18

THE STACK BEHIND THE AI COWORKER

When an Agent Acts, Who Acted?

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

An agent buys you wrong shoes and the charge clears. Who authorised that, and who pays? Seven links decide.

An essay in the series **Architecting the AI Coworker**.
Approx. 27 minute read · Essay 18 of 22

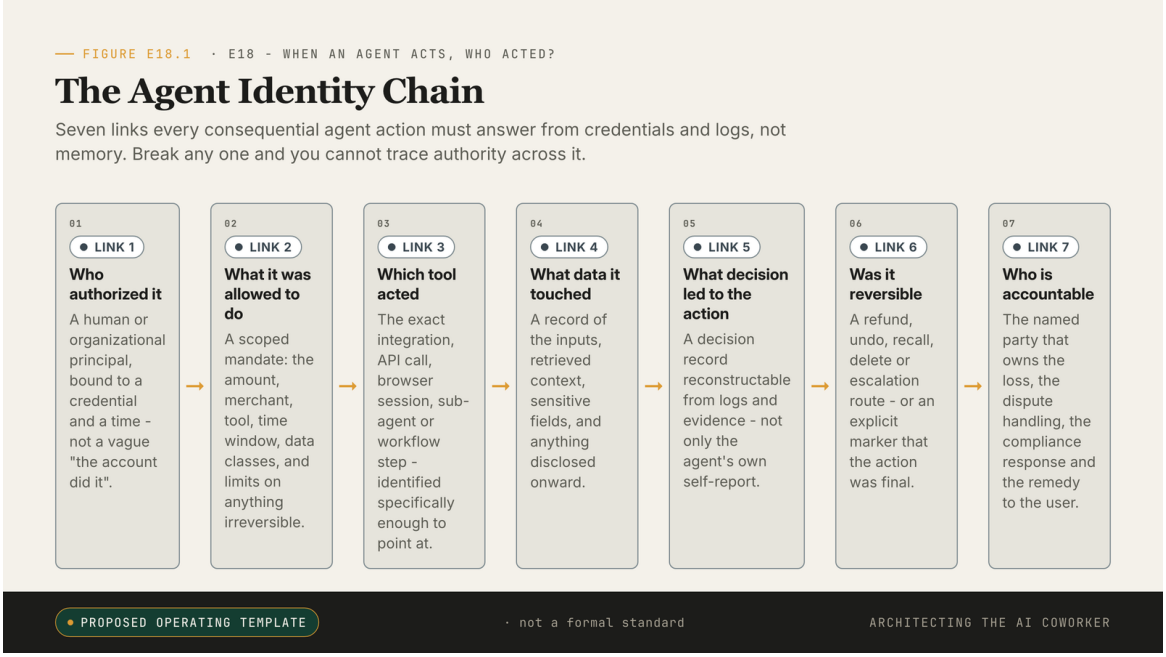
Dr Peter McCann Strain
CTO, DPhil/PhD in AI from Oxford University

> Picture a Wednesday in March. Someone asks their agent to buy a pair of running shoes. By agent I mean software that takes a goal, breaks it into steps, and acts in the world on a person's behalf, not a chatbot that answers and stops. The instruction is specific in the way a person is specific to an estate agent or a tailor: the brand, the size, a rough budget, the marathon in eight weeks, the left knee that is, in the technical sense, a problem. By morning there is a box on the doorstep. They open it. The shoes are the wrong colour, half a size too small, and $40 more than they would have paid sitting at the laptop. The charge has already cleared. The bank app, which has the emotional intelligence of a smoke alarm, has sent a polite chirp.

So they ring the bank to dispute the charge, and the call stalls on the first question. The adviser asks, reasonably, whether they authorised the transaction. And there is no clean answer. They want to say no, this is not the purchase they meant. But they also have to say yes, they told a piece of software to buy them running shoes and gave it their payment details to do it. They authorised the agent. They did not authorise that decision. The adviser has a form with two boxes on it, and the situation needs a third. There is a box on a doorstep, a charge on a statement, a person who gave an instruction and never clicked a button, and a question that becomes harder the longer you look at it: when an agent acts, who acted?

I composed that scene to make an abstract problem physical. It is illustrative, not a report of a single real dispute, and the rest of this essay is about the third box the bank's form is missing.

