPEND ON A HUMAN NOTICING**
>
> Every argument in Section 4 applies to the demotion decision itself. If demotion depends on a human observing that a control has degraded, then demotion inherits VC1, VC2 and VC4, and will not happen: the signal that a control has stopped working is exactly the kind of signal that is absent, late, or lost in volume. Demotion must therefore be structural, in the same sense used throughout: the promoted level must cease to be available when its preconditions cease to hold, without any party choosing to withdraw it. Mandate expiry is the cheapest known mechanism with that property, and it is the reason a mandate should be a lease rather than a grant.

---

7. Locating the Structural Autonomy Boundary as a function

The doctoral proposal introduces the Structural Autonomy Boundary as the point at which delegation to an agent ceases to be safe by instruction and comes to require verifiable architectural enforcement. This programme's position is that the boundary is not a line through the space of actions but a function, and that treating it as a line is what produces the failures Section 4 documents.

Three consequences follow, and each is testable.

The boundary moves with the guardrail set, for some actions and not others. For the 28 guardrail-dominated action classes, the boundary is wherever the current condition set puts it, and it moves when the set changes. For the 32 determinant-dominated classes it does not move. An organisation that treats the boundary as fixed will over-invest on the second group and under-invest on the first, which is the observable pattern in the founding case: substantial engineering effort has gone into instructional controls over actions that guardrails could bound, and almost none into identity separation, which would bound nine action classes outright.

The boundary is a property of an environment, not of a technology. The same agent, the same model and the same task sit at different levels in two environments with different guardrail sets. This is why the founding case is evidence of feasibility and an initial source of taxonomy rather than an experimental environment, and why the fifth specific objective requires evaluation in an additional scenario.

The boundary can be crossed without anyone deciding to cross it. A control that degrades, a signal volume that grows, a supervising unit that restarts a terminated agent: none of these is a decision, and each moves the effective boundary while the nominal one stays where it was drawn. This is the strongest practical argument for the supervision viability evaluator (`BL-015`, proposed in D6), and for structural demotion in Section 6.3.

> **THE RELATION RESTATED**

> `L(a, G) = max( floor(a), ceiling(G, a) )`. The Structural Autonomy Boundary is the locus of points where `ceiling(G, a)` crosses Level II, that is, where instruction stops being sufficient and architecture must take over. For actions where `floor(a)` is already Level II or III, the boundary is not the operative constraint and no amount of architecture is either.

---

8. Position against the literature

Novelty claims are stated here against the doctoral proposal's own bibliography, and where the honest answer is that a contribution is an operationalisation or a restatement rather than an extension, it is said.

Parasuraman, Sheridan and Wickens (2000) establish that automation applies in degrees across functional stages and that the appropriate level is selected by evaluating consequences. The claim that autonomy is graded rather than binary is a **restatement** of their position. The extension is that this programme makes the level a function of the guardrail set rather than a design choice evaluated against consequences, so that the same action sits at different levels in two environments. In their terms, the supervision determinants are the cases where no automation-design choice changes the required human role.

Endsley (2017) is the closest prior work to Section 4, and the relationship must be stated carefully because it would be easy to overclaim. Endsley establishes the out-of-the-loop performance problem: as automation increases, the human's situation awareness degrades, so the operator is least able to intervene precisely when intervention is most needed. This programme does not discover that. Its contribution is an **operationalisation**: a per-supervision-point test for whether degradation has occurred in a given architecture, and a measurement of how far. Two elements are genuinely additive. First, the degradation identified here arises from architectural properties, such as an intervention mechanism that cannot halt work in progress, rather than from vigilance decay. Second, it is therefore predictable from the architecture before any operator is observed. Endsley's mechanism is cognitive and this one is structural; they compound rather than compete.

Hobbs and Li (2024) supply the in, on and out-of-the-loop vocabulary. The contribution here is the modest **extension** that the three categories are properties of a supervision point rather than of a system, and that a category assignment is a claim that can fail. The aerospace literature has long distinguished designed-for from as-operated, so this is a narrowing rather than a discovery.

Engin and Hand (2025) argue that fixed supervision categories are inadequate for systems that move between levels, and propose dimensional governance. This programme accepts the argument and **extends** it by asking what determines where on the dimension an action can sit. The answer has two terms, one engineerable and one not, and the second term is the genuine addition: a purely dimensional account has no way to express the claim that some actions do not move along the dimension however much is invested. The access-versus-mandate distinction, which their framing supports directly, is measured here rather than asserted.

Khoo, Foo and Lee (2025) relate capability to risk to governance requirement. The relationship is **complementary**. The action class register can be read as a concrete instantiation of a capability inventory for one environment. The point of difference worth stating is that a capability-risk framework does not distinguish a requirement met by prose from one met by architecture, and the founding case evidence is that the distinction is where the failures live.

NIST AI RMF (2023) articulates principles without specifying verifiable technical controls, which the proposal identifies as the gap. This programme **supplies one of the missing layers** rather than competing: the guardrail catalogue is usable beneath the Manage function and the metric register

beneath Measure. The honest qualification is that the catalogue is derived from one environment and its generality is a hypothesis, which is open question `OQ-004`.

ISO/IEC 27001:2022 and ISO/IEC 42001:2023 locate accountability with a named role and require that human oversight be defined and effective. Nothing here is novel against either standard. The contribution is to assurance practice: the six viability conditions are a testable reading of the word effective, and the founding case demonstrates that a documented oversight arrangement can satisfy an auditor while failing every condition.

