The Vergara Agentic Autonomy Scale (V0–V6)

A risk-gated, operational standard for agents
acting on business systems

Fernando Vergara
Hubents Tech S.L. — Koble
ORCID: 0009-0000-7355-7107
hello@fernandovergara.com

Canonical definition: fernandovergara.com/escala-vergara
Reference kit (CC BY 4.0): github.com/NandoVer/vergara-scale

Cite as: Vergara, F. (2026). *The Vergara Agentic Autonomy Scale (V0–V6).*
DOI 10.5281/zenodo.21893261 · Licensed CC BY 4.0

August 2026

Abstract

Vendors describe AI agents as "autonomous" without a shared unit of measurement. Existing frameworks classify autonomy but stop short of making it *operational*: a dial the data controller can move, enforced in code, with guarantees attached to each position. This paper proposes the **Vergara Agentic Autonomy Scale** (VAAS), a seven-level scale (V0–V6). Six levels are obtained by composing two orthogonal axes — *initiative* (who starts the loop) and *execution authority* (what the agent may commit without a human); the seventh moves a third axis, the *authority domain*, where an agent acts across organisations under explicit composition rules. The core is a *risk-gated* execution model in which the level is never sufficient on its own: the decision to execute, propose, or block is a function of the level *and* the risk class of the individual action, subject to an invariant ceiling that no level overrides. We state seven properties (P1–P7) that an operational autonomy scale must satisfy to be more than a marketing label, and we define an *Autonomy Statement*: a ten-field disclosure that turns the claim "we are V3" into something a customer or auditor can check. The scale is not a proposal on paper: it is derived from, and validated against, two multi-tenant SaaS products in production (Hubents, event management; Koble, agency operations), which classify 24 and 12 write actions respectively, share identical thresholds (autonomous execution from V3; 7-day undo window), and enforce level gates in code rather than in prompts. Three findings are reported: (i) the two implementations agree in their runtime gates but *disagreed in the ladder each published to its customers*, so one product's V1 advertised authority its own gates withheld — a live failure of controller legibility while enforcement itself held, which is the case that proves those two properties must be stated separately, and which yields a general ordering principle (minimum marginal authority); (ii) they *agreed* on the risk class of all seven actions common to both domains, suggesting risk class is a property of the action type rather than of the vertical; and (iii) adoption data from 160 production organisations show that granting a level is not the same as exercising it: 157 sit at the fail-safe default, one tenant has been raised to V4 yet has run zero missions, and the autonomy actually exercised anywhere is V3's — 14 low-risk actions committed unaided over two weeks, none of them reversed — while 61 % of all proposals were dismissed. We are explicit about what this does and does not support: the scale is validated as a design and disclosure instrument, and the outcome data are two weeks old on a single tenant.

1 Introduction

The word "autonomous" currently carries no measurable content in enterprise software. A product that drafts an email and a product that files a tax document without review are both marketed as agentic. The buyer has no unit in which to ask "how autonomous?", the operator has no dial to set, and the auditor has no artefact to inspect.

The automation literature has offered ordinal scales since Sheridan and Verplank's ten levels of supervisory control [1], refined by Endsley and Kaber [2] and by Parasuraman, Sheridan and Wickens [3]. One scale escaped academia and reached the market: SAE J3016 [4], whose six driving-automation levels are now quoted in regulation, insurance and advertising. Its success is instructive. J3016 did not win because it ranked intelligence; it won because each level answers a liability question — who is responsible for what, and when must the human be able to take over.

Recent work has begun to port this idea to AI agents. Feng, McDonald and Zhang [5] define five levels by the role the human occupies (operator, collaborator, consultant, approver, observer) and propose autonomy certificates. The Cloud Security Alliance [6] publishes a six-level control framework (L0–L5) across five dimensions — decision authority, scope, reversibility, impact, temporal duration — with prescribed controls per level. Both are advances, and this paper does not claim priority over them. Both, however, remain *descriptive*: they classify a deployment after the fact. Neither specifies the mechanism by which a level becomes a runtime constraint, who is entitled to change it, what must be true of an action before a given level may commit it, or what the customer is owed when it does.

That gap is where this paper sits. We report a scale that was built *inside* two shipping products because the products needed a dial, and then generalised. Our contributions:

C1 — An axis-based derivation.
V0–V6 are not an arbitrary ladder. V0–V5 are the coherent compositions of *initiative* and *execution authority* (§3), which explains why certain intuitive orderings are wrong; V6 moves a third axis — the *authority domain* — which is why acting across organisations is a level of its own and not a variant of multi-agent work (§3.3).

C2 — Risk-gated execution.
The level alone never authorises an action. Authority is the conjunction of level and the action's risk class, under an invariant ceiling that holds at every level, including the highest (§4).

C3 — Seven conformance properties.
P1–P7 (§3.4) state what makes a scale operational: monotonicity, invariant ceiling, reversibility, fail-safe defaults, immediate descent, controller legibility, and enforcement in code rather than prompt.

C4 — The Autonomy Statement.
A ten-field disclosure (§7) that makes a level claim falsifiable.

C5 — Two reference implementations in production.
Their action catalogues, thresholds, guardrails and tenant-level distribution are reported in full (§5).

C6 — Three empirical findings.
Drawn from those implementations (§6), including a disagreement that generalises into an ordering principle and a distinction between granted and exercised autonomy.

On the name. The scale is named after its author for lack of an existing standard to extend, following the convention by which ordinal instruments carry the name of whoever first specified

them. The genealogy above is acknowledged deliberately: the contribution claimed here is the operational and risk-gated construction, not the idea that autonomy admits levels.

