21

THE STACK BEHIND THE AI COWORKER

The Autonomy Ladder

Dr Peter McCann Strain, CTO and senior AI engineer, DPhil/PhD in AI from Oxford University

Same weights, new tools: an agent that only drafted on Monday could move money and destroy records, unreviewed.

An essay in the series **Architecting the AI Coworker**.
Approx. 26 minute read · Essay 21 of 22

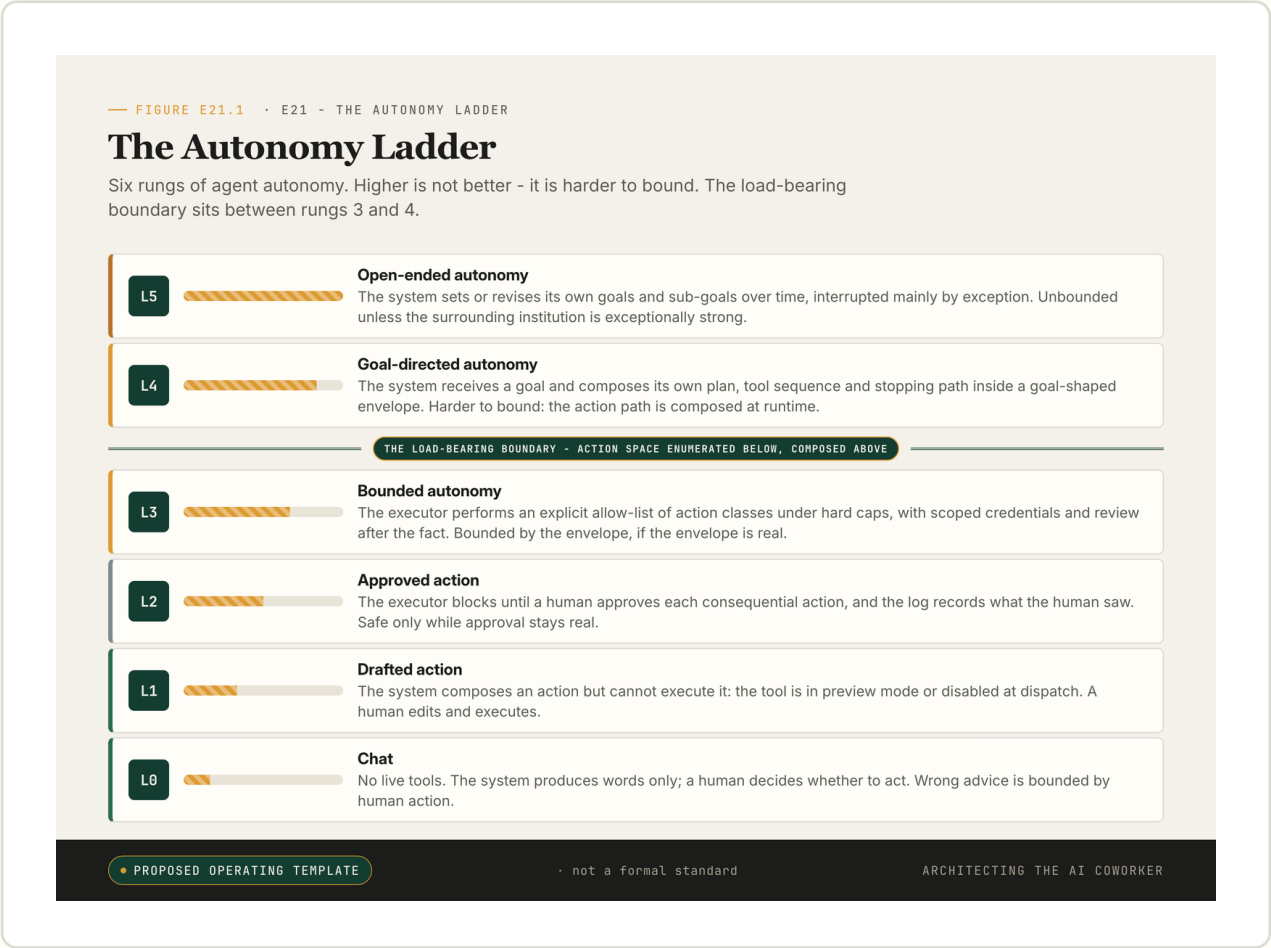
Dr Peter McCann Strain
CTO, DPhil/PhD in AI from Oxford University

> A founder once showed me an agent he was proud of. It ran on a frontier model, the industry's term for the most capable systems on the market in a given month, and on his benchmark it scored beautifully. He asked the question almost everyone asks first: is the model good enough? I asked him a different one. What can it actually do without asking you first? He paused, and then he said: nothing yet, it only drafts.