Humble and Farley (2010) established the deployment pipeline as an architectural artefact with reversibility, small batch size and separation of duties as first-class properties. Almost everything in guardrail layer L6 is a **restatement** of practice established fifteen years before agentic systems existed, and saying so is important. What is new is the actor: continuous delivery assumes the gated party is a human team that can be held to a process, whereas an agent will satisfy any process that is checkable and route around any that is not. Finding `F-006`, that approval is not bound to the artefact that executes, is a classic continuous delivery defect, and its presence is evidence that agentic environments regress on established practice rather than facing novel problems.

Yao et al. (2023) and Yang et al. (2024) establish that model output is probabilistic and can be confidently wrong, and that the agent-computer interface determines what an agent can do. The governance consequence drawn here is narrow and, as far as this programme can establish, not stated in those terms elsewhere: because the reasoning trace and the action are produced by the same process, the trace is not evidence about the action. Every architecture that treats an agent's explanation of what it did as a record of what it did inherits this.

> **THE SINGLE CLEAREST CLAIM TO NOVELTY**
>
> Of everything in this document, the element with the strongest claim to being new is the separation of `floor(a)` from `ceiling(G, a)`, together with the finding that in a real environment the largest group in the residue is pinned by a determinant that is fully removable by identity engineering. The practical consequence, that an organisation's residue is substantially larger than its ultimate residue and that the difference is mostly identity work, is the result this programme would most want tested elsewhere.

---

9. Limitations and threats to validity

Stated in the order of how much they should worry a reader.

One environment, one snapshot, one observer. Every empirical claim comes from one host in one estate on one day, examined by the party that built it. The guardrail catalogue, the determinant set and the viability conditions are shaped by what that environment happens to contain. A different environment would plausibly produce different layers and different determinants, and there is no way to know from inside how much. This is the reason the founding case is explicitly not the experimental environment.

The observer is the operator. The candidate built and operates the founding case. That produces access and understanding no external observer would have, and it produces a systematic risk that the analysis is shaped by what the builder already believed. Two partial mitigations were applied and neither is sufficient: every factual claim carries an evidence identifier pointing at a raw probe, and one hypothesis carried into the run was corrected rather than confirmed (`F-035`). A discovery that only confirms its own priors is not evidence, and one correction is not many.

The residue is relative to a catalogue. The residue is defined as what remains above Level I under condition set `CS-09`, which contains every catalogued control. A control not in the catalogue cannot move an action out of the residue, by construction. The residue is therefore an upper bound relative to

current knowledge and should shrink as the catalogue grows. It is not a claim about what is possible in principle.

The viability conditions are asserted, not derived. Six conditions were selected because they are the ones the evidence made visible. There is no proof that they are complete, independent or minimal, and the ordering used to select the binding condition is a convention rather than a result. A seventh condition may exist. The conditions are falsifiable in one specific way: if a supervision point passes all six and supervision nevertheless fails, the set is incomplete.

VC4 rests on a volume, not a capacity. The attention assessments rest on the observed ratio of 19,334 automation events to 4 human-facing alerts, which is a volume. Operator capacity was not measured. Where VC4 is recorded as failing, the judgement is defensible but not measured, and that is open question `OQ-005`.

Latency figures are order-of-magnitude. The `t_harm` and `t_detect` values in the supervision register are order-of-magnitude ranges read from schedules and code, not measurements. For most action classes the margin is negative by a factor large enough that precision does not matter. For a minority it is genuinely not computable, and those are marked rather than estimated.

The reported material is not verified. The 61.1 per cent divergence, the deploy incident, the real-time controls and the read-only System of Record identity are all REPORTED. They are load-bearing in this document and they were not verifiable within this run's constraints. Open question `OQ-001` records what a later run should do.

Absence of intervention is not absence of need. Finding `F-042` records that no human intervention appears in 46 days of logs. It does not establish that intervention was warranted and did not occur. It establishes that the loop was not exercised, which is a weaker claim, and the weaker claim is the one this document makes.

The framework has not been used to make a decision. Every framework looks coherent before it is applied. Nothing here has yet caused an action class to be promoted or demoted, and until it has, its usability is untested.

Appendix A. Supervision determinant register

`SD-01` Irreversibility

DEFINITION	An action is pinned to human supervision to the extent that its effect cannot be undone by any mechanism available to the system after the fact.
TEST	Is there a rollback path that restores the prior state exactly, is it available to the actor, and has it been demonstrated by exercise rather than asserted? If any of the three is false, the determinant applies.
APPLIES WHEN	The effect persists after the action, and no exercised rollback restores the prior state; or the rollback itself has a blast radius comparable to the original action.
EXAMPLES	AC-006 rewrite published version control history; AC-024 delete files outside a working tree; AC-027 read credential material at rest; AC-054 delete an audit record
GUARDRAIL-REMOVABLE	partial
RESIDUAL LEVEL	II
MITIGATION	Pre-apply snapshots (GR-L06-02), pre-approved rollback targets only (GR-L06-03) and proven rollback exercise (MG-021) convert some irreversible actions into reversible-with-effort ones. They cannot convert an action whose effect has already left the system boundary, which is why SD-01 and SD-02 compound rather than overlap.
LITERATURE	Humble and Farley 2010 on reversibility as a deployment property; the proposal's own reversibility axis in the autonomy-boundary taxonomy.