What the scale is not. It does not measure model capability, answer quality, or breadth of data access — the last is authorisation (RBAC/ABAC) and is orthogonal. A V3 agent restricted to one project and a V3 agent with organisation-wide read access differ in scope, not in autonomy. Nor does the scale rank domain risk: V3 in event catering and V3 in clinical records demand different action catalogues, which is precisely why the catalogue, not the level, carries the risk semantics.

2 Related work and the gap

Levels of automation (1978–2000). Sheridan and Verplank [1] scaled human–computer control of undersea teleoperators from fully manual to fully autonomous, with intermediate levels where the computer narrows options or suggests. Endsley and Kaber [2] distributed the functions of monitoring, generating, selecting and implementing across ten levels. Parasuraman et al. [3] separated *stages* of automation (information acquisition, analysis, decision selection, action implementation) from *levels* within each stage, and argued for evaluating automation by its human-performance consequences. This tradition supplies our vocabulary and one essential idea that survives into the Vergara Scale: authority over the *action* stage is categorically different from authority over the *information* stages. It was designed, however, for a single operator supervising one machine in real time — not for a software agent acting on a multi-tenant database on a cron schedule, where the human is absent by design and reversibility is a database property rather than a physical one.

SAE J3016 as the template for adoption. J3016 [4] defines six levels by allocating the dynamic driving task and its fallback between human and system. Two features explain its market penetration and are imitated here: level 0 is a real level (not the absence of a product), and each level is stated as an obligation, not a capability.

Autonomy levels for AI agents (2025–2026). Feng et al. [5] treat autonomy as a design choice distinct from capability and environment, and characterise five levels by the user's role, proposing third-party autonomy certificates. Their framing is human-relational: it tells us where the human stands, and it is the closest prior work to our P6 (controller legibility). It does not bind levels to a catalogue of permissible actions. The CSA framework [6] is closer to practice: six levels, five classification dimensions, and prescribed controls that escalate from per-action approval to kill switches and anomaly detection, under the principle that autonomy must be granted, enforced and proportionally controlled rather than assumed. It is the most direct antecedent of this work. It differs in four respects that motivate the present paper:

1. *Audience.* CSA addresses the security and governance function of an enterprise deploying agents. The Vergara Scale addresses the vendor's product surface and the customer who operates the dial — the level is something the tenant sets in the application, not something a CISO documents about it.

2. *Reversibility.* CSA treats reversibility as a dimension used to *classify* a deployment. We treat it as an *enabling condition*: V3 does not exist unless every autonomously executed action carries an undo affordance and an audit row, within a stated window.

3. *The ceiling.* A level-indexed control matrix implies that a sufficiently high level unlocks everything below it. We assert the opposite for a defined set of actions: money, fiscally

or legally effective documents, and any communication leaving the organisation require a human at *every* level, V5 included.

4. *Evidence.* Neither prior framework, to our knowledge, publishes a reference implementation with its action catalogue, thresholds and observed level distribution. §5–6 do.

Governance instruments. The NIST AI Risk Management Framework [7] organises practice into Govern, Map, Measure and Manage without prescribing autonomy levels. Regulation (EU) 2024/1689 (the AI Act) [9] requires, for high-risk systems, that they be designed so they *can* be effectively overseen by humans (Art. 14), and places operational duties on deployers (Art. 26). Both are compatible with, and under-specified by, an autonomy scale: they demand oversight without defining the granularity at which it is exercised. §8 maps the scale onto them.

Summary of the gap. No prior framework is simultaneously (a) declarative enough for a market to quote, (b) executable as a runtime constraint, (c) accompanied by per-level guarantees to the customer, and (d) operable by the data controller rather than by the vendor.

3 The Vergara Agentic Autonomy Scale

3.1 Two orthogonal axes

Autonomy in business systems conflates two questions that vary independently.

Axis I — Initiative: who starts the loop?

- I_0: only the human, per interaction.
- I_1: the agent starts *communication* on a schedule or an event (a briefing, a pre-deadline check-in).
- I_2: the agent starts *candidate actions* (proposals) that a human dispositions.
- I_3: the agent pursues an *objective* across sessions, deciding its own next steps between human contacts.
- I_4: the agent delegates to other agents and integrates their work.

Axis A — Execution authority: what may it commit unaided?

- A_0: no writes at all.
- A_1: writes only under a direct human instruction, with confirmation in the same exchange.
- A_2: writes of *low-risk* actions without a human in the loop, audited and reversible.
- A_3: as A_2, plus the authority to sequence such writes towards a goal without per-step contact.

A third axis stays fixed throughout V0–V5 and is named here only because V6 moves it: the *authority domain* — whether the agent acts inside one organisation, with one Statement and one ceiling, or across two (§3.3).

The two axes are orthogonal in principle but not all sixteen combinations are coherent products. An agent that executes unaided (A_2) but never reports (I_0) is unusable: the operator cannot supervise what is never surfaced. Conversely, initiative without authority (I_2, A_1) is the most common useful configuration in practice, and it is a genuine level rather than a half-measure. The scale is the ordered set of coherent compositions, and it is ordered by *marginal authority granted*, not by apparent sophistication.