The previous essay argued that an agent's trace is evidence, not truth, and ended on a handoff: evidence can show you the path an action took, but not whose authority travelled along that path. This essay takes up that question. When an agent can buy, reserve, file, send, delete or move money on your behalf, the old test, "did the user click?", stops being enough. You need an identity chain: who authorised the action, which platform accepted it, what tool carried it out, what data it touched, and who owns the consequences when it goes wrong.



The Agent Identity Chain: seven links running from the human who authorised an action to the party that owns its consequences.

Did they buy those shoes?

In one sense, plainly yes. The intent was expressed. They wanted shoes. They supplied the constraints. In another sense, plainly no. They never saw the merchant's page, never inspected the return policy, never approved the exact price, and never decided that this particular shade of blue was acceptable for the marathon. The bank adviser's form cannot hold both of those at once, and that is the problem in miniature. They authorised the agent, but not that decision. They authorised the decision, but not the platform access that made it reachable. The gap between those two sentences is the whole essay: it is where the charge cleared, where the platform's objection lives, and where the bank's two-box form runs out of boxes.

The discomfort of that question is not a quirk of shopping. It is the shape of every consequential agent action. The seven-link chain below, and the worked table that follows it, make the spine of the argument operational: every consequential action has more than one principal whose consent the action required, and an architecture that records only one of them has hidden the rest.

This is not a thought experiment any more. As I write, in the spring of 2026, agents are quietly buying things for people. They are also opening tickets, sending emails, filing expenses, scheduling interviews, deleting files, and moving small amounts of money between accounts. Each of those actions lands somewhere. Each has a sender, a receiver, and a consequence for a real person whose day has now been altered. For most of the chatbot era, the default safety question was a question about speech: did the model say the right thing? The whole apparatus

of content filters and refusal tuning was built around that. Once an agent can take a consequential action, the centre of gravity moves. The harm is no longer only an utterance. It is a side effect. The relevant question becomes did the right person cause this, and, when it goes wrong, can we prove who did.

That is, in the most literal sense, an identity problem and a delegated-authority problem at the same time. NIST, the United States standards body, launched its AI Agent Standards Initiative in February 2026 and named three things that have to be true at once for agents to be more than a demo: that they can be adopted with confidence, can function securely on behalf of users, and can interoperate across the digital ecosystem [1]. Those three words are load-bearing. "On behalf of users" is a delegation claim. "Securely" is an identity claim. "Interoperate" is a standards claim. If a system cannot say who carried authority across a boundary, it is not an agent stack. It is a chain of hopeful clicks.

> **A note on currency.** This essay rests on legal proceedings and standards efforts that are still moving. Every named case, regulator and protocol is current as of May 2026, and the dated status of each is set out in the box at the end. Read the analysis with that posture in mind: the load-bearing argument is the structure of the identity chain, not the procedural date of any one filing.

Trace the authority through seven links, or you do not know who acted

So "who acted?" has to become an operational question, not a philosophical one. It cannot simply mean "whose account produced the instruction?" That is only the first link. A serious answer follows the authority all the way: from the human to the agent, from the agent to any sub-agent or tool, from the tool to the external system, and from that system out to the person or organisation affected. Every consequential action should be able to answer seven questions, and answer them in something close to real time, from credentials and logs rather than from memory.

The chain runs from the human who authorised the action, through the mandate that scoped it, the tool that carried it out and the data it touched, to the decision behind it, the route to undo it, and the party that owns the consequence. The figure above draws those seven as a single chain because that is what they are; the box below states each one precisely.

The Agent Identity Chain

For any consequential agent action, the system must be able to answer all seven from credentials and logs, not from memory:

1. **Authorising principal.** The human or organisational principal, bound to a credential and a time.
2. **Scoped mandate.** What the agent was allowed to do: amount, merchant, tool, time window, data classes, irreversibility limits.
3. **Acting tool.** The exact integration, API call, browser session, sub-agent or workflow step.

4. **Data touched.** The inputs, retrieved context, sensitive fields and anything disclosed onward.
5. **Decision record.** Reconstructable from logs and evidence, not only the agent's self-report.
6. **Reversibility path.** Refund, undo, recall, delete, escalation, or an explicit marker that the action was final.
7. **Accountable owner.** The named party that owns the loss, the dispute, the compliance response and the remedy.

Break any one link and you cannot trace authority across it.