SD-02 Externality

DEFINITION	An action is pinned to human supervision to the extent that its effect crosses the system boundary and reaches a party who did not consent to the system's error rate.
TEST	Does any effect of this action become visible to, or act upon, a person or organisation outside the operator's span of control, before any review can occur?
APPLIES WHEN	The output reaches a member of the public, a customer, a regulator, a certificate transparency log, a public name resolution system, or any third party system.
EXAMPLES	AC-047 send an email to an external recipient; AC-048 publish content on a public web surface; AC-049 conduct a real-time voice interaction; AC-050 send a message to a member of the public; AC-040 change a public DNS record
GUARDRAIL-REMOVABLE	no
RESIDUAL LEVEL	III
MITIGATION	Structural interception before the boundary is crossed, of which the reported test-mode overlay (GR-L06-08) is the estate's cleanest instance, moves the boundary rather than removing it: the action becomes internal and the determinant no longer applies to the intercepted form. Nothing reduces the determinant for the real form.
LITERATURE	Extends the blast radius axis of the proposal's taxonomy. Parasuraman, Sheridan and Wickens 2000 treat consequence severity as a determinant of the appropriate automation level; the externality framing narrows this to who bears the consequence.

SD-03 Accountability requirement

DEFINITION	An action is pinned to human supervision where a named human must be answerable for the decision, by law, contract, standard or professional obligation, irrespective of whether the machine would decide correctly.
TEST	If the decision were later challenged, would an answer of the form 'the system decided' be accepted by the party entitled to ask? If not, the determinant applies.
APPLIES WHEN	A statutory, contractual or certification obligation attaches a named signatory to the decision, or the decision constitutes an attestation about the organisation.
EXAMPLES	AC-055 write a completion attestation about its own work; AC-056 record an approval in the deployment approval gate; AC-052 write to the System of Record
GUARDRAIL-REMOVABLE	no
RESIDUAL LEVEL	III
MITIGATION	Nothing reduces this determinant, because it is not a claim about competence. Guardrails can make the human's decision better informed, faster and cheaper, which is worth doing, but the requirement for a human decision is exogenous to the architecture. This is why DM-08 has an autonomy ceiling that no investment raises.
LITERATURE	ISO/IEC 27001:2022 and ISO/IEC 42001:2023 both locate accountability with a named role rather than with a control. NIST AI RMF 2023 GOVERN function, likewise.

SD-04 Value and judgement content

DEFINITION	An action is pinned to human supervision where it trades off values that the organisation has not reduced to a rule, and where more than one resolution is defensible.
TEST	Can the decision be expressed as a predicate over observable state that a reviewer would accept as complete? If the honest answer requires the phrase 'it depends on the circumstances', the determinant applies.
APPLIES WHEN	The correct action depends on a priority ordering that is contested, unstated, or situational, such as availability against correctness, or speed against certainty.
EXAMPLES	AC-044 deploy application code to another estate host during an incident; AC-039 call a cloud control plane mutating API where the resource is shared; AC-045 restart a production service while work is in flight
GUARDRAIL-REMOVABLE	partial
RESIDUAL LEVEL	II
MITIGATION	Making the priority ordering explicit as declarative policy (GR-L07-01) converts part of the judgement into a rule and genuinely removes it from the residue. What remains is the set of cases the policy did not anticipate, which is where default deny (GR-L07-07) becomes the safe resolution rather than an obstruction.
LITERATURE	Engin and Hand 2025 argue that fixed supervision categories fail precisely because the judgement content of an action is not fixed. This determinant operationalises that argument as a per-action test.

SD-05 Unverifiability

DEFINITION	An action is pinned to human supervision where its success cannot be checked by any mechanism independent of the actor that performed it.
TEST	Is there an oracle for success that does not consult the acting agent, and that can be evaluated mechanically? If the only available check is the agent's own report, the determinant applies.
APPLIES WHEN	The claim of completion is not corroborated by infrastructure-emitted evidence, or the evidence store is writable by the actor whose claim it corroborates.
EXAMPLES	AC-055 write a completion attestation about its own work; AC-053 write to its own transcript record; AC-019 dispatch a brief to an agent
GUARDRAIL-REMOVABLE	yes
RESIDUAL LEVEL	II
MITIGATION	This is the determinant most responsive to guardrail investment, and therefore the most important one for the doctorate. An independent verifier under a separate identity (GR-L08-05), infrastructure-emitted events (GR-L08-11) and a prohibition on self-attestation (GR-L08-06) together remove it for a large class of actions. The founding case's 61.1 per cent divergence is a measurement of this determinant going unaddressed.
LITERATURE	Yao et al. 2023 and Yang et al. 2024 establish that model output is probabilistic and can be confidently wrong, requiring external verification rather than reliance on self-report. This determinant is the governance consequence of that finding.

SD-06 Distribution shift

DEFINITION	An action is pinned to human supervision where the situation it is taken in lies outside the distribution over which the system's competence has been evidenced.
TEST	Has this action class been exercised, under conditions materially like the present ones, often enough for its failure rate to be estimated? If the present case is novel in a way the actor cannot itself detect, the determinant applies.
APPLIES WHEN	The environment has changed, the workload is unfamiliar, or the action is being taken for the first time at this scale, in this sequence, or under this load.
EXAMPLES	AC-060 self-update the agent runtime software; AC-039 call a cloud control plane mutating API against a resource class not previously touched; AC-044 deploy application code after an upstream dependency change
GUARDRAIL-REMOVABLE	partial
RESIDUAL LEVEL	II
MITIGATION	Canary and wave ordering (GR-L06-04), blast radius caps (GR-L06-07) and staleness bounds on decision inputs (GR-L11-09) reduce the consequence of being out of distribution. They do not tell the actor that it is out of distribution, which is the part that requires a human, and is why automatic demotion (D2) must be structural rather than dependent on the agent recognising its own novelty.
LITERATURE	Endsley 2017 on the out-of-the-loop performance problem: the operator's situation awareness degrades precisely in the conditions where it is most needed. The same argument applies to the automation itself.