3.2 The seven levels

Level	Name	I	A	Marginal grant over the previous level
V0	Responder	I_0	A_1	Nothing. Answers on request; writes only on direct order, confirmed in the exchange.
V1	Briefer	I_1	A_1	The right to *address the operator* unprompted: scheduled and event-anchored information.
V2	Proposer	I_2	A_1	The right to *prepare* work: concrete actions queued for per-item human disposition.
V3	Bounded executor	I_2	A_2	The right to *commit* low-risk actions unaided, audited and reversible within the undo window.
V4	Mission manager	I_3	A_3	The right to *sequence* such commits towards an assigned objective, under run caps and blocking escalation.
V5	Orchestrator	I_4	A_3	The right to *delegate* to other agents under a coordinator, inheriting every ceiling above.
V6	Federated	I_4	A_3	Not more of the same: the right to act *across* organisations, under the composition rules of §3.3. Adds a third axis, not a rung.

Table 1: The Vergara Agentic Autonomy Scale. Spanish level names as deployed: V0 Consulta, V1 Asistente, V2 Copiloto, V3 Autopiloto, V4 Gestor, V5 Multiagente, V6 Federado. V5 and V6 are specified and reserved (§3.2).

Each level is cumulative (P1): V3 includes the briefing of V1 and the proposal queue of V2, because an executor that stops reporting is a regression in oversight even as it advances in authority.

The two decisive transitions. V2→V3 is the only transition that changes who commits a write; every earlier step changes only who speaks first. V3→V4 is the only transition that removes the per-action human decision *point* rather than the per-action human *decision*: at V4 the human sets an objective and the agent decides the sequence. Products should price and gate those two transitions differently from the others, and operators should audit them differently.

Reserved status of V5 and V6. In both implementations described in §5, V5 is declared but behaves as V4 (level gates use $\geq$), and V6 is not implemented at all. We report both as *reserved*: their semantics are specified in the scale but not yet exercised in production, and claiming conformance to either today would violate P7. We specify them anyway, for the reason SAE J3016 specified level 5 years before any vehicle could claim it: a scale is normative, and the level that does not yet exist is the one that tells implementers what they are building towards. Two of seven levels being reserved is a statement about the state of the art, not a gap in the instrument.

3.3 V6: the inter-organisational case

V0–V5 all share an assumption so basic it is easy to miss: there is one domain of authority. One data controller sets the level, one Autonomy Statement defines the catalogue, one ceiling holds, one undo window applies, one audit trail records. Every rung so far moves along initiative or execution authority *inside* that domain.

An agent that coordinates with the agent of a *different* organisation is not a further step along either axis. It introduces a third one — call it the *authority domain* — and that is why it cannot be folded into V5. V5 delegates to sub-agents that inherit its ceiling; V6 negotiates with an agent whose ceiling it does not control. Three properties break at once:

- *Authority is no longer set by one Statement.* Two parties bring two catalogues, two ceilings and two thresholds.
- *Undo stops being local* (P3). Reversing what your agent committed inside the counterparty's system requires their cooperation, so reversibility becomes a negotiated property rather than a database one.
- *The audit trail splits* (P3). Two records exist, and neither is complete on its own.

Why this is possible at all. In the risk criteria of §4, any communication leaving the organisation is high risk *because the far side is opaque*: an unbounded recipient, no knowable constraints, no reversal. Between two agents that each publish a machine-readable Autonomy Statement, that opacity is precisely what disappears. The far side's catalogue, ceiling and undo window are readable before acting. This is the point at which the Autonomy Statement stops being a disclosure document and becomes a *protocol*: without one, published and machine-readable, V6 is not available at all.

Composition rules. A V6 interaction is governed by the two Statements in play, resolved conservatively:

1. **Effective authority is the intersection.** An action is autonomously executable only if both Statements classify it as low risk and both parties are at $\ell \geq \theta$.
2. **The effective ceiling is the union.** If an action sits in either party's invariant ceiling, it requires a human — whichever side proposes it.
3. **The effective undo window is the shorter of the two**, and an action reversible on only one side is treated as irreversible.
4. **Audit is bilateral.** Each party records what its own agent committed, with a shared correlation identifier so the two records can be reconciled after the fact.
5. **Descent is unilateral and immediate** (P5). Either party lowering its level ends the federated interaction at once, with no negotiation and no drain period.

Under those rules V6 grants no authority that neither party had alone: it grants the right to exercise the intersection *without a human relaying between the two*. That is the whole increment, and it is worth a level because relaying is what makes inter-organisational work slow — and because getting the composition wrong is how one organisation's autonomy becomes another's incident.

A concrete case. In an event-management deployment, a planner and a catering provider already share a scoped collaboration. At V6 the planner's agent and the provider's agent settle headcount changes between themselves within what both Statements permit — reading updated guest counts, adjusting internal schedules — while anything that touches money, a fiscal document or a message to the end client stays behind a human on both sides, because rule 2 makes either party's ceiling binding on both.

3.4 Seven properties of an operational scale

A scale that only labels deployments cannot be enforced or audited. We hold that the following are necessary; a system satisfying all seven for level n, with a verifiable Autonomy Statement (§7), is *Vn-conformant*.

P1 Monotonicity. Capabilities of Vn $\subset$ V(n+1). No level trades away an oversight affordance for authority.

P2 Invariant ceiling. A non-empty set H of action classes requires human confirmation at *every* level. H is a business decision, stated publicly, and not derivable from the level.

P3 Reversibility and audit. Every action committed without a human produces an audit record and, where the action is reversible in principle, an undo affordance valid for a stated window W. Absent this, V3 is not available.

P4 Fail-safe defaults. The initial level is V0. An action absent from the risk catalogue is treated as maximum risk. Failure of an autonomous execution degrades to a pending proposal, never to a silent drop.