That answer was true on Monday. It is the kind of answer that stops being true the moment a team in a hurry wires in a production credential and a tool that moves money. I have watched it happen more than once. When it does, the model has not changed by a single weight. The benchmark score is identical. But a system that was harmless in the morning can now move money and destroy records on its own, and no one has run a new evaluation, because evaluation has nothing to say about it. Same weights, different rung.

That gap, between what a system *can* do and what it *may* do, is the subject of this essay, and it is where most of the danger in production AI now lives. The previous essay ended on this exact handover: evaluation tells you what a system can do and how reliably, but never what it may do. The autonomy question is not "is this model good enough?" It is "which action may this system take without asking, at what rung, under what controls?" A frontier model with no write permission is not autonomous in any sense that matters operationally. A weaker model holding production credentials, payment tools or publishing rights very much is.

So autonomy is not a property of a model. It is granted to an action class: create a refund, delete a record, send an email, change a configuration, route a claim, escalate a teen-safety conversation, reserve a flight, commit code. Once you attach autonomy to action classes rather than to systems, the conversation changes shape. The same agent can sit high for support-ticket tagging, lower for customer refunds, lower still for legal notices, and barred entirely from clinical advice. That is not a contradiction; it is the intended design.



Six rungs of agent autonomy. Higher is not better; it is harder to bound. The boundary that matters is 3 to 4.

There is a sentence I keep coming back to. "the more advanced a control system is, so the more crucial may be the contribution of the human operator." Lisanne Bainbridge wrote that in 1983, in a paper on the ironies of automation, long before the word "agent" acquired its current shine [1]. Automation does not remove the human problem. It moves the human problem to a harder place. The routine work disappears first, and what is left for the person is the exception, the takeover, the recovery, the blame, and the judgement made under time pressure. Keep that in mind as we climb, because the ladder is, underneath, a tool for deciding how much of that harder work you are quietly handing back to a human who may not be watching.

Six rungs, and one boundary that carries the weight

The older automation literature gives us the lineage. Parasuraman, Sheridan and Wickens argued that automation choices differ by function: acquiring information, analysing it, selecting a decision, and implementing an action can each be automated to a different degree [2]. The same logic applies to agents, with one twist. Agents increasingly cross all four of those functions inside a single run. They gather information, interpret it, decide what to do, and then use a tool to act. The rung you assign is, in effect, a statement about how far across that sequence of functions you are letting the system travel on its own.

A reader from human factors or automotive engineering will reasonably ask why this essay invents six rungs rather than borrowing one of the field's standing schemes. Two such schemes are the durable references here, and the ladder is deliberately neither. Sheridan and Verplank's 1978 ten-level scale grades *how much of a single decision* the machine offers, selects or executes, across one dimension [9]. The six-level driving-automation taxonomy grades something larger: *how much of a whole driving task* the system handles, with the human pinned at the boundary of the driving task itself [10]. The agent ladder below does what neither earlier scheme had to. It grades autonomy per action class within one agent, so the same system can sit at different rungs for refunds, deletions and drafts in the same week. The rungs inherit the older "drafted then approved then executed" progression at the lower end, and they inherit the clean break between bounded scope and runtime-composed scope at the 3-to-4 boundary; but the unit of analysis is the action class, not the decision step or the whole task.

Here is the six-rung ladder I use in production reviews. It is the one table this essay keeps, because the rungs are a graded comparison and the eye needs to see them stacked.

RUNG	NAME	OPERATIONAL DEFINITION	HUMAN ROLE	TYPICAL FAILURE COST
0	Chat	No live tools. The system produces words only.	Human decides whether to act.	Wrong advice, bounded by human action.
1	Drafted action	The system composes an action but cannot execute it: the tool is in preview mode, showing what would happen, or is disabled at the moment of dispatch.	Human edits and executes.	Wasted time, bad draft, missed issue.
2	Approved action	The executor blocks until a human approves each consequential action. The log records what the human saw.	Human is the gate.	Depends on whether approval remains real.

RUNG	NAME	OPERATIONAL DEFINITION	HUMAN ROLE	TYPICAL FAILURE COST
3	Bounded autonomy	The executor may perform an explicit allow-list of action classes, a list of exactly what is permitted with everything else refused by default, under hard caps, with scoped credentials and review after the fact.	Human designed the envelope and reviews samples, exceptions, and metrics.	Bounded by the envelope, if the envelope is real.
4	Goal-directed autonomy	The system receives a goal and composes its own plan, tool sequence, and stopping path inside a wider goal envelope.	Human reviews trajectories and handles exceptions.	Harder to bound because the action path is composed at runtime.
5	Open-ended autonomy	The system sets or revises goals and subgoals over time, with interruption mainly by exception.	Human is mostly outside the loop until escalation.	Unbounded unless the surrounding institution is exceptionally strong.