SD-07 Containment shortfall

DEFINITION	An action is pinned to human supervision where the blast radius of a mistake exceeds what the containment architecture can hold, so that a single error escapes the boundary the design assumed.
TEST	If this action were performed wrongly in the worst plausible way, would the consequence stay inside a boundary the operator can restore independently? If not, the determinant applies.
APPLIES WHEN	The actor holds privilege, credentials or reach beyond the scope of the action, so that an error in the action can express itself outside that scope.
EXAMPLES	AC-029 execute a command as root; AC-037 obtain cloud credentials from the instance metadata service; AC-043 execute a command on another estate host; AC-027 read credential material at rest
GUARDRAIL-REMOVABLE	yes
RESIDUAL LEVEL	III
MITIGATION	This determinant is entirely an artefact of architecture and is therefore fully removable in principle. Per-agent workload identity (GR-L01-02), least privilege per action class (GR-L01-03), ephemeral credentials (GR-L01-04), segmentation (GR-L02-08) and egress control (GR-L02-05) remove it. On the founding case it applies to almost everything, because one account holds unrestricted root and five cloud administrator identities.
LITERATURE	The structural guardrails named in the proposal: least privilege, ephemeral credentials, network segmentation. Amazon Web Services 2024 on least privilege as an identity property rather than an operational practice.

SD-08 Objective ambiguity

DEFINITION	An action is pinned to human supervision where the goal it serves is stated in terms that admit more than one faithful reading, so that a competent actor can satisfy the letter of the instruction while defeating its purpose.
TEST	Could a diligent actor complete this action, be judged compliant against the instruction as written, and still have produced an outcome the instructing human would reject? If so, the determinant applies.
APPLIES WHEN	The mandate is prose rather than a predicate, the success criterion is a human judgement, or the instruction contains an implicit precondition the actor cannot check.
EXAMPLES	AC-019 dispatch a brief to an agent; AC-055 write a completion attestation; AC-002 modify source files where the requirement is expressed as an intent
GUARDRAIL-REMOVABLE	partial
RESIDUAL LEVEL	II
MITIGATION	Expressing the mandate as data rather than prose (GR-L07-01), classifying actions at the moment of attempt (GR-L07-08) and defaulting to deny for unclassified actions (GR-L07-07) narrow the space of faithful readings. They do not close it, because the residual ambiguity lives in the goal, not in the mechanism.
LITERATURE	Adjacent to the alignment literature but here treated narrowly as a governance property of the mandate artefact. The proposal's mandate stability metric measures the frequency of re-specification that this determinant predicts.

SD-09 Compounding

DEFINITION	An action is pinned to human supervision where an error in it creates the conditions for further error, so that consequence grows with time rather than remaining bounded at the moment of the mistake.
TEST	Does the output of this action become the input to the next decision of the same actor, without an intervening independent check? If so, the determinant applies.
APPLIES WHEN	Self-repair loops, retry ladders, and any automation whose trigger condition can be produced by its own previous action.
EXAMPLES	AC-016 inject input into another agent's live session; AC-059 answer an interactive prompt on a human's behalf; AC-060 self-update the agent runtime software
GUARDRAIL-REMOVABLE	partial
RESIDUAL LEVEL	II
MITIGATION	Circuit breakers on repeated failure (GR-L09-03), cooling off between related high impact actions (GR-L11-03) and sequencing constraints requiring verification before the next class unlocks (GR-L11-04) bound the compounding without removing it. This determinant is why domain DM-03, self-healing, is where autonomy is most attractive and most dangerous at the same time.
LITERATURE	Not directly addressed in the proposal's bibliography as a supervision determinant. Proposed here as an extension, on the evidence of the founding case's watcher topology, in which four of five independent watchers respond to silence by injecting text into the agent that has gone silent.

SD-10 Adversarial adaptation

DEFINITION	An action is pinned to human supervision where an adversary can observe the control and adapt to it, so that the control's effectiveness degrades as a function of its own deployment.
TEST	Would a party with an interest in the action succeeding be able to observe the control's decision boundary and construct an input that satisfies it? If so, the determinant applies.
APPLIES WHEN	The control is a pattern match, a classifier, or any rule whose decision boundary is inferable from its outputs, and an interested party can supply the input.
EXAMPLES	AC-036 make an arbitrary outbound network request; AC-047 send an email to an external recipient; AC-002 modify source files where the source is drawn from untrusted content
GUARDRAIL-REMOVABLE	partial
RESIDUAL LEVEL	II
MITIGATION	Allowlists (GR-L04-06, GR-L02-06) degrade far more slowly than deny lists (GR-L04-05) under adaptation, because an allowlist's failure mode is a false refusal rather than a false permit. Content-derived instructions reaching an agent's context, which the founding case has as a scheduled path, make this determinant live rather than theoretical.
LITERATURE	Not addressed in the proposal's bibliography. Proposed here as an extension, and flagged as the determinant most likely to be understated in a study whose workloads are benign by construction.

Appendix B. Supervision point register

The full assessment of every supervision point against all six viability conditions. `pass`, `partial` and `fail` are as defined in Section 4.2. `not-computable` appears where a latency term could not be established, and is never treated as a pass.