P5 Immediate descent. Lowering the level takes effect at once and requires no queue draining, cancellation flow or vendor intervention. Descent is the kill switch and must never be harder than ascent.

P6 Controller legibility. The level is stated in the language of the business, not of the model, and is set by the data controller (the customer), not by the vendor. Levels the vendor reserves must be visibly marked as such.

P7 Enforcement in code. Level gates and risk classes live in program logic under regression test, not in system prompts. A prompt is a request; a gate is a constraint.

P7 deserves emphasis because it is the property most often violated in current products. An agent instructed "never send invoices without confirmation" holds that boundary at the pleasure of the model. The same boundary expressed as a branch in the execution path holds regardless of what the model decides, what a user writes into a prompt, or what an injected document requests.

3.5 What must match across implementations, and what must not

A scale meant to be adopted by more than one product has to say where conformance ends and local judgement begins. Two implementations of the Vergara Scale are comparable when they share the *semantics* of the levels; they are expected to differ in the *parameters*.

Fixed — identical in any conformant system	**Free — set per product, and disclosed**
What each level means, and their order (Table 1)	The threshold θ at which low risk becomes autonomous
The three risk classes and the four criteria that assign them	The undo window W
P1–P7	The membership of the invariant ceiling H (non-empty)
That the decision is level $\times$ risk class, never level alone	The action catalogue and its size
That the published ladder states what the gates enforce	The self-service ceiling, and which levels the vendor reserves

Table 2: Conformance is semantic, not parametric.

The distinction is not a convenience. If the parameters were fixed too, the scale would be a product specification and could not travel: an agency CRM and a clinical records system need different catalogues and different ceilings, and forcing one on both would guarantee that at least one of them is wrong. If the semantics were free, "V3" would carry no information across vendors and the Autonomy Statement would have nothing to compare. The two reference implementations of §5 differ in every free parameter — 24 versus 12 actions, a ceiling of one

action versus four, self-service to V4 versus V3 — and agree in every fixed one. That is the intended shape of two conformant systems, and it is why §6 treats the label divergence as a defect while treating the ceiling divergence as ordinary variation.

4 Risk-gated execution

4.1 Risk classes

The unit of risk is the *action type*, not the request. Each write action the agent can perform is assigned one of three classes by four criteria: technical reversibility, externality (does the effect leave the organisation?), legal or monetary effect, and third-party visibility.

Low. Internal, fully reversible, no accounting or legal effect. Creating an internal task, posting to an internal thread, scheduling an internal reminder, storing a learned note.

Medium. Internal but consequential to shared state: duplicate or noise risk, downstream reports, other people's work. Reversible with effort. Creating a contact or an event, setting an attendance response, changing a vendor's status, drafting an email that stays a draft.

High. Money, fiscally or legally effective documents, and *any* communication leaving the organisation. Irreversible in the sense that matters: the recipient has already read it.

The externality criterion is what makes the classification transferable across domains. "Reply to the lead" is high not because replying is difficult but because it cannot be recalled and it speaks in the organisation's name.

4.2 The decision function

Let ℓ be the tenant's level, a an action, $r(a)$ its class, and H the invariant ceiling of §3.4 (P2). Let θ be the level from which low-risk autonomous execution is granted ($\theta = 3$ in both implementations).

The decision $D(a, \ell)$ is taken on the first row that matches, in order:

Condition	$D(a, \ell)$
$\ell = 0$ and a was not directly ordered	BLOCK
$a \in H$ (at any ℓ)	HUMAN
$r(a) =$ high	HUMAN
$r(a) =$ low and $\ell \geq \theta$ and $a \notin H$	EXECUTE
otherwise	PROPOSE

Table 3: The decision function, evaluated top to bottom.

An action absent from the catalogue takes $r(a) =$ high (P4). Two guards are deliberately redundant: high risk already excludes the members of H, but H is checked first and separately so that a future reclassification error, or a level raised in haste, cannot turn a payment into an automatic operation. In the reference implementations this appears as an explicit comment on the double lock; we consider the redundancy part of the specification rather than defensive clutter.

4.3 Why a matrix beats a dial

A monolithic dial forces the operator to buy exposure in order to obtain value. In the Hubents catalogue, 10 of 24 write actions (42%) are high risk; 9 (37%) are low. Under a dial, the operator who wants internal task upkeep automated must either accept the level that also

releases client-facing email, or forgo automation entirely. Under the matrix, the 37% that is internal and reversible is exactly what V3 grants, and the 42% that is external or financial remains behind a human at every level. The scale's commercial proposition — selling autonomy in levels rather than selling "more intelligence" — is only defensible because of this decoupling: what the customer buys at each level is a bounded, enumerable set of commitments.

5 Reference implementations

Two multi-tenant products implement the scale in production. Both are Next.js/TypeScript applications over PostgreSQL. Hubents (`app.hubents.com`) is an event-management platform whose agent is surfaced as "SofIA"; Koble (`app.koble.es`) is an agency-operations platform whose agent is "KobIA". Koble's implementation was derived from Hubents' after the fact, which bounds the independence claim (§9).