I have started calling this the Agent Identity Chain. It is not a diagram for a standards meeting. It is a procurement test.

The seven-link enumeration is my own coinage, but its load-bearing pieces are not. The authorising-principal, scoped-mandate, acting-tool, data-touched and accountable-owner links map directly onto the three criteria NIST's initiative names as preconditions for agent deployment, namely "adopted with confidence", "function securely on behalf of users" and "interoperate across the digital ecosystem" [1]. They map, too, onto OWASP's catalogue of how agents fail when they are handed too much room: too many permissions, too much functionality, too much autonomy, which is exactly what a delegated agent's scope mandate has to refuse [9]. The decision-record and reversibility-path links sit inside the EU AI Act's duty to keep event logs for high-risk systems [11] and Quebec's right to learn the main factors behind an automated decision [8]. What the seven-link chain adds is the procurement reading: the questions a buyer can ask before signing, in an order the receiving system can answer one at a time.

Take one agent action your company would not want to read about on the front page of a newspaper. If you cannot answer these seven questions for it from credentials and logs, you do not yet know who acted, and who acted is the question that decides who pays.

A courtroom makes the two-principal split concrete

The sharpest public warning that the chain is real, and not merely tidy, has landed in a courtroom. In *Amazon.com Services LLC v. Perplexity AI, Inc.*, Amazon sued Perplexity in the Northern District of California in November 2025 over Comet, Perplexity's AI shopping agent, alleging that the agent accessed Amazon's systems without authorisation. On 9 March 2026, Judge Maxine M. Chesney signed a preliminary-injunction order. A preliminary injunction is an interim ruling, not a final decision on the merits, so the case is not over. But the operational sentence in that order is the one every agent team should read. At the preliminary-injunction stage, the court described the agent's access as occurring "with the user's permission, but without authorization by Amazon" [2].

That sentence separates the two principals exactly. The user could permit the agent to act with the user's own account, data and payment instrument. The platform could still withhold authorisation for that agent to enter its password-protected areas, to disguise itself as ordinary

human browsing, or to use its commercial workflow in a way the platform rejects. User permission does not automatically supply platform authorisation [2]. The appeal posture is live: the Ninth Circuit appeal appears in the public docket summary as No. 26-1444, recording a stay pending appeal and expedited briefing, though the precise procedural dates should be confirmed against the official docket [3]. A stayed injunction is persuasive reading, not operative law, so the argument here does not depend on how the appeal lands. It depends only on the warning the district-court order makes plain. If your agent borrows a user's logged-in session and treats that as complete authorisation, the system on the other side may have a very different view, and a court may agree with it.

The user-versus-platform split is not a doctrine the Northern District invented, which is why the chain stands even if the Ninth Circuit reverses tomorrow. It is a structural feature of every system where one principal grants delegated access and a second principal owns the resource, and every privacy regime reaches the same split by a different route. Canada's federal law pins accountability on the deploying organisation, however the action was carried out [7]. Quebec hands the affected person a right to know a decision was automated and to demand the main factors behind it [8]. The EU AI Act makes the deployer keep event logs and nominate human oversight that can actually intervene [11]. Reverse the injunction and the platforms that block agentic traffic still block it; the regulators that ask "who authorised this action under which mandate" still ask. The chain survives the loss of any one anchor because it is doing identity engineering, not appellate prediction.

California has now closed one of the escape routes directly. A law signed on 13 October 2025, effective from the start of 2026, prevents a defendant from claiming an AI autonomously caused harm to the plaintiff, while preserving the ordinary causation and foreseeability defences [10]. Every regime asks the same upstream question in a different accent (who authorised what, through which tool, under which mandate, with what reversal path), so a system built to answer the chain answers all of them at once. The lenses differ; the underlying engineering question, unchanged by jurisdiction, does not.

The chain answers "who", not "whether they should have"

That courtroom sentence is also the place to be honest about what the chain cannot do, because the Perplexity case shows the limit as sharply as it shows the strength. The chain has two honest limits, and it is cleaner to admit both than to defend it as if it had none.