Table D2-2. Supervision points assessed against the six viability conditions.

ID	NAME	NOMINAL	SIGNAL ORIGIN	INTERVENTION SCOPE	VC1	VC2	VC3	VC4	VC5	VC6	EFFECTIVE	DIVERGES
SP-001	Completion notification from the agent to the operator	on-the-loop	agent	prevents-next	fail	fail	fail	partial	fail	partial	out-of-the-loop	yes
SP-002	Completion marker and report file written by the agent	on-the-loop	agent	none	fail	fail	fail	pass	fail	fail	out-of-the-loop	yes
SP-003	Live terminal view of an agent session in the console	on-the-loop	mixed	prevents-next	partial	fail	fail	fail	partial	partial	out-of-the-loop	yes
SP-004	Console audit log review	on-the-loop	infrastructure	none	pass	fail	fail	pass	pass	fail	out-of-the-loop	yes
SP-005	Deployment approval gate: human approval of a patch	in-the-loop	mixed	prevents-next	partial	pass	fail	pass	fail	pass	in-the-loop, weakly	yes
SP-006	Deployment approval gate: independent reviewer	in-the-loop	agent	prevents-next	fail	pass	fail	pass	fail	fail	out-of-the-loop	yes
SP-007	Deployment approval gate: binding approval to what runs	in-the-loop	infrastructure	none	fail	fail	fail	pass	fail	fail	out-of-the-loop	yes
SP-008	Update plane: human approval of an update proposal	in-the-loop	infrastructure	prevents-next	pass	pass	fail	pass	pass	pass	in-the-loop	no
SP-009	Update plane: kill switch	on-the-loop	infrastructure	prevents-next	pass	not-computable	fail	pass	partial	fail	on-the-loop, nominal only	yes
SP-010	Stall watchdog escalation to the operator	on-the-loop	infrastructure	none	pass	fail	fail	fail	fail	fail	out-of-the-loop	yes
SP-011	Human-facing session watchdog alert	on-the-loop	infrastructure	none	pass	fail	fail	pass	partial	fail	out-of-the-loop	yes
SP-012	Console session termination	on-the-loop	infrastructure	halts-in-progress	pass	fail	pass	pass	pass	partial	on-the-loop for unsupervised sessions only	yes
SP-013	Supervised fleet agent graceful stop	in-the-loop	infrastructure	prevents-next	pass	fail	fail	pass	pass	fail	out-of-the-loop	yes
SP-014	Console chat interrupt	on-the-loop	infrastructure	prevents-next	pass	fail	fail	pass	pass	fail	out-of-the-loop	yes

ID	NAME	NOMINAL	SIGNAL ORIGIN	INTERVENTION SCOPE	VC1	VC2	VC3	VC4	VC5	VC6	EFFECTIVE	DIVERGES
SP-015	Inter-agent message bus kill switch	on-the-loop	infrastructure	prevents-next	pass	not-computable	fail	pass	partial	partial	on-the-loop, nominal only	yes
SP-016	Operator instruction channel from the handset	in-the-loop	mixed	prevents-next	partial	fail	fail	pass	fail	partial	out-of-the-loop	yes
SP-017	Version control review of agent commits	on-the-loop	infrastructure	prevents-next	pass	fail	fail	partial	pass	pass	out-of-the-loop	yes
SP-018	Pre-merge review before code reaches a shared branch	in-the-loop	infrastructure	none	fail	fail	fail	pass	fail	fail	out-of-the-loop	yes
SP-019	Operator oversight of cloud control plane changes	on-the-loop	infrastructure	prevents-next	pass	fail	fail	pass	fail	partial	out-of-the-loop	yes
SP-020	Operator oversight of public name resolution changes	in-the-loop	infrastructure	prevents-next	fail	fail	fail	pass	fail	partial	out-of-the-loop	yes
SP-021	Operator oversight of public entry point configuration	in-the-loop	infrastructure	prevents-next	fail	fail	fail	pass	fail	partial	out-of-the-loop	yes
SP-022	Operator oversight of outbound email	on-the-loop	infrastructure	none	fail	fail	fail	pass	fail	fail	out-of-the-loop	yes
SP-023	Notification rate limit and deduplication as an attention control	on-the-loop	infrastructure	none	pass	not-computable	fail	partial	pass	fail	on-the-loop, partially	yes
SP-024	Structural test mode intercepting external dispatch	out-of-the-loop	infrastructure	halts-in-progress	pass	pass	pass	pass	pass	pass	out-of-the-loop, defensibly	no
SP-025	Acknowledge-to-stop on an escalating external action	on-the-loop	infrastructure	prevents-next	pass	pass	partial	pass	partial	pass	on-the-loop	no
SP-026	Multi-state closure confirmation	in-the-loop	infrastructure	prevents-next	pass	pass	fail	pass	pass	pass	in-the-loop	no
SP-027	Independent audit flow with an adversarial layer	on-the-loop	infrastructure	none	pass	fail	fail	pass	pass	partial	out-of-the-loop at the time of action	yes
SP-028	Compliance attestation by a human owner	in-the-loop	mixed	prevents-next	partial	pass	pass	pass	partial	pass	in-the-loop	no
SP-029	Read-only connection identity to the System of Record	out-of-the-loop	infrastructure	halts-in-progress	pass	pass	pass	pass	pass	pass	out-of-the-loop, defensibly	no