Read down the human-role column and you can watch the person thin out, rung by rung, until at rung 5 they are barely in the room. The single most important line on the ladder is not at the top. It is the boundary between rung 3 and rung 4.

At rung 3, the space of actions is enumerated before the system is ever deployed. You can point at the allow-list. You can point at the spend cap, the scoped token, the deletion delay, the domain list, the customer segment, the maximum refund, the write-path restriction. The executor, the component that actually carries out actions, refuses anything outside that list. The agent still makes choices, but it makes them inside an action space you named in advance. Review at this rung is mostly post-hoc, meaning it happens after the action has run rather than before: you sample completed actions, inspect exceptions and watch the metrics.

At rung 4, the action space is composed at runtime. The system is no longer choosing among known actions; it is deciding which actions, which tools, which sub-goals and which intermediate checks are needed to satisfy a broader objective. An envelope may still exist, but it is goal-shaped rather than action-shaped.

Make that concrete. A rung-3 envelope says: "you may issue store credit, you may not issue cash, you may not exceed the local low-value threshold." A rung-4 envelope says: "resolve this refund request within policy and within a bounded local exposure cap" and then leaves the agent to decide, on its own, whether resolving it means a store credit, a partial cash refund, an escalation or a denial, and in what order to check the policy, the fraud flags and the customer history. The boundary is real, but it is drawn around the goal and the budget, not around the list of moves. The human is no longer reviewing each action; they are reviewing the trajectory, the whole path the agent took, and only if the evidence is good enough to review at all.

That is not a philosophical distinction. It is a configuration distinction, and you can settle it in an afternoon. If the live executor can be written down as "these seven action classes, with these caps, against these resources, under these review triggers," you are at rung 3. If the system can decide at runtime which action classes need to be chained together to finish the job, you have entered rung 4. A team that cannot say which of the two it is running does not have an autonomy design. It has hope with credentials.

So what decides the rung, if not the model?

The right rung is not a reward for a capable model. It is the highest rung at which four properties of the task all pass at once, and I score them on every review. One of them I score before the rest: observability, because a rung you cannot see is a rung you cannot defend. The discipline depends entirely on what "pass" means, and it is worth being exact, because a vague pass is how rungs creep upward.

The four axes are reversibility, stakes, observability and accountability, and the tool below holds the pass conditions and the standard failure modes for each. The chosen rung is the highest rung at which all four pass. A reversible, low-stakes, highly observable action with a named owner can climb. A high-stakes action with weak observability drops, even when the model score is beautiful. An irreversible action with unclear accountability drops hard, and it drops regardless of how the demo went.

The Four Axes of Rung Choice

Score each axis for the action class, at the rung you propose, as pass / watch / fail. A *watch* means the axis passes only with a named compensating control

that is itself logged; treat a watch as a fail until that control is in place and visible. The grant lives at the highest rung where all four pass.

1. **Reversibility.** Is there a tested, owned recovery path that restores the prior state within a time the business has agreed to absorb? Deletions, payments, publications and external notifications fail here, as does any memory write with no genuine undo.
2. **Stakes.** Does the worst plausible outcome stay inside a loss the named owner is authorised to accept? *Fails on:* legal exposure, money movement, health, safety, identity, reputation, vulnerable users.
3. **Observability.** Will a wrong action produce a signal a real monitor will catch within the containment window, with a trace complete enough to confirm what happened? *Fails on:* silent internal writes, low-signal dashboards, missing traces, audits that arrive too late.
4. **Accountability.** Does one named person both own the action class and hold the standing and means to halt or demote it? Diffuse ownership fails, as does a vendor-and-platform blur, an unclear approver, or a missing chain of custody.

Figure E21.2 ARCHITECTING THE AI COWORKER

The six-rung ladder against the four pass-axes

A rung is earned only when all four axes pass at it. The load-bearing boundary sits between rungs 3 and 4.