The first is foreseeability. Protocols capture what was authorised, not what was reasonably foreseen, so a system can answer the seven questions cleanly and still produce harm a court would say the operator should have seen coming. The shopping agent shows it: an operator could answer all seven identity-chain questions cleanly (the user did sign in, the mandate did say "buy shoes", the tool was a known browser automation, the data accessed was the user's own session, the decision record was preserved, the order was refundable, and the agent operator was named) and still face the objection that crossing into Amazon's password-protected area on borrowed credentials was a foreseeable harm the scope mandate should have re-

fused. Identity says who acted; foreseeability asks whether that scope should have been granted in the first place. Cleanly answering "who acted?" does not end the inquiry. It is the precondition that lets the rest of the inquiry begin, answered by the people who decided what scope the agent was allowed to have.

The second limit is sharper. Picture the insider: an employee who tells the agent to exfiltrate a customer list it is entitled, day to day, to read, or a user who mandates a purchase precisely so they can later dispute it. The chain answers "who acted" most cleanly in exactly that case, because the action was legitimately authorised, and by recording it faithfully the chain can appear to launder the harm. The honest framing is that the chain is an attribution mechanism, not an intent detector. It makes the authorised principal undeniable, which is exactly what a fraud or insider-threat investigation needs as its starting evidence, but it relocates the trust question rather than removing it. Knowing beyond dispute who authorised an action is the precondition for asking whether they should have, not a substitute for it.

So the chain is one layer of three. It composes with a triage, set out next, that decides which kind of evidence each link can carry, and with a foreseeability layer that lives outside the chain entirely, in the scope mandate, the risk tier and the named owner. The chain is the identity question, made answerable. The other two are separate questions, and they begin where it ends.

Four emerging standards each answer a different link in the chain

The encouraging news is that the industry is not starting from zero. Four serious efforts have crystallised in parallel, each addressing a different link in the chain. But it is worth being blunt about how unevenly they have matured, because a protocol that has shipped and a protocol that has been announced are not the same kind of evidence, and a vendor will happily blur the two.

AP2, the Agent Payments Protocol, gives payment-grade language for bounded user authority. Its mandate model is designed so an agent can carry verifiable evidence of exactly what the user authorised, and so that role boundaries can be checked by deterministic code, fixed rules that give the same answer every time, even though the shopping agent itself is not deterministic [4]. To the merchant on the receiving end, a signed mandate arrives as a small cryptographically sealed token that pins the buyer, the spending limit, the item and the time window, and refuses to verify if any of those is altered. In plain terms, "the user told me to buy shoes" becomes a bounded object a merchant or a payment network can inspect rather than a sentence it has to trust. That is the strongest of the four on paper, but the published material is a specification, and a specification is a design, not a deployment; the question to ask is which payment networks and merchants have actually implemented the mandate model end to end, not whether the document is elegant. ACP, the Agentic Commerce Protocol, maintained by OpenAI and Stripe, is further along in one narrow sense: it is openly licensed and running in beta with real checkout traffic, which is more than a draft. Its design choice is to keep the business as the merchant of record, the party that owns the customer relationship and the

transaction, while letting agents and payment providers pass constrained information through the checkout [5]. "Beta" is the honest word, though, and a beta protocol is not yet a stable contract.

The other two efforts sit further from a usable answer. FIDO, the authentication body behind the passkey, the phone-or-laptop login that replaces a password with a cryptographic key the device holds, is the least mature of the four. It has opened agentic authentication and payments workstreams, observing that today's authentication models were built for direct human interaction and that service providers now need interoperable ways to verify a user's intent, conditions and limits for delegated agent actions [6]. That is an announcement of intent, not a usable standard: the working groups are newly formed and have not produced final specifications, so anyone citing FIDO for agent identity today is citing a promise. NIST CAISI, the standards initiative described above, is not a protocol at all; it is a coordination effort trying to make agent identity a public agenda rather than a contest between products [1]. Coordination is valuable and also slow, and it ships guidance, not mechanism.

So the four answer different links, and they answer them at different stages of readiness. AP2 is strongest on the user mandate and the payment evidence, but mostly as a specification. ACP is strongest on the merchant-of-record relationship and who owns the checkout, and is actually running, in beta. FIDO is where user authentication, agent authentication and delegated instructions are being standardised, and is the furthest from a shippable answer. NIST is the public coordination layer, and produces agendas rather than code. None of them, alone, answers every question, none of them is finished, and a buyer who treats any of the four as a solved problem has mistaken a roadmap for a road. The field is moving as I write, too, and the dated detail of where each effort currently stands is set out in the status box at the end of this essay; anyone wiring an integration today against a snapshot of these efforts should expect the snapshot to change.