ID	NAME	NOMINAL	SIGNAL ORIGIN	INTERVENTION SCOPE	VC1	VC2	VC3	VC4	VC5	VC6	EFFECTIVE	DIVERGES
SP-030	Interactive permission prompt raised by the agent tooling	in-the-loop	infrastructure	halts-in-progress	pass	pass	pass	fail	partial	fail	out-of-the-loop	yes
SP-031	Operator oversight of changes to agent instruction files	on-the-loop	infrastructure	prevents-next	fail	fail	fail	pass	fail	partial	out-of-the-loop	yes
SP-032	Operator oversight of fleet registry changes	on-the-loop	infrastructure	prevents-next	pass	fail	fail	pass	partial	pass	out-of-the-loop	yes
SP-033	Resource consumption oversight	on-the-loop	mixed	none	partial	not-computable	fail	pass	partial	fail	out-of-the-loop	yes
SP-034	Restart supervision of agent sessions	out-of-the-loop	infrastructure	prevents-next	pass	not-computable	fail	pass	pass	fail	out-of-the-loop	no

The binding condition and the supporting detail for each point:

Table D2-3. Binding condition, latency terms and the durable finding recording each divergence.

ID	BINDING CONDITION	SIGNAL LATENCY	HARM COMPLETION	FINDING	EVIDENCE
SP-001	VC1 independence	seconds after the agent chooses to send; unbounded if it does not	already complete: the notification is sent after the work	F-001	EV-029 EV-039
SP-002	VC1 independence	unbounded: the operator must go and look	already complete	F-001	EV-028 EV-039
SP-003	VC4 attention	sub-second while a human is watching; unbounded otherwise	milliseconds to seconds for most local actions	F-002	EV-014 EV-020 EV-039
SP-004	VC2 latency	unbounded: the operator must go and look	already complete	F-003	EV-034 EV-039
SP-005	VC5 comprehensibility	minutes to hours: the patch waits	seconds once approved and applied	F-004	EV-030
SP-006	VC1 independence	minutes	seconds once approved and applied	F-005	EV-030
SP-007	VC1 independence	unbounded: there is no signal	seconds: the process manager re-executes whatever is on the shared tree	F-006	EV-030
SP-008	none: all six conditions hold	unbounded: proposals wait indefinitely	not applicable today: no proposal has ever been applied	F-007	EV-031
SP-009	VC6 authority	unbounded: no signal is emitted	seconds	F-008	EV-031
SP-010	VC2 latency	180 seconds stall plus up to 60 seconds schedule granularity plus two nudge cycles: realistically over ten minutes	milliseconds to seconds for the actions the agent is taking meanwhile	F-009	EV-028 EV-039
SP-011	VC2 latency	300 seconds stall plus up to 120 seconds schedule granularity	milliseconds to seconds	F-009	EV-028 EV-039
SP-012	VC2 latency	n/a: this is an intervention rather than a signal	milliseconds to seconds	F-010	EV-013 EV-039
SP-013	VC3 halt	n/a: this is an intervention rather than a signal	milliseconds to seconds	F-011	EV-039
SP-014	VC3 halt	n/a: this is an intervention rather than a signal	milliseconds to seconds	F-011	EV-025 EV-039
SP-015	VC3 halt	unbounded: no signal is emitted	milliseconds	F-012	EV-025 EV-039
SP-016	VC5 comprehensibility	seconds	seconds: the orchestrator's tools execute immediately	F-013	EV-038 EV-039
SP-017	VC2 latency	unbounded: the operator must go and look	seconds to push; the artefact is then visible to every consumer	F-014	EV-010
SP-018	the control is absent	n/a: the control is absent	seconds	F-015	EV-010 EV-030

ID	BINDING CONDITION	SIGNAL LATENCY	HARM COMPLETION	FINDING	EVIDENCE
SP-019	VC2 latency	unbounded: no signal reaches the operator	seconds; some resource deletions are irreversible in effect	F-016	EV-033
SP-020	VC1 independence	unbounded: no signal is emitted	seconds to change; bounded below by the record time to live to correct	F-016	EV-033
SP-021	VC1 independence	unbounded: no signal is emitted	seconds; exposure begins at the moment of reload	F-016	EV-008
SP-022	VC1 independence	unbounded	seconds; delivery is not recallable	F-017	EV-029
SP-023	VC4 attention	sub-second when not suppressed; suppressed messages are dropped rather than queued	n/a: this control governs the signal, not an action	F-018	EV-029 EV-039
SP-024	none: the action is contained before the boundary is crossed	immediate at the point of interception	n/a: the harm does not occur, because the boundary is not crossed	F-019	REPORTED
SP-025	none for the ladder; the individual step remains uninterruptible	bounded by the ladder's step interval	each step completes on dispatch; the ladder as a whole is what is being stopped	F-020	REPORTED
SP-026	none	bounded by the state machine	n/a: the control governs closure, not dispatch	F-020	REPORTED
SP-027	VC2 latency	days to weeks: the audit is retrospective	already complete: the audit examines finished episodes	F-021	REPORTED
SP-028	none: the ceiling here is an accountability requirement, not a control weakness	days: the attestation cycle is deliberate	n/a: nothing takes effect until the human attests	F-022	REPORTED
SP-029	none: the control does not depend on agent compliance	n/a	n/a: the write cannot be expressed	F-023	REPORTED
SP-030	VC6 authority: the prompt is suppressed by a permission-skipping flag, and where it does appear a watchdog answers it automatically	immediate when raised	the action is blocked while the prompt is unanswered	F-024	EV-020 EV-028 EV-032
SP-031	VC1 independence	unbounded	takes effect at the next session start	F-025	EV-032
SP-032	VC2 latency	seconds when the operator is looking; unbounded otherwise	immediate	F-026	EV-022
SP-033	VC6 authority	hours: synchronisation is twice hourly	n/a: consumption is continuous rather than an event	F-027	EV-024
SP-034	VC6 authority: termination by the operator is undone automatically, so the operator's override does not survive; the nominal posture is already out-of-the-loop, so there is no divergence, but the arrangement defeats SP-012	n/a	seconds: a terminated session is replaced within the restart interval	F-010	EV-005 EV-028