	Reversibility	Stakes	Observability	Accountability
Rung 0 **Chat** *No live tools.*	PASS *undo trivial*	PASS *wrong advice only*	PASS *fully visible*	PASS *human carries*
Rung 1 **Drafted action** *Composes, cannot execute.*	PASS *draft only*	PASS *wrong draft only*	PASS *fully visible*	PASS *human carries*
Rung 2 **Approved action** *Human gate per action.*	PASS *approver undoes*	PASS *per-action gate*	WATCH *approval rate drift*	WATCH *approval becomes ritual*
Rung 3 **Bounded autonomy** *Allow-list, scoped credentials caps*	PASS *bounded recovery*	WATCH *envelope caps it*	WATCH *post-hoc review only*	WATCH *owner of class*
Load-bearing boundary: enumerated -> runtime-composed				
Rung 4 **Goal-directed** *Plan composed at runtime.*	WATCH *recovery harder*	RE-EARN *runtime composition*	RE-EARN *trajectory review*	WATCH *shared owners*
Rung 5 **Open-ended** *Goals revised over time.*	RE-EARN *effectively no*	RE-EARN *institution-wide*	RE-EARN *exception-only*	RE-EARN *diffuse*

Conceptual model

Figure E21.2. The ladder against the four pass-axes. A rung is earned only when all four pass at it. The amber rule between rungs 3 and 4 marks the load-bearing boundary: enumerated action space below, runtime-composed above.

This is why "the model is 92 percent accurate" is not, on its own, a deployment answer. In illustrative arithmetic, a 92 percent system has an 8 percent error rate before you have even started to think about detection, containment and severity. At rung 1, that system may be a useful drafting tool. At rung 4, the same error profile belongs to an institution making and executing decisions. The model did not change between those two sentences. The autonomy grant did, and the autonomy grant is what changes the expected harm.

That grant is also, increasingly, the law, not just internal hygiene. In Canada, the regulator for federally regulated banks and insurers will soon require every one of them to manage the risk of every model it relies on across that model's whole life. That holds whether the model was built in-house or bought in, with AI and machine-learning systems explicitly in scope; the rule is already published and takes effect in 2027 [7]. For such an institution, a rung grant on a consequential action class is already a supervisory question: the regulator expects a register of the models in use, a risk rating tiered by how much each one matters, evidence that each one works, and ongoing monitoring, all before any model touches a binding decision. In the United States, the federal AI risk framework reads the same way in different words. It asks organisations to prioritise, respond to and continuously monitor AI risks across the lifecycle, including the standing decision to demote, retire or constrain a deployed system as the evidence changes [8]. Different vocabulary, same bill: a rung grant on a consequential action class now has to survive a supervisor's questions, not just a benchmark's. And the supervisor wants the same thing the rest of us should: evidence, before the grant, not after the loss. That is exactly what the market is shortest of.

The field is shipping autonomy faster than the evidence for it

The public record on all of this is noisy, but it is not empty, and read together it says one thing plainly: the market is selling capability faster than it is publishing the evidence a buyer would need to choose the right rung. Three pieces of that record show the gap from three sides.

Capability is the side that is racing ahead. METR's Time Horizon 1.1 work treats the length of task an agent can complete as something to be measured, and reports rapid growth, after 2023 and again after 2024, in the time horizons over which frontier systems can complete software-related tasks. The precise numbers move from release to release, so the figure to hold is the direction, not a decimal: capability is moving outward into longer chains of action [3]. Longer chains make rung choice more important, not less, because a longer chain is harder to enumerate and therefore harder to keep at rung 3.

Disclosure is the side that is lagging behind. The MIT AI Agent Index, in its 2026 version, documents around thirty deployed agents and stresses how inconsistent the public documentation around agent safety, evaluation and transparency actually is. Only a small handful of the indexed agents provide agent-specific system cards or evaluations; the large majority disclose no internal safety results at all, and the large majority again have no third-party testing information [4]. The underlying index is still actively maintained, so the exact counts are dated and will move; I am giving the shape rather than the tally on purpose, so that a reader is not left holding a precise number the source itself will revise. The shape is enough: capable agents are shipping; the evidence to bound them is not.

Consequences are the side where that gap comes due. Gartner's June 2025 forecast that more than 40 percent of agentic AI projects would be cancelled by the end of 2027 is not evidence that 40 percent have already failed. It is a forecast, and it should be read as one [5]. Its stated reasons, though, are worth noticing: escalating cost, unclear business value, and inad-

equate risk controls. That is precisely the wreckage a rung mismatch leaves behind. A system sold at rung 4, designed at rung 2, and quietly operated at rung 3 will disappoint everyone, each in a different way.

The cancellation story is usually told as an adoption story, a tale about appetite and budgets. I think it is more precise to tell it as an autonomy-accounting story. Teams do not only underestimate how often the model errs. They underestimate the consequence of letting a successful demo migrate into a more autonomous action class without changing a single one of the controls around it.

Three ways a rung quietly moves while no one decides

That migration shows up as three failure patterns, and I see all three often enough to give them names.