To see why the chain matters, run the running shoes through it twice. The same purchase looks entirely different in a protocol world and in a borrowed-session world. A borrowed session is the common shortcut where the agent simply logs in as you, driving your already-authenticated browser or reusing your stored credentials, so that to the platform on the other side the agent is indistinguishable from you at the keyboard. It is borrowed because the agent never had an identity of its own; it is passing through the world under your name, on your credentials, indistinguishable from you. The two worlds answer the chain side by side:

CHAIN QUESTION	PROTOCOL WORLD	BORROWED-BROWSER/SESSION WORLD
Who authorised it?	A signed user instruction or mandate binds the human, device, time, and purchase intent [4 6].	A chat message says "buy shoes"; whoever controls the session can type it.

CHAIN QUESTION	PROTOCOL WORLD	BORROWED-BROWSER/SESSION WORLD
What was it allowed to do?	Scope is explicit: road shoes, size 10, under $200, delivery by Saturday, one purchase, no recurring authority [4].	Scope lives in prose and inference. The agent decides what "rough budget" means.
Which tool acted?	A merchant-approved checkout integration receives a verifiable request; the merchant can accept or reject it [5].	A browser automation uses the user's logged-in session; the platform may not recognise the agent as authorised [2].
What data did it touch?	Data is minimised to purchase-relevant fields: size, budget, address, payment credential, checkout state [45].	The session may expose account history, recommendations, addresses, subscriptions, and other context beyond the task.
What decision led to the action?	The cart can be checked against the mandate and logged with the chosen item, price, merchant, and payment proof [45].	A receipt exists, but the decision path may be a transcript plus opaque browser events.
Was it reversible?	The merchant of record runs ordinary refund handling, and the transaction carries dispute evidence [54].	Remedy depends on platform policy, the bank's dispute process, and whether anyone accepts responsibility.
Who is accountable?	Accountability can be allocated across user, agent operator, merchant, and payment participant using signed evidence [45].	Everyone has a plausible sentence: the user asked, the agent clicked, the platform objected, the bank processed.

The protocol world is not perfect. Mandates can be drawn too broadly. Logs can be incomplete, and merchants can implement a protocol badly. And the failure a practitioner will actually hit is subtler than any of those: the mandate is correctly signed and correctly scoped, but the human approved a scope they did not really understand. Picture a user who signs an AP2 mandate reading "buy running gear under $200" and the agent dutifully buys a $190 item the user would never have chosen at the laptop. The chain answers all seven questions cleanly, the principal is bound, the scope is explicit, the tool is approved, the decision is logged, the purchase is even refundable, and the user still has no real recourse, because consenting to a scope is not the same as consenting to every action inside it. This is why "the mandate was signed" is the start of the accountability story, not the end of it, and why the design has to keep the scope narrow enough that consent to it means something.

Commerce is also the easy case, because payments already have mature dispute rails behind them. The honest claim is narrower: the protocol world makes the seven questions answerable in principle, and the borrowed-session world tends to make them answerable only after a fight.

That matters because most agent actions are not purchases. An agent filing an expense, deleting a file, posting a message in a team chat, opening a support ticket or touching a patient record needs exactly the same chain, and the payment protocols do not solve those domains for you. What they give you is the pattern: bounded authority, platform-side acceptance, constrained tools, audit logs, a revocation path, and an accountable owner.

The two-principal split shows up everywhere once you look for it, and rarely as commerce. An agent that files an expense carries an employee's mandate, but it is the finance system that authorises a claim against the budget. An agent that drafts a support credit has the user's permission to compose, but only the finance tool authorises money leaving the company. An agent scheduling interviews holds delegated email authority and touches candidate data, yet the candidate, the calendar platform and data-protection law each set terms it must satisfy. An agent deleting files acts on a user instruction, while the repository owner authorises what may actually be removed and from where. An agent sending a customer email becomes the brand's speaker, needs internal approval, and produces an external effect that lands in someone else's inbox. In each case there is a person who gave the instruction and a system that decides whether the instruction may take effect, and an architecture that records only the first has hidden the second.