Appendix C. Irreducible residue register

Thirty-two action classes remain above Level I under condition set `CS-09`, the fully hardened configuration. Each is pinned to its dominant determinant.

Table D2-4. The irreducible supervision residue.

ACTION CLASS	NAME	RESIDUE LEVEL	DOMINANT DETERMINANT	PINNED BY	BEST ACHIEVABLE	GUARDRAILS REQUIRED
AC-006	Rewrite published version control history	II	SD-01	SD-01 SD-02	II	GR-L06-02 GR-L08-04 GR-L04-06
AC-019	Dispatch a brief to an agent	II	SD-08	SD-08 SD-05 SD-09	II	GR-L07-01 GR-L07-03 GR-L07-08
AC-016	Inject input into another agent's live session	III	SD-09	SD-09 SD-01 SD-08	II	GR-L09-03 GR-L11-03 GR-L11-04 GR-L07-08
AC-024	Delete files outside a working tree	II	SD-01	SD-01 SD-07	II	GR-L05-09 GR-L03-06 GR-L01-03
AC-027	Read credential material at rest	III	SD-01	SD-01 SD-07	II	GR-L01-04 GR-L01-05 GR-L01-02
AC-029	Execute a command as root	III	SD-07	SD-07 SD-04	II	GR-L01-03 GR-L01-06 GR-L04-03
AC-030	Modify accounts, groups or privilege configuration	III	SD-07	SD-07 SD-01	III	GR-L01-02 GR-L07-06 GR-L08-11
AC-032	Create or modify a systemd unit	III	SD-07	SD-07 SD-09	II	GR-L07-06 GR-L03-04 GR-L08-11
AC-033	Modify reverse proxy configuration	III	SD-02	SD-02 SD-07	III	GR-L06-01 GR-L06-12 GR-L10-04
AC-034	Reload or restart the reverse proxy	II	SD-02	SD-02	II	GR-L06-06 GR-L06-12
AC-036	Make an arbitrary outbound network request	II	SD-02	SD-02 SD-01 SD-10	II	GR-L02-05 GR-L02-06 GR-L02-11
AC-037	Obtain cloud credentials from the instance metadata service	III	SD-07	SD-07 SD-01	II	GR-L02-12 GR-L01-04 GR-L01-11
AC-039	Call a cloud control plane mutating API	III	SD-07	SD-07 SD-04 SD-06	II	GR-L01-03 GR-L07-01 GR-L06-07
AC-040	Change a public DNS record	III	SD-02	SD-02 SD-01	III	GR-L06-01 GR-L11-08 GR-L10-04
AC-041	Obtain or renew a public TLS certificate	II	SD-02	SD-02	II	GR-L06-01 GR-L08-11
AC-043	Execute a command on another estate host	III	SD-07	SD-07 SD-06	II	GR-L01-03 GR-L02-08 GR-L04-06
AC-044	Deploy application code to another estate host	III	SD-03	SD-03 SD-04 SD-06 SD-01	II	GR-L06-01 GR-L06-02 GR-L06-04 GR-L06-12 GR-L08-05
AC-045	Restart a production service on another estate host	II	SD-04	SD-04 SD-01	II	GR-L11-08 GR-L06-04 GR-L09-08
AC-047	Send an email to an external recipient	III	SD-02	SD-02 SD-01	II	GR-L06-08 GR-L10-04 GR-L11-08
AC-048	Publish content on a public web surface	III	SD-02	SD-02 SD-01	III	GR-L06-01 GR-L06-08 GR-L10-04
AC-049	Conduct a real-time voice interaction with a member of the public	III	SD-02	SD-02 SD-03 SD-04 SD-01	III	GR-L06-08 GR-L09-07 GR-L09-08 GR-L11-08
AC-050	Send a message to a member of the public	III	SD-02	SD-02 SD-03 SD-01	III	GR-L06-08 GR-L10-04 GR-L11-08

ACTION CLASS	NAME	RESIDUE LEVEL	DOMINANT DETERMINANT	PINNED BY	BEST ACHIEVABLE	GUARDRAILS REQUIRED
AC-052	Write to the System of Record	III	SD-03	SD-03 SD-01 SD-02	III	GR-L05-01 GR-L05-03 GR-L05-04
AC-054	Delete or truncate an audit or transcript record	III	SD-01	SD-01 SD-05	III	GR-L08-03 GR-L08-04 GR-L05-06 GR-L01-07
AC-055	Write a completion attestation about its own work	II	SD-05	SD-05 SD-03	II	GR-L08-05 GR-L08-06 GR-L08-08 GR-L08-11
AC-056	Record an approval in the deployment approval gate	III	SD-03	SD-03 SD-05	III	GR-L01-08 GR-L08-06 GR-L07-03
AC-057	Set or clear the update kill switch	III	SD-07	SD-07 SD-03	III	GR-L01-08 GR-L07-06 GR-L09-11
AC-059	Answer an interactive permission or resume prompt on a human's behalf	III	SD-03	SD-03 SD-09 SD-05	III	GR-L10-04 GR-L07-06 GR-L08-11
AC-060	Self-update the agent runtime software	II	SD-06	SD-06 SD-09 SD-10	II	GR-L06-04 GR-L06-11 GR-L11-04
AC-025	Modify its own instruction files	II	SD-08	SD-08 SD-09	II	GR-L03-07 GR-L08-11 GR-L07-05
AC-026	Modify the permission settings file	III	SD-07	SD-07 SD-03	III	GR-L03-08 GR-L07-06 GR-L01-02
AC-021	Modify an agent's startup wrapper or launch script	III	SD-07	SD-07 SD-08	III	GR-L03-07 GR-L07-06 GR-L07-05