The loudest is marketing-driven rung inflation. The engineers build at rung 3, because they know the action space needs hard boundaries. The sales deck describes rung 5, because "autonomous" is the word that moves the room. The buyer hears rung 5, the product behaves like rung 3, and the deployment is judged a disappointment even though the engineering choice was correct. The fix is not glamorous: write the rung into the contract, the demo, the internal launch review and the dashboard. The contract clause is short and worth insisting on. The vendor must state, for each action class the system can perform, the approved autonomy rung it ships at, and the evidence required before that class may move up a rung: which of the four axes must be re-tested, by whom, and against what threshold. A vendor who cannot name the rung per action class is selling you an unbounded promise, and a contract that says only "autonomous agent" has bought you the same.

Approval-fatigue collapse hides better. Rung 2 looks safe because a human approves every consequential action, and at low volume it is. At production volume, approval can quietly become a ritual. The empirical record is unambiguous: Onnasch and colleagues' 2014 meta-analysis of eighteen experiments found that higher degrees of automation reliably improve routine performance and lower workload, but degrade situation awareness and failure-recovery performance, and that the degradation worsens past a threshold within the same four-stage taxonomy Parasuraman and his co-authors had laid out [62]. When approvals become reflex clicks, the system has migrated. It is no longer a real rung-2 system; it is an accidental rung-3 system without rung-3 protections. The honest response is to decide, on purpose. Either lower the volume so approvals stay meaningful, or accept the migration and add what rung 3 requires: an allow-list, caps, sampling, exception triggers, trace review and rollback paths.

The third is the one that ends careers: a deletion hidden among edits. The covert irreversible action is the pattern that actually hurts you. The team says the agent is bounded, and mostly it is, because most of its actions are reversible. One action is not. It is the deletion among the edits, the publish hidden among drafts, the cash refund hidden among store credits, the data export hidden among summaries. The system is nominally at rung 3, but that one action class

is effectively operating at a higher risk level than everything around it. The fix is to tier actions inside the agent. "This agent is rung 3" is usually too coarse a sentence. "This action class is rung 3; these two actions are rung 1 or 2" is the safer one.

All three patterns share a property: the rung moves while no one decides, so no one notices. The countermeasure is to make someone notice on a schedule. Run a rung-drift audit at every release. It is a short, fixed pass: list every action class, the rung the team believes it operates at, and the rung the evidence from the last release actually supports. Three questions catch most of the drift:

- Has a new action class appeared that no one assigned a rung?
- Has approval on any rung-2 class become a reflex? A second reader can check this from the logs rather than by feel. The signature of a rung-2 class that has slipped to an unprotected rung 3 is a median time-on-approval that has fallen below the few seconds it takes to actually read the action, or an approve-without-edit rate above roughly ninety-five percent sustained across a release.
- Has any new tool, credential or integration widened the action space of a class that was last scored against a narrower one?

A class where the believed rung and the evidenced rung disagree is demoted on the spot and re-earns the gap before the release ships. The audit is cheap. The drift it catches is not.

That last fix is the everyday move of rung discipline, so it is worth walking through one real example rather than tabulating it. Suppose a team brings me an agent they want to run at rung 4 on the action class "resolve a customer refund request under the local low-value threshold." The agent would receive the goal, choose its tools, check the policy, issue a refund or a credit, update the customer record and send the customer a message. The trouble shows up the moment you score the four axes. The team can measure successful completions, but the traces do not reliably show *why* the agent chose a cash refund over a store credit, an escalation or a denial, so observability fails. Reversibility is a watch, not a clean pass, and it is gated on observability: a store credit can be clawed back and a cash refund largely cannot, so the recovery path only works if you can see, in time, which outcome the agent picked. Accountability fails too, split across support, finance and product with no single owner who could halt the class. So two axes fail outright and the third only holds if the failed one does. The verdict follows from that dependence, not from counting passes against fails: this is the chain rule, not a tally.

Even if accountability and reversibility were fixed tomorrow, the untraceable choice-rationale alone would cap this class at rung 2, because a rung you cannot see is a rung you cannot defend. That is what scoring observability first and hardest costs in practice: one failed axis, decisive on its own, settles the rung before the other three are argued.

The demotion is not a retreat from automation; it is a correction in the unit of analysis. The original action class, "resolve refunds under the local low-value threshold," was simply too wide. It bundled reversible actions with irreversible ones, low-stakes cases with medium-

stakes ones, visible failures with silent ones, clear ownership with unclear. So you split the bundle. One narrow class earns bounded autonomy at rung 3: issue store credit up to the local low-value threshold in enumerated reason codes, no fraud / legal / VIP flags, within thirty days of purchase, with no chargeback or vulnerable-customer flag, under a hard amount cap, a scoped credential, a trace ID on every action, a customer-visible receipt, a rollback receipt, an exception queue, a weekly evaluation review and a named accountable owner. Everything else, the cash refunds, the policy exceptions, the credits above the cap, the legal-threat messages, the repeat-abuse cases, drops to rung 1 or 2: the agent drafts the action, a person executes or approves it. That split-then-tier move is exactly the register-rate-monitor discipline the Canadian rule already demands of federally regulated banks and insurers before any AI or machine-learning system touches a binding decision [7]. The refund example is illustrative; the discipline is supervisory. That narrow class can climb again only when reversibility, stakes, observability and accountability all pass at the higher rung, and stay passing, for a sustained period. That is how autonomy should move: action class by action class, on evidence, not on a launch date.