Walk that pattern through something deliberately mundane, an agent that triages support tickets. A support manager authorises it to draft and tag tickets during business hours: that is the principal and the time-bound mandate. Its scope is written down and narrow, it may classify priority, draft replies and propose account credits below a cap, but it may not touch legal-threat, safety, fraud or vulnerable-customer cases. The tool that acts is the support-system integration that writes the tag and the draft, and credit issuance is deliberately a separate tool behind a separate approval, so the agent cannot quietly graduate from drafting to spending. The data it touches, the ticket, the customer tier, recent orders, policy snippets, prior history, is logged as it is read. The decision behind each action is reconstructable: the trace records the policy it matched, its confidence, any contrary evidence, and why a human was or was not asked. Reversibility is graded, drafts and tags can be undone freely, while credits and customer notices need stronger approval because they leave the building. And accountability is split where the authority is split: support operations owns the queue policy, finance owns the credit authority, and the product owner owns how much autonomy the agent has at all. This time there is no second world to compare against, just one agent answering the same seven links for itself:

CHAIN QUESTION	SUPPORT-TRIAGE AGENT ANSWER
Who authorised it?	The support manager, through an SSO credential, scoped to a business-hours window.

CHAIN QUESTION	SUPPORT-TRIAGE AGENT ANSWER
What was it allowed to do?	Classify priority, draft replies, propose account credits up to $50; explicitly excludes legal-threat, safety, fraud and vulnerable-customer cases.
Which tool acted?	The ticket-system write API, used only to tag and draft; credit issuance is a separate tool behind a separate approval.
What data did it touch?	The ticket, customer tier, recent orders, policy snippets and prior history, each logged as it is read.
What decision led to the action?	A trace recording the policy matched, the model's confidence, any contrary evidence, and whether a human was asked.
Was it reversible?	Drafts and tags are undoable freely; credits and customer notices are gated behind stronger approval because they leave the building.
Who is accountable?	Split where the authority is split: support operations owns queue policy, finance owns credit authority, the product owner owns the autonomy level.

Two engineers handed that blank table would fill it the same way for their own internal agent, which is the point: the chain is a reproducible instrument, not a commerce-only trick. I keep that example boring on purpose. Most identity failures will not begin with a headline shopping case. They will begin with an ordinary internal action whose authority chain was never written down, and surface months later as a dispute nobody can resolve because nobody can reconstruct who allowed it.

Risk-tier the chain, or everyone routes around it

An engineer reading this is entitled to push back, and the pushback is a fair one. If every agent action has to carry credentials and mandates, lightweight agent use becomes unbearable. Not every calendar edit needs a payment-grade protocol. Not every draft email needs a courtroom chain of custody. If every click becomes a notarised ceremony, users will simply route around the system, and a control that everyone evades protects no one. That objection is right, and the answer is that the chain has to be risk-tiered, so the evidence each link carries scales with what the action can do.

A low-stakes, reversible action, retitling a draft, tagging a ticket, adding a tentative calendar hold, can carry the lightest chain: ordinary authentication, a short log entry, and a clear undo. Nothing more is warranted, because nothing worse than an easy correction can result. A medi-

um-stakes action, one that touches shared state or other people's work, needs scoped tool permission granted in advance, a trace retained long enough to investigate later, and a named owner who answers for it. A high-stakes or cross-organisation action, money leaving the company, data crossing a boundary, anything irreversible, needs the full chain: a signed mandate or an explicit human approval, platform-side acceptance from the receiving system, and a defined dispute route for when it still goes wrong. The tier is set by a single question asked at tool-registration time, not at runtime: what is the worst irreversible outcome this tool can produce in one call? A tool that can only write a reversible draft is low; one that can move money or cross an organisation boundary is high; and a tool that can do both must be split so each capability carries its own tier. That makes the sorting reproducible rather than a judgement call, and it is the same discipline that puts credit issuance behind its own approval in the support example above. The standard is not maximum ceremony everywhere. A reversible action earns a log and an undo; an irreversible one earns a mandate and a dispute route. What matters is that the architecture knows which tier an action belongs to before it runs, not after.

But the objection fails the moment it treats friction as the only cost. Missing authority is also friction. It simply arrives later, and in a worse form: as chargebacks, platform blocks, regulatory letters, lost customers and disputes nobody can win. The real choice is not between bureaucracy and speed. It is between visible authorisation up front and invisible authorisation debt that lands after the action has already gone out into the world.

