A regulator asks what your agent did between two and three. The vendor sends a thumbs-up.

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

Swipe ››

— THE BUYER RISK

A policy PDF describes what a system was supposed to do. An agent retrieves, reasons, calls tools, writes memory and revises its plan, so governance has to be a recording the system produces as it works, not a description filed at audit time.

— THE REFRAME

What a regulator can reconstruct from the trace.

THE OLD QUESTION

Does the binder describe an intended control?

↓ ↓ ↓

THE QUESTION THAT HOLDS UP

Can the system reconstruct what actually happened?

— WHAT TO ASK FOR

Twelve primitives

For high-risk systems, the EU AI Act requires automatic event recording, human oversight, log retention and meaningful explanations. California AB 316 (effective 1 Jan 2026) blocks the 'the AI autonomously caused the harm' defence.

SOURCE

EU AI Act (Regulation 2024/1689) and California AB 316, effective 1 January 2026.

CHECKLIST LOGIC

Keep three kinds of evidence separate.

01 Read the **reasoning trace** as model narration: clue, not proof.

02 Capture the **system audit trail**: tool calls, approvals, versions, data flows.

03 Layer the **external check**: what a verifier or auditor concludes on top.

— THE ARTIFACT

Twelve runtime-compliance primitives, four buckets.

Registries
systems, tools, data, owners

Authority & policy
mandates, purpose, risk tier

Runtime records
reads, writes, tools, memory

Reversal & audit
remedy, trace store, export

The join (Primitive 12): an audit-ready trace store that links the other eleven into one account a regulator can read. The four buckets, each answering one plain question. Registries: what is running (agent registry, tool registry, versioning record). Authority and policy: what it was allowed to do (permission and mandate registry, purpose and policy record, risk-tiering record). Runtime records: what it actually did. Reversal and audit: whether any of it can be undone and proven. The seven runtime and reversal primitives sit in the supporting detail.

— ASK THIS ON MONDAY

Pick one hour this week, before anyone asks. For one named customer and one named agent, produce the data reads, tool calls, approvals, memory writes, model version, policy version, owner and external outputs. Name the primitive that does not yet exist.

VENDOR TRAP

Treating the SIEM dump (raw security log export) as the audit trail. Raw logs are not a joined record. Build the evidence join (the linkage across systems), name the owner, and rehearse producing one hour's record before the regulator does.

— USE THE CHECKLIST

Compliance Is Not a PDF

Read the full essay — the argument, the sources, the figures and a reader-ready working artifact.
Substack petermccannstrain.substack.com · Medium @peter.mccann.strain · LinkedIn peter-strain-dphil-15a607128
New essays twice weekly, 2 June – 21 July 2026.

Next: E20 — You Cannot Benchmark a Coworker

— THE STACK SO FAR

E19 · Essay 19 of 22 complete · Arc IV: Proof and accountability

YOU JUST ADDED

The twelve runtime compliance primitives

STACK LAYER LIT UP

Runtime evidence / Named owner

YOU CAN NOW ASK

convert compliance into runtime evidence.

NEXT

E20 asks how to evaluate an agent for promotion, not as a leaderboard entry.

— THE ARTIFACT, CONTINUED

Twelve runtime-compliance primitives, four buckets.

THE REMAINING NODES

Runtime records, what it actually did:

- Real-time data-flow inventory: every read, write, export and onward recipient.

- Action ledger: timestamp, trace ID, tool, inputs, outputs, policy check, result.

- Human approval log: what each approver saw, accept/reject/override, reason, time.

- Evaluation records: test suites, failure classes, regression history, release gate.

Reversal and audit, whether it can be undone and proven:

- Rollback, revocation and memory-forgetting record: reversal receipts plus the verification probe.

- Audit-ready trace store (Primitive 12): the single joined, queryable store, the join named above.

Dr Peter McCann Strain

CTO, DPhil/PhD in AI from Oxford University

I build production AI systems and write about making agentic AI useful, inspectable, governable and safe enough for real work.

Follow on Substack for the full 22-essay series
petermccannstrain.substack.com