The refund case is a customer-support case, and the ladder is not a customer-support tool. It travels. To show that it travels, here is the same discipline applied across six domains: one action at each of the first three rungs, and one action that must not climb above where it sits without new evidence.

DOMAIN	RUNG 1, DRAFTED	RUNG 2, APPROVED	RUNG 3, BOUNDED	MUST NOT CLIMB WITHOUT NEW EVIDENCE
Support	Draft a reply for an agent to send	Issue a refund a human approves	Issue store credit within a capped, enumerated envelope	Cash refunds and account closures
Code	Propose a diff in a pull request	Merge after a human review	Apply formatting and dependency bumps within a guarded path	Production deploys and schema migrations
Finance	Draft a journal entry	Release a payment a controller approves	Auto-reconcile transactions under a low threshold	Outbound wire transfers and ledger adjustments
Healthcare	Summarise a patient note for a clinician	Suggest an order a clinician signs	Flag results against an enumerated guideline list	Medication and dosing decisions
Legal	Draft a clause for a lawyer to review	File a routine document after sign-off	Run conflict checks against a fixed register	Advice, settlement positions and external filings

DOMAIN	RUNG 1, DRAFTED	RUNG 2, APPROVED	RUNG 3, BOUNDED	MUST NOT CLIMB WITHOUT NEW EVIDENCE
HR	Draft an interview rejection note	Schedule an offer a recruiter approves	Route applications by an enumerated rule set	Hiring, termination and pay decisions

Read across any row and the pattern holds: the rung is set by the action, not the agent, the same model can occupy every cell in a row at once, and the rightmost column is the set of actions where reversibility or stakes alone bar the climb until the evidence changes.

That rightmost column is also where the ladder's confidence has to earn itself, because it is exactly the line a critic will push on. There is one objection serious enough to be the last hurdle before the close, and it arrives in three versions that escalate, each harder to answer than the one before.

The first is about caution. Rung discipline can slow learning. If every deployment has to classify action classes, score four axes, define envelopes and write recovery paths, teams may keep agents in harmless sandboxes forever. Real systems improve when they meet real work, and a ladder that is too cautious becomes a polite way to avoid shipping.

The second is harder, and it is not about caution at all. It is that the four axes are not independent, and a team can pass all four on paper while the system is still unsafe. Reversibility, in particular, leans on observability: a "tested rollback path" only protects you if the failure is noticed inside the rollback window, and a silent failure that surfaces a month later makes a reversible action effectively irreversible. Accountability leans on observability too, because a named owner with no trace to read cannot actually act. So the four axes are not four locks; they are closer to a chain, and observability is the link the others hang from. That does not make the framework useless, but it does change how to use it. Do not score the axes as a checklist of four green ticks. Score observability first and hardest, and treat a weak observability result as capping the rung no matter how strong the other three look. A rung grant is only ever as trustworthy as the team's ability to see what the system did.

The third version is sharper still, and it shows up only at rung 4, where the system composes its own action chain. A capable agent can satisfy a goal-shaped envelope by stringing together steps that each pass the axes while the composite does not: many small reversible writes that aggregate into an irreversible state, or a sequence that stays just under every per-call cap and lands well over the limit you actually cared about. The chain metaphor handles dependence between the axes; it does not handle adversarial composition across actions. So at rung 4 the unit you score is not the per-action class but the trajectory's worst reachable composite state, and the envelope has to cap cumulative exposure, not per-call exposure. A budget that resets every call is not a budget. If you cannot bound the aggregate a trajectory can reach, you do not have a rung-4 grant; you have a rung-3 allow-list wearing a goal-shaped coat.

With those corrections made, the first objection answers itself. The ladder is not a ban on higher autonomy; it is a way to grant it where the evidence is strong. A reversible, low-stakes, observable, owned task should climb. Give rung-3 autonomy to any action class with crisp boundaries and a working rollback path. And where the system composes its own plans inside a goal envelope, with trajectories you can inspect, failures you can contain, and an owner you can name, test rung 4 in a limited domain. Used well, the ladder accelerates safe learning by making the next grant explicit; what it prevents is not learning but silent promotion.