	Hubents (SofIA)	**Koble (KobIA)**
Domain	Event management	Agency operations / CRM
Levels implemented	V0–V5 (V5 reserved)	V0–V4
Published level names	canonical (Table 1)	canonical since 11 Aug 2026 (§6)
Self-service ceiling	V4 (owner/admin)	V3 (tenant admin)
Vendor-granted levels	V5	V4
Default level	V0	V0
Write actions classified	24 (9 low / 5 medium / 10 high)	12 (5 low / 3 medium / 4 high)
Autonomous-execution threshold θ	3	3
Undo window W	7 days	7 days
Invariant ceiling H	1 action (`registerPayment`)	4 actions (payments, reconciliation, fiscal documents, dunning email)
V4 guardrails	12 LLM steps/run, 3 actions/run, 60 total runs, 5 active missions/tenant, blocking escalation, pause/cancel kill switch	same engine, ported
Audit surface	one row per action with risk class, auto-execution flag, undo payload, and who undid it	idem
Regression test pinning the classes	`action-risk.test.ts`	`autonomy.test.ts` (also asserts the ladder has no gaps)
Agent in production since	17 Feb 2026 (268 conversations, 789 messages)	Jul 2026
Scale added	risk engine 7 Jul 2026; self-service ladder 27 Jul 2026	risk engine 20 Jul 2026; self-service ladder 3 Aug 2026
Level distribution (11 Aug 2026)	160 organisations: 157 V0, 1 V1, 1 V2, 1 V4	all at V0

Table 4: The two reference implementations.

Where the gates live. Level enforcement is distributed across three sites, all in code: (i) the scheduled jobs that give the agent initiative are gated by level ($\geq$1 for the daily briefing, $\geq$2 for proposal generation, $\geq$4 for missions), so an under-levelled tenant produces no work at all rather than producing work that is later filtered; (ii) the execution path evaluates $D(a, \ell)$ before every write; (iii) the API that changes the level refuses vendor-reserved levels with an explicit reason, and always permits descent.

The V4 loop. A mission is an objective assigned to the agent ("carry this wedding through to the event date"), advanced in observe–plan–act–summarise passes with persistent working memory. Each pass is capped at 12 model steps and 3 committed actions; a mission is capped at 60 passes, after which it pauses itself and escalates. An unanswered escalation blocks the loop — the agent waits for the human rather than proceeding on assumption. Medium- and high-risk actions raised inside a mission go to the proposal queue exactly as at V2: V4 grants sequencing authority, not new commit authority. This is the concrete meaning of P1 and P2 at the top of the self-service range.

Audit as a precondition, not a feature. Autonomously executed actions are written to the same table as human-approved proposals, flagged as auto-executed, carrying the undo payload needed to restore prior state. The undo affordance is therefore not a separate subsystem that could lag behind the executor; an action that cannot describe its own reversal cannot be classified low, which closes P3 by construction.

6 Findings

6.1 Finding 1: the published ladder drifted from the enforced one

The two implementations ordered the lower levels differently — but only in what they *told the customer*. Hubents publishes proactive *information* at V1 and *proposals* at V2. Koble's catalogue named level 1 "suggests" ("every action lands in the proposal queue") and level 2 "watches" ("daily report of what deserves your attention").

Their code does not differ at all. In both products the briefing job is gated at $\ell \geq 1$, the proposal job at $\ell \geq 2$, autonomous execution at $\ell \geq 3$ and missions at $\ell \geq 4$. Koble's published ladder therefore described level 1 as doing something its own gate withheld, and attributed to level 2 a briefing the tenant already received at level 1. A customer who selected level 1 expecting proposals would have received none, with no error and nothing in the interface to explain it.

This is the paper's most useful single observation, because it is a failure that a level-as-label framework cannot even represent. P7 (enforcement in code) held perfectly — no gate was wrong, no boundary was crossed. P6 (controller legibility) failed completely — the operator could not learn from the product what a level would do. Enforcement and legibility are therefore not two statements of one idea; a system can satisfy either while violating the other, and an autonomy scale that does not separate them will call this deployment conformant. It also shows what the scale is good for beyond design: auditing the two implementations against it is what surfaced the defect, which had been in production since the ladder shipped and which no test could catch, because the tests pin the gates and the gates were right.

The ordering itself resolves under a general principle, which we adopt as the rule for the scale:

> *Minimum marginal authority.* Levels are ordered by the smallest increment of authority granted, where authority over prepared *action* outranks authority over *attention*. A proposal queue is closer to a commit than a briefing is, because it presents a decision already framed for approval — which is where automation bias operates [3].

Hubents' order is therefore canonical, and Koble's catalogue was corrected to match its own gates on 11 August 2026. The repair is worth describing because it generalises: the level thresholds now live as named constants in the same pure module as the published ladder (`BRIEFING_MIN_LEVEL`, `PROPOSALS_MIN_LEVEL`, alongside θ), the scheduled jobs import them instead of carrying the numbers inline, and three regression tests assert the ladder against

them — that the thresholds are ordered inform < propose < execute, that no level below the proposal threshold mentions proposals, and that no level below θ promises autonomous execution. The general lesson is that P6 needs its own tests: a test suite that pins gates cannot detect a description that contradicts them, so the published text has to be asserted against the same constants the gates use.

We report the defect rather than hiding it because it identifies two failure modes for anyone building a ladder: ordering levels by how sophisticated a behaviour *feels* rather than by authority granted, and — more insidious — letting the published ladder drift from the enforced one, which looks like a copywriting detail and is in fact the customer's only source of truth about what they just switched on.

6.2 Finding 2: risk class travels across domains

Seven actions exist in both catalogues: task creation, storing a learned note, contact creation, fiscal document creation, bank reconciliation, payment registration, and dunning email. The two implementations assign identical classes to all seven (2 low, 1 medium, 4 high). Under the four criteria of §4, this is the expected result — externality and monetary effect are properties of the action type, not of the vertical — and it supports publishing the criteria rather than a fixed catalogue: a new domain should be able to derive its own catalogue and reach comparable classes.