If you are building, ask two authorisation questions, not one

If you are building agents, the single most important change is to ask two authorisation questions for every consequential external action, not one.

The first is the familiar one: what authority did the user grant the agent? That is the delegation problem, and it belongs in your product surface, your identity provider, your policy engine and your audit trail. The second is the leg most architectures are missing: what authority did the receiving platform grant this agent, or this class of agents? That belongs in platform terms, protocol integrations, merchant APIs, access tokens, bot policies, and a machine-readable acceptance or rejection. If the receiving system has not authorised the agent, the user's permission may not save you [2].

This is a job for IAM, identity and access management, the discipline that already governs which humans and which services in your systems may do what. For a reader new to the term: IAM is the system of record that says which principals (users, services, agents) hold which credentials, what scopes those credentials carry, when access is granted and revoked, and who approves changes; it is the layer that turns "Alice is logged in" into "Alice, acting through tool T, with scope S, until time E, on behalf of mandate M."

Agent IAM has to do two things older IAM did not. It has to name more principals than a human-and-service model expects: at least the human, the agent itself, any tool or sub-agent the agent calls, the receiving platform (often called the merchant of record when money is moving,

the party that owns the commercial relationship with the customer and bears the regulatory and dispute burden), and the affected person or account. And at every link in that chain it then has to record five things: the credential that travelled, the scope it carried, the log that captured it, the revocation path that could stop it, and the named owner accountable for it. A user login alone does not do that. A chat transcript does not do that. A line in a terms-of-service document certainly does not.

A reviewer signing off on an agent deployment has to decide which parts of this argument they can lean on now and which they have to keep watching. Two pieces will hold under a hostile cross-examination, and they are the ones worth building against. The user-versus-platform split is not merely theoretical, because the preliminary-injunction order states the distinction directly and the appeal is live [2][3]; and the standards field keeps converging on the same handful of ideas (mandates, merchant-of-record control, delegated authentication, public coordination) because AP2 [4], ACP [5], FIDO [6] and NIST [1] each stake out a piece. The same reviewer should treat the rest as still moving, and budget for it to change: how the appeal resolves, whether the injunction's reasoning survives intact, which of the four bodies of material outlasts the standards settling, and whether a single cross-domain authorisation profile consolidates fast enough to help teams shipping now. The one fact the reviewer cannot read off any vendor's page is whether a particular product's "agent mode" carries genuine platform-side authorisation; only the receiving platform can confirm that, so the review has to go and ask it. None of this weakens the chain; it is precisely the list a responsible review exists to work through.

So take one consequential action your agent performs today, or might perform soon. What credential travels with that action, and who, on the receiving end, can verify it? If the honest answer is "nothing" and "no one", you have found your next piece of work.

When an agent acts, responsibility has to follow the authority chain. User permission, platform authorisation, tool identity, data access, decision record, reversibility and accountable owner are seven separate links. Lose one, and the system may keep working while the responsibility story quietly becomes fiction. The bank, in the scene this essay opened on, would refund that charge eventually, but only once someone could answer the seven questions on its behalf. The box on the doorstep was never the problem. The empty third box on the form was.

> **Carry This Forward**
>
> Identity tells you who acted; it does not, by itself, prove the action complied with your policy, your contracts, the law or your internal controls. The chain has to become runtime governance, evidence the system produces continuously as it works. The next essay puts that system in front of a regulator who wants to know exactly what one agent did to one customer in one hour, and asks whether it can answer. Compliance, it turns out, is not a PDF you generate at audit time.

Status, current as of May 2026

Three procedural deltas affect any team wiring against the sources here. AP2 was donated to the FIDO Alliance in late April 2026 (released as v0.2), so governance and the standardisation work now sit with FIDO; the protocol's location and the eventual standard may continue to move. The *Amazon v. Perplexity* injunction is currently stayed pending the Ninth Circuit appeal at docket No. 26-1444, which makes the preliminary order persuasive reading, not operative law. And FIDO's agentic authentication and payments workstreams, announced in April 2026, have produced no final specification yet, so anyone citing FIDO for agent identity today is citing a workstream, not a standard.

Three regulatory lenses US · EU · UK

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

US Can you show who authorised what, through which tool, under which mandate, with what reversal path, given California AB 316 (effective 1 January 2026) has eliminated the autonomous-AI defence?