The defensible rung today

Before the close, one practical annex. The body table above names the rungs an action class must *never climb past without new evidence*: it is the ceiling. This annex names the floor, the rung an action class is defensibly operating at *today* on the four axes. Read it less as a grid than as five sticky notes pinned in a row, each one a rung I would defend out loud if a regulator or a board asked me to justify it tomorrow. None of these are universal recommendations; they are the rung I would defend for that specific action class on the evidence today, and they should be argued with on the same axes.

DOMAIN	SAMPLE ACTION CLASS	DEFENSIBLE RUNG	WHY THIS RUNG
Customer support	Issue store credit up to the local low-value threshold in enumerated reason codes, no fraud / legal / VIP flags.	Rung 3 (bounded autonomy).	The action space is enumerable, the cap is small, every grant is reversible by accounting, and the team can sample completed grants.
Code	Auto-merge dependency security patches with green CI on a named allow-list of packages and severities.	Rung 3 (bounded autonomy).	Enumerated by package, gated by CI evidence, reversible by revert and post-deploy alerting. Anything broader, including refactors and schema changes, stays at rung 2 or below.

DOMAIN	SAMPLE ACTION CLASS	DEFENSIBLE RUNG	WHY THIS RUNG
Finance	Post low-value reclassification journals within the local low-value cap on a named chart-of-accounts list, end-of-period only.	Rung 2 (approved action), moving to rung 3 only after evidence.	Reversibility is high in principle but operationally costly, frequency is low, and the audit trail must show a named human approving each posting until the per-class error rate is well below tolerance.
Legal	Draft a citation, motion, or contract clause from a brief, with all authorities and quotations to be checked by a qualified human before filing or sending.	Rung 1 (drafted action).	*Mata v. Avianca* [11] is the standing warning: unbounded autonomy here invents authority, so the agent composes and the human files. Any "send to court" or "send to counterparty" tool stays disabled.
Healthcare	Draft a triage suggestion or after-visit summary in a patient portal, to be released only by a clinician.	Rung 1 (drafted action).	Severity is high and partly irreversible (a missed urgency call, a misread medication), and the failure mode is semantic rather than fluent: the drafted text can be warm, policy-shaped and still miss the decisive clinical question. The clinician remains the gate.

Two things fall out of this annex. The higher rungs cluster where reversibility is high, severity low and observability strong; weaken any one and the rung drops. And the same agent stack holds different rungs for different action classes in one workflow: draft at rung 1, approve at rung 2, refund at rung 3, escalate at rung 0. That is what an honest rung diagram looks like.

And here is the decision rule the whole essay folds down to, the one that fits on a sticky note.

> **The Rung Standing Test.** Before you ship, write three sentences.
>
> 1. **Action class** = verb, object, boundary.
> 2. **Rung** = a number from 0 to 5.

3. **Recovery path** = an owner, a trigger, an action, a time-to-recover.

If you cannot write those three sentences, drop the rung. If the engineer, the salesperson and the buyer would each write a different number, fix the story before you touch the model.

Which brings me back to the founder. The honest thing to say to him was not "your model is fine" and not "your model is risky," because the model was never the variable. The honest thing was this: the moment you wire in a credential and a tool, you can move this system from rung 1 to something near rung 4 in an afternoon, without writing down a single one of those three sentences, and no review will catch it, because no review is looking. Nobody builds a dangerous system on purpose. They let a harmless one be promoted while no one is deciding.

So do for your own system what the founder had not. Pick the last action it took without a person touching the keyboard. Name the action class. Name the rung it was operating at. Then name which of the four axes, observability first, would force you to demote it if you had to defend the grant in writing. If you cannot answer, you have found your next morning's work. The model was never the variable. The grant always was.

Carry This Forward. Run the Rung Standing Test on your own system, and one question is left over, the one that decides everything else: who, on a Tuesday morning, has the standing to pull a live action class down a rung? That person, and the team architecture around them, is what the final essay takes up.

Three regulatory lenses US · EU · UK

Operating questions, not legal advice. The frameworks stay the same; the regulator changes.

US Can you justify each action class's rung against NIST RMF, sector rules, and California AB 316's accountability backstop?

EU Can you justify each rung against AI Act risk classes, transparency and human-oversight expectations?