The claim must be stated at its true strength. Both catalogues were authored by the same person, from shared design lineage. This is consistency, not inter-rater reliability. Establishing the latter requires independent teams classifying the same action set, which we propose as the first validation study in §9.

The divergence in H is more informative than the agreement. Hubents holds one action as always-confirm; Koble holds four, having promoted three high-risk actions from "needs a human because of its class" to "needs a human by policy, irrespective of class". Both are conformant with P2, which requires H to be non-empty and published, not to have a particular size. The scale should therefore expose H as an operator-configurable set with a floor, and the Autonomy Statement must disclose it.

6.3 Finding 3: a granted level is not an exercised one

The figures below are the production state of the Hubents deployment on 11 August 2026, read from the database rather than from recollection.

Measure	**Observed**
Organisations	160 (157 at V0, 1 at V1, 1 at V2, 1 at V4)
Agent in production since	17 Feb 2026 — 268 conversations, 789 messages
Proposals raised (Jun–Aug 2026)	266 total
dismissed	161 (61 %), none with a reason recorded
still pending	91 (34 %), 87 of them in one tenant
autonomously executed	14 (5 %)
Autonomous actions	7 `createTask`, 7 `updateTask`, 28 Jul – 10 Aug
Undo exercised	0 of 14
Missions run at V4	0
Learned notes stored	2

Table 5: Observed state, 11 August 2026. The V4 tenant is the vendor's own sandbox organisation.

Three things follow, and only the first was anticipated.

P4 holds in practice. 157 of 160 organisations sit at the fail-safe default. Autonomy that must be claimed is not claimed by accident: the deployed control requires an explicit confirmation on the same button. For a property whose purpose is to prevent unintended autonomy, a distribution flat at the default is the success condition, not a failure.

A granted level is not an exercised one. One tenant has been at V4 for weeks and has run *zero* missions. Its actual autonomy is V3's: 14 low-risk actions committed unaided in two weeks. This distinction is invisible to any framework that classifies a deployment by its configured level, and it matters in both directions — a vendor may advertise a level nobody uses, and an auditor reading configuration alone will overstate what the system does. We therefore propose that an Autonomy Statement disclosing a level distribution should report *exercised* autonomy alongside granted autonomy: actions committed per level, not tenants configured per level.

The proposal queue is the weak point, not the executor. 61 % of proposals were dismissed and 34 % are still pending, 87 of those in a single V2 tenant. Meanwhile the 14 autonomous commits produced zero reversals. The part of the system that requires a human on every item is the part that is degrading: the queue accumulates, and dismissals carry no recorded reason even though the field exists to hold one. Read against the ordering principle of §6, this is the automation-bias risk arriving from the direction we did not expect — not humans rubber-stamping proposals, but humans abandoning them.

We report the dismissal rate prominently because it is the number least flattering to the scale, and because it is the one a prospective adopter needs: V3's value proposition is that low-risk work stops needing the queue, and these figures are the first weak evidence that the queue is what breaks first.

What 14 actions do and do not license us to say. They establish mechanism: autonomous execution happens, is audited, and has not been reversed. They do not establish that the risk classification is correct — two weeks and a single tenant cannot distinguish "well classified" from "nobody has looked". The measurement protocol of §9 is written to tell those apart, and the honest status of this paper is that the protocol has begun to run rather than that it has returned a result.

7 The Autonomy Statement

A level claim is worth as much as its verifiability. We propose a ten-field disclosure that a vendor publishes per product, and that a customer or auditor can check against the running system.

1. **Maximum level offered** and, separately, **levels reserved to the vendor**.
2. **Default level** on provisioning (P4 requires V0).
3. **Who may change it**: role, and whether the vendor can change it unilaterally.
4. **Risk criteria** used to classify actions.
5. **Action catalogue** with the class of every write action — the disclosure that makes the level meaningful.
6. **Invariant ceiling** H: the actions that always require a human, at every level.
7. **Undo window** W and the actions it covers, with those classified irreversible named explicitly.
8. **Audit record**: what is stored per autonomous action, for how long, and who can read it.

9. **Gate locations and tests**: where in the code the level is enforced, and which test pins the classification (P7).
10. **Descent latency and kill switch**: how long a level reduction takes to bind, and what it does to work in flight.

Two properties of this instrument matter more than its exact fields. It is *falsifiable* — a customer can attempt a high-risk action at the claimed level and observe whether a human is required — and it is *comparable*: two vendors claiming V3 can be read side by side, and their difference will show up in fields 5, 6 and 7 rather than in adjectives.

8 Mapping to governance frameworks

The scale is not a compliance regime and does not certify anything. It does, however, supply the granularity that oversight obligations leave open.

Obligation	**What the scale contributes**
AI Act Art. 14 — human oversight by design [9]	The level states *at what granularity* oversight is exercised (per action at V0–V2, per action class at V3, per objective at V4). P5 supplies the interruption capability; P3 supplies the record that makes oversight informed rather than nominal.
AI Act Art. 26 — deployer duties [9]	P6 puts the level in the deployer's hands, which is what makes their duty dischargeable; the Autonomy Statement is the artefact they retain.
NIST AI RMF — Govern / Manage [7]	The action catalogue with classes is a risk register at the granularity at which the system actually acts; θ, W and H are named, testable risk controls.
Security review of agentic systems [6]	P7 answers the control-plane question: the boundary is a branch in code with a test, not an instruction a prompt injection can renegotiate.