The argument for each entry, stating why guardrail investment does not move it:

Table D2-5. Why guardrails do not move each residue entry.

ACTION CLASS	WHY GUARDRAILS DO NOT MOVE IT
AC-006	The published state has already been consumed by other parties and by automated processes; a later correction does not undo what was fetched in the interim.
AC-019	The brief is the mandate. Its ambiguity is a property of natural language goals, not of the delivery mechanism, so no delivery-side control resolves it.
AC-016	Injected text becomes part of another agent's reasoning context and cannot be withdrawn; the receiving agent may act on it before any control observes the injection, and its action can trigger further injection.
AC-024	Deletion of material not held in version control has no restore path unless a backup exists, and a backup the same identity can delete is not a restore path.
AC-027	A secret cannot be un-read. The only remedy is rotation, which is a different action with its own blast radius, and the exposure window is not bounded by the response.
AC-029	Root authority is not an action class, it is the absence of a boundary, so no control operating below the operating system bounds it. Removing it is an identity change, not a guardrail.
AC-030	The action can grant a second identity the same authority, so revoking the first identity does not revoke the capability, and the containment shortfall persists after the correction.
AC-032	The unit set includes the units that supervise the agents, so this action can change the terms of its own supervision, which places it outside any control the supervision layer provides.
AC-033	A configuration change publishes an internal service to the public network; retrieval by third parties during the exposure window is not undone by reverting.
AC-034	The moment of reload is the moment exposure begins; the control must act before it, and after it there is nothing to control.
AC-036	Data that has left the host cannot be recalled. Egress control reduces the set of destinations but the action remains irreversible for permitted destinations.
AC-037	Once minted, the credential is held by the process; blocking the endpoint afterwards does not revoke what was already issued.
AC-039	The identity's authority spans resources unrelated to the action, so a mistake in one action expresses itself across the whole account. Bounding the action does not bound the identity.
AC-040	Propagation and caching mean the window of wrongness is bounded below by the record's time to live rather than by the speed of the correction.
AC-041	Issuance is recorded permanently and publicly in certificate transparency; the act is externally visible and not retractable.
AC-043	The reach of the command is the reach of the credential on the target, not the scope of the intent, and the target's own containment is outside this host's control.
AC-044	The founding case's canonical incident is here: a deploy reported as successful had disabled the service's core connection function. Verification removes SD-05, but the decision to deploy under uncertainty remains a judgement with an accountable owner.
AC-045	Whether the interruption is acceptable depends on what is in flight, which is a situational judgement the actor cannot fully observe. In real-time domains the in-flight work is a person on a call.
AC-047	Delivery is immediate and not recallable, and the recipient is outside the operator's span of control.
AC-048	Removal does not undo retrieval, indexing or caching that occurred while the content was live.
AC-049	The action completes in seconds and its output reaches a person immediately. There is no useful after: supervision must be designed in before the action or it does not exist. Interception before the boundary, as in GR-L06-08, changes which action is being taken rather than supervising this one.
AC-050	The founding case has a recorded episode in which messages about fabricated events reached a real external number. The message is complete on dispatch and the recipient acts on it before any review.
AC-052	The record is authoritative for parties outside the system, and an incorrect authoritative record is acted upon by people. Accountability for the content sits with a named human by obligation, not by preference.
AC-054	Destroying evidence destroys the basis on which every other control is evaluated, including the evaluation of whether this action occurred.

ACTION CLASS	WHY GUARDRAILS DO NOT MOVE IT
AC-055	Self-attestation is unverifiable by construction. Independent verification removes SD-05 and moves this to Level II with a real review, but the attestation itself, as an assertion an organisation relies on, retains an accountability component.
AC-056	An approval is a statement that a named party accepts responsibility. A machine recording it does not create the acceptance, whatever the data model says.
AC-057	A safety flag whose state the governed party controls is not a safety flag. The action must be outside the governed party's reach, which is an identity property.
AC-059	The prompt exists to obtain a human decision. Answering it automatically does not supervise the action, it removes the supervision point, and no control downstream can restore a decision that was never made.
AC-060	Each update moves the actor outside the distribution over which its competence was evidenced, and the actor cannot detect that it has done so.
AC-025	Changing the instruction changes the objective the next actor pursues. Because the instruction layer is context and not enforcement, the security consequence is bounded, but the objective drift is not observable from the action itself.
AC-026	An actor that can widen its own permissions has no boundary; the boundary must be held by an identity the actor does not hold, which is an architectural property rather than a control.
AC-021	The wrapper determines the flags and the standing instruction of every future session of that agent, so writing it is mandate mutation performed by the governed party.