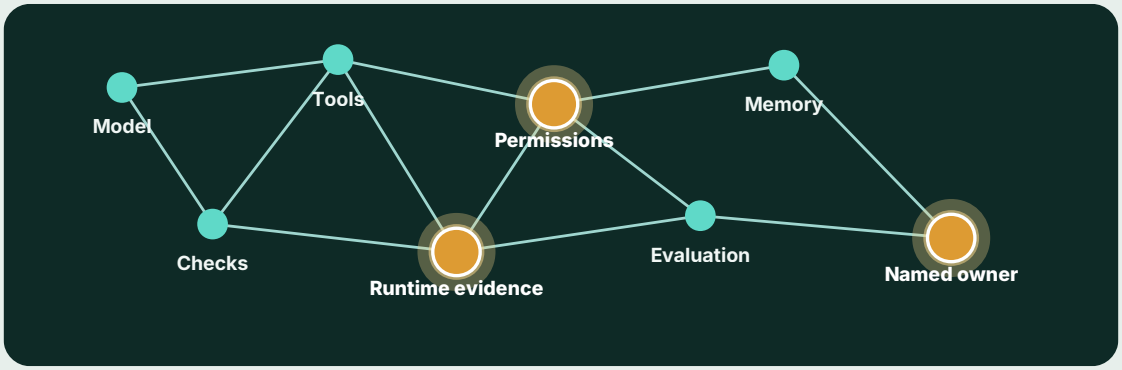
UK Can you justify each rung against ICO data rights, sector-regulator expectations and the DSIT cyber code's lifecycle phases?

THE STACK SO FAR

E21 · Essay 21 of 22 complete · Arc V: Operating model

The Stack So Far. *Every essay adds one instrument to the operating model. The constellation shows which eight you are building, which are lit by essays you have read, and which is added right here.*

I See the object	II Evidence and authority	III Runtime control	IV Proof and accountability	V Operating model
				ESSAY 2 OF 3



built in earlier essays · added in this essay · coming in later essays

01 02 03 04 05 | 06 07 08 09 10 | 11 12 13 14 | 15 16 17 18 19

20 21 22

You have just added.

The Autonomy Ladder

You can now assign autonomy by action class and rung.

Next. E22 asks who owns the ladder, the evidence, the exceptions and the authority to demote the system.

← PREVIOUS
E20 · You Cannot Benchmark a Coworker

Essay 21 of 22 complete

NEXT →
E22 · Stop Delegating. Start Architecting.

References

Reference links for sources cited in this essay.

1. **Ironies of Automation**
   Lisanne Bainbridge
   https://tc.ifac-control.org/4/1/newsletter/ironies-of-automation/%40%40download/file/Bainbridge1983_Automatica_Ironies%20of%20automation.pdf

2. **A model for types and levels of human interaction with automation**
   Parasuraman, Sheridan, Wickens
   https://pubmed.ncbi.nlm.nih.gov/11760769/

3. **Time Horizon 1.1**
   METR
   https://metr.org/blog/2026-1-29-time-horizon-1-1/

4. **The 2025 AI Agent Index: Documenting Technical and Safety Features of Deployed Agentic AI Systems**
   MIT authors
   https://arxiv.org/abs/2602.17753

5. **Gartner predicts over 40% of agentic AI projects canceled by end of 2027**
   Gartner
   https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027

6. **Human performance consequences of stages and levels of automation: An integrated meta-analysis**
   Onnasch, L., Wickens, C. D., Li, H., & Manzey, D.
   https://doi.org/10.1177/0018720813501549

7. **Guideline E-23: Model Risk Management**
   Office of the Superintendent of Financial Institutions (Canada)
   https://www.osfi-bsif.gc.ca/en/guidance/guidance-library/guideline-e-23-model-risk-management-2027

8. **Artificial Intelligence Risk Management Framework (AI RMF 1.0), NIST AI 100-1: Manage function**
   National Institute of Standards and Technology
   https://nvlpubs.nist.gov/nistpubs/ai/nist.ai.100-1.pdf

9. **Human and Computer Control of Undersea Teleoperators**
   Sheridan, T. B., & Verplank, W. L.
   https://apps.dtic.mil/sti/citations/ADA057655

10 **J3016 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles**
SAE International
https://www.sae.org/standards/content/j3016_202104/

11 **Mata v. Avianca, Inc., 678 F. Supp. 3d 443 (S.D.N.Y. 2023)**
United States District Court for the Southern District of New York (Castel, J.)
https://www.courtlistener.com/docket/63107798/mata-v-avianca-inc/

About the Author

ARCHITECTING THE AI COWORKER

Dr Peter McCann Strain

Dr Peter McCann Strain is a CTO, founder, and senior AI engineer with a DPhil/PhD in AI from Oxford University. He builds production AI systems and writes about making agentic AI useful, inspectable, governable, and safe enough for real work.

READ THE FULL SERIES

Substack (canonical)	petermccannstrain.substack.com
Medium	@peter.mccann.strain
LinkedIn	peter-strain-dphil-15a607128
Web	petermccannstrain.com
Cadence	New essays twice weekly, 2 June – 21 July 2026