Table 6: Mapping. The scale is complementary to these instruments, not a substitute.

The international process: what a scale can and cannot contribute

The United Nations does not legislate on artificial intelligence, and this paper claims nothing of the sort. What exists is a mechanism for shared assessment: Resolution A/RES/79/325 (26 August 2025) established the Independent International Scientific Panel on AI — forty members appointed in February 2026 — and the Global Dialogue on AI Governance, whose first session met in Geneva on 6–7 July 2026, informed by the Panel's preliminary report [8]. The Panel's remit is evidence-based scientific assessment, reported annually.

That remit has a measurement problem, and it is the same one this paper opens with. An annual assessment of how autonomously agents are being deployed cannot be assembled from vendor self-description, because "autonomous" is not a comparable quantity across products, sectors or jurisdictions. What such a process can consume is a unit: a level whose meaning is fixed, whose grant is enumerable as a set of actions, and whose claim is falsifiable by inspection rather than by trust. The Autonomy Statement (§7) was designed to be checkable by a customer; the same property makes it usable as evidence by an assessor, which is the difference between a scale and a survey response.

Three properties of the scale matter specifically at that altitude:

- *It interoperates without harmonising law.* A vendor can publish a Statement under the EU AI Act, under a US framework or under no regime at all, and the ten fields remain comparable. An international process needs vocabulary that survives jurisdictional difference, not a treaty per term.

- *It separates autonomy from capability.* Frontier-capability assessment and autonomy assessment answer different questions — what a model can do versus what a deployment is permitted to commit unaided. Conflating them is how a governance discussion ends up regulating model size instead of authority over money.
- *It is auditable in the small.* P7 puts the boundary in code under test, so conformance can be examined at the level of an individual action, not inferred from a policy document. That is the granularity at which an incident actually happens.

The honest limit is the one stated throughout: this is one instrument, validated as a design and disclosure device in two products of a single author, and offered for criticism. If a body of forty scientists finds a better unit, the useful outcome is that a unit exists to argue about.

9 Limitations and threats to validity

Single author, shared lineage. Both implementations are the work of one author, the second derived from the first. Agreement between them (Finding 2) is consistency, not independent replication.

Outcome data are two weeks old on one tenant. The 14 autonomous actions of Table 5 come from a single organisation — the vendor's own — over a fortnight. They license a statement about mechanism, not about efficacy: no error rate, no comparison against the same work done by hand, and a zero undo rate that two weeks cannot distinguish from an audit surface nobody reads. V4 remains unexercised even where it is granted, so nothing here speaks to objective-level delegation at all.

Expert-judgement classes. The three risk classes and the four criteria are engineering judgement, not derivations from a formal harm model. The choice $\theta = 3$ — that low risk becomes autonomous exactly when proposals already work — is a design decision that a different organisation might place elsewhere.

Domain scope. Both deployments are business-management SaaS in the Spanish market. Agents with infrastructure or source-code authority differ in a way the scale does not yet address: there, reversibility is entangled with deployment state, and "internal and reversible" is a far weaker guarantee.

V5 and V6 unexercised. Both are specified and reserved, not validated. V5's coordination semantics — ceiling inheritance across sub-agents and attribution in the audit trail — and V6's composition rules — intersection of authority, union of ceilings, bilateral audit — are the least tested part of the proposal. V6 in particular rests on an untested assumption: that two organisations will publish Statements precise enough to compose mechanically.

Ordinal, not interval. The levels are ordered but not spaced. The V2→V3 step is qualitatively larger than V0→V1, and nothing in the scale suggests otherwise; it should not be averaged, scored, or treated as a maturity index.

Validation programme. Three studies would move the scale from proposal to instrument, in order of value: (1) an inter-rater study in which independent practitioners classify a common action set under the published criteria, testing Finding 2; (2) a longitudinal V3 deployment measuring undo exercise, intervention and reclassification events; (3) an audit study in which third parties attempt to falsify Autonomy Statements against running systems.

Measurement protocol for study (2). We state it here because the instrumentation is a consequence of P3 rather than an addition to it, and any V3-conformant system therefore already has it. The audit row each autonomous action must produce (§5) carries the action, its risk class, an auto-execution flag, the execution timestamp, the undo payload, and who undid it and when. Those fields yield the following measures without further instrumentation:

- *Autonomy volume* — autonomously executed actions per tenant-week, by action type. Establishes whether V3 does anything at all.
- *Undo rate and latency* — share of autonomous actions reversed, and time from execution to reversal. The central measure: a high rate falsifies the risk classification, and a rate of zero over a long window is evidence either of good classification or of an audit surface nobody reads — which the next measure separates.
- *Proposal disposition* — approved versus dismissed, with dismissal reason, at V2 and at V3. A tenant that stops reviewing proposals after moving to V3 is exhibiting the automation bias the ordering principle warns about.
- *Reclassification events* — actions moved between risk classes after experience, which is how the criteria of §4 get tested against reality.
- *Descent events* — level reductions, with the action that preceded them. This is P5 measured in use, and the single most informative signal about whether an autonomy grant was regretted.
- *Escalation latency* (V4 only) — time an unanswered escalation blocks a mission, which bounds how much objective-level delegation is usable in practice.

A study of this shape is honest at $n = 1$ tenant: it yields mechanism, not effect sizes. The threshold it must clear is narrow and was worth stating in advance — that autonomous actions occur, that a non-trivial fraction is never reversed, and that proposal review does not collapse — because a scale whose V3 nobody can use, or whose undo everybody needs, is refuted by its own instrument.