EU Can you map the authorisation chain to data protection, consumer, AI Act, platform and product-liability obligations?

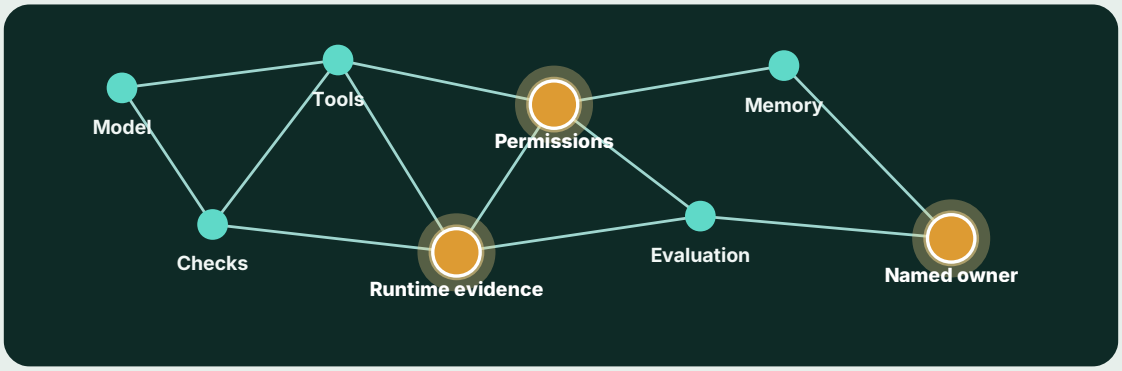
UK Can you read the chain through contract, consumer protection, financial-services, data-protection and sector-regulator expectations?

THE STACK SO FAR

E18 · Essay 18 of 22 complete · Arc IV: Proof and accountability

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** ESSAY 4 OF 5 | V Operating model



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 Agent Identity Chain

You can now trace who acted through authority, tool, data, decision, reversibility and owner.

Next. E19 asks how compliance becomes runtime evidence rather than a PDF.

← PREVIOUS
E17 · The Three Witnesses to a Run

Essay 18 of 22 complete

NEXT →
E19 · Compliance Is Not a PDF

References

Reference links for sources cited in this essay.

1. **AI Agent Standards Initiative**
   NIST
   https://www.nist.gov/news-events/news/2026/02/announcing-ai-agent-standards-initiative-interoperable-and-secure

2. **Amazon v. Perplexity preliminary injunction order**
   U.S. District Court, N.D. Cal.
   https://cases.justia.com/federal/district-courts/california/candce/3%3A2025cv09514/459191/81/0.pdf

3. **Ninth Circuit docket No. 26-1444**
   Justia Dockets
   https://dockets.justia.com/docket/circuit-courts/ca9/26-1444

4. **Agent Payments Protocol specification**
   AP2
   https://ap2-protocol.org/ap2/specification/

5. **Agentic Commerce Protocol specification**
   ACP GitHub
   https://github.com/agentic-commerce-protocol/agentic-commerce-protocol

6. **FIDO Agentic Authentication announcement**
   FIDO Alliance
   https://fidoalliance.org/fido-alliance-to-develop-standards-for-trusted-ai-agent-interactions/

7. **PIPEDA Fair Information Principles: Principle 1 (Accountability), clause 4.1 (incl. 4.1.3 third-party processing)**
   Office of the Privacy Commissioner of Canada
   https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/p_principle/

8. **Act respecting the protection of personal information in the private sector (P-39.1), s. 12.1 (Automated Decision-Making)**
   National Assembly of Quebec
   https://www.legisquebec.gouv.qc.ca/en/document/cs/P-39.1

9. **OWASP Top 10 for Large Language Model Applications 2025: LLM06:2025 Excessive Agency**
   OWASP Foundation
   https://genai.owasp.org/llm-top-10/

10 **California AB 316**
California Legislature
https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=202520260AB316

11 **Regulation (EU) 2024/1689 (EU AI Act), Article 12 (record-keeping/logging) and Article 14 (human oversight) for high-risk AI systems**
European Parliament and Council of the European Union
https://eur-lex.europa.eu/eli/reg/2024/1689/oj

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.

Architecting the AI Coworker · Essay 18, "When an Agent Acts, Who Acted?". Code-first figures, evidence-tiered references.

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