Against that threshold, the first fortnight of data (Table 5) clears the first two conditions and fails the third: actions occur (14), none were reversed (0 of 14), and review is visibly collapsing (61 % dismissed, 34 % pending). We record the failure rather than restating the threshold, because a protocol that only reports its successes is not a protocol.

10 Conclusion

Autonomy is sold today as an adjective. This paper argues it should be sold as a level, and that a level is only meaningful if it is enforced in code, bounded by an invariant ceiling, reversible within a stated window, defaulted to zero, lowerable instantly by the customer, and disclosed in a form that can be checked. The Vergara Scale (V0–V6) is offered as that unit: derived from two orthogonal axes, gated by action risk rather than by level alone, and carried into production in two multi-tenant products whose catalogues, thresholds and adoption distribution are reported here in full, including the finding that almost nobody has moved off the default.

The scale is published for adoption and criticism. Its levels, criteria, properties and Autonomy Statement template may be used, extended or renamed freely; the request is only that implementations state which properties they satisfy, since a ladder without P1–P7 reproduces the ambiguity the scale exists to remove.

Data and code availability

The classification catalogues, thresholds and gate locations described in §5 are drawn from the production source of Hubents and Koble, which are proprietary. Appendix A reproduces both

action catalogues in full so that the classification — the part of the system a third party needs in order to reuse or contest the scale — is public.

Everything a third party needs in order to adopt, extend or contest the scale is published under CC BY 4.0 at github.com/NandoVer/vergara-scale: the Autonomy Statement template and its JSON Schema, the name policy stating what a conformance claim requires, and the machine-readable Autonomy Statements of both reference implementations — 24 and 12 classified actions with their risk class, ceiling, undo window, gate locations and pinning tests — each validated against the schema. The canonical definition of the scale, with the same material in narrative form, is at fernandovergara.com/escala-vergara, and this preprint is archived under DOI 10.5281/zenodo.21893261.

Conformance to this scale is self-declared and deliberately falsifiable: a customer or auditor attempts a high-risk action at the declared level and observes whether a human is required. We publish the instrument rather than operate a certification body, because a scale that needs a gatekeeper to mean something is a scale nobody can check.

A Full action catalogues

Class	Hubents (24)	Koble (12)
Low (autonomous from V3)	create task, update task, post internal message, toggle checklist item, mark notifications read, save learned note, create reminder, schedule meeting, add schedule item	create task, update contact, add tag, schedule follow-up, save learned note
Medium (always proposed)	create contact, create event, set attendance response, set vendor status, create email draft	create contact, move conversation stage, update conversation status
High (always human)	create fiscal document, send document, register payment, reconcile bank transaction, send dunning email, enrol lead in sequence, send gallery link, send email, reply to email, reply in lead conversation	create fiscal document, register payment, reconcile bank transaction, send dunning email
Invariant ceiling H	register payment	create fiscal document, register payment, reconcile bank transaction, send dunning email

Table 7: Both catalogues. Unlisted actions default to high (P4). Hubents exposes 82 tool definitions in total; the 24 above are the write actions the risk engine governs.

B Decision function, as implemented

```
function canAutoExecute(action, level):
    if action in ALWAYS_CONFIRM:        return false   # P2, checked first
    return level >= AUTO_EXECUTE_MIN_LEVEL             # theta = 3
           and risk(action) == "low"

function risk(action):
    return CATALOGUE[action] or "high"                 # P4, fail-closed
```

```
# Dispatch, per action raised by the agent
if canAutoExecute(a, level):
    execute(a)
    audit(a, autoExecuted=true, undo=payload)
else:
    enqueueProposal(a, risk(a))
# An execution that throws degrades to a pending proposal (P4).
```

References

[1] T. B. Sheridan and W. L. Verplank. *Human and Computer Control of Undersea Teleoperators*. Technical Report, Man–Machine Systems Laboratory, Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA, 1978.

[2] M. R. Endsley and D. B. Kaber. Level of automation effects on performance, situation awareness and workload in a dynamic control task. *Ergonomics*, 42(3):462–492, 1999.

[3] R. Parasuraman, T. B. Sheridan, and C. D. Wickens. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics — Part A*, 30(3):286–297, 2000.

[4] SAE International. *J3016: Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. SAE International, latest revision.

[5] K. J. K. Feng, D. W. McDonald, and A. X. Zhang. Levels of Autonomy for AI Agents. *arXiv preprint arXiv:2506.12469*, 2025. `https://arxiv.org/abs/2506.12469`

[6] Cloud Security Alliance. *Agentic AI Autonomy Levels and Control Framework*. CSA, v1 January 2026; v2 March 2026. `https://cloudsecurityalliance.org/blog/2026/01/28/levels-of-autonomy`

[7] National Institute of Standards and Technology. *Artificial Intelligence Risk Management Framework (AI RMF 1.0)*. NIST AI 100-1, January 2023.

[8] United Nations General Assembly. Resolution A/RES/79/325: Terms of reference for the Independent International Scientific Panel on Artificial Intelligence and modalities for the Global Dialogue on Artificial Intelligence Governance. 26 August 2025. Panel members appointed February 2026; first Global Dialogue held in Geneva, 6–7 July 2026.

[9] European Parliament and Council of the European Union. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). *Official Journal of the European Union*, 2024. Arts. 14 (human oversight) and 26 (obligations of deployers).