The Regulator Asked Who Decided. The Answer Cannot Be "The Model".

A regulated financial institution must be able to demonstrate that its controls work. Not assert it — demonstrate it, with evidence, to three independent lines of defence and ultimately to a prudential supervisor. That work is done today, in most banks, with spreadsheets, email chains and the institutional memory of whoever has been there longest.

We were engaged as engineering partner to build an assurance intelligence platform for a prudential regime that requires exactly this demonstration. The platform coordinates assurance across the three lines of defence, scores the quality of evidence before anyone is allowed to rely on it, and surfaces governance intelligence to the Chief Risk Officer and the Board. AI does substantial work inside it. **No AI decision takes effect anywhere in the system without a named human confirming it first, and that confirmation is written before the change executes, not after.**

The regime is Australian. The engineering problem is not. Every US bank deploying AI into a governance process faces the same question from its own model risk function, its internal audit team and eventually its examiner: when the model recommended something and a person acted on it, who decided, on what basis, and can you prove it? This document describes how we built a system in which that question always has an answer.

The client is not named, the institution is not named, and the metrics are not published. The engagement is bound by non-disclosure and the platform is a third party's product. What transfers is the engineering, and the engineering is what follows.

WHAT THE PLATFORM DOES

01

Three lines of defence orchestration

Control testing, risk and compliance, and internal audit coordinated as one system rather than three disconnected functions exchanging spreadsheets and hoping the versions match.

02

Evidence quality scoring

Evidence is scored before anyone may rely on it. Below the floor, the reliance action is not merely discouraged — it is unavailable in the interface.

03

Human-in-the-loop as an invariant

An immutable confirmation record is written before any state change executes. If the write fails, the state never changes. Accountability enforced by ordering, not by policy.

04

Explainable AI reasoning chains

Every agent decision carries its reasoning, its confidence, and the sources it drew on. A reviewer can reconstruct why, months later, without asking the team.

05

Tamper-evident audit trail

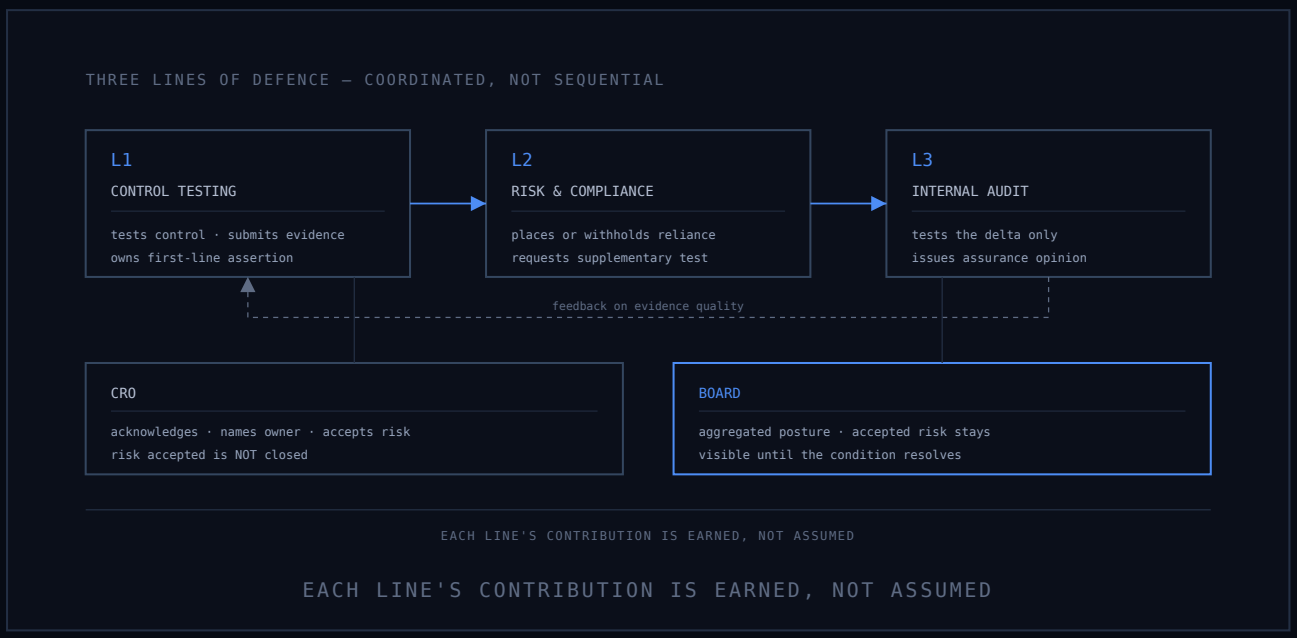
Append-only storage with a cryptographic hash chain. Evidence of what happened that does not depend on trusting the people who operate the system.

+

Grounded retrieval, not general knowledge

A knowledge graph of the regulatory obligations plus retrieval over the institution's own documents. The model answers from the bank's reality, not from the internet's.

THE ASSURANCE MODEL



EACH LINE'S CONTRIBUTION IS EARNED, NOT ASSUMED

WHY THIS IS HARD

The obvious build is a workflow tool: tasks, statuses, a dashboard. That product already exists many times over and it does not solve the problem, because **the problem is not tracking the work. The problem is establishing whether the work can be relied upon.**

Consider what actually goes wrong in a three lines of defence model. The first line tests a control and says it passed. The second line reads that assertion and, reasonably, relies on it. The third line reads the second line's conclusion and relies on that. Three lines of defence have now become one line of defence wearing three hats, and nobody involved did anything wrong. Each of them made a locally sensible decision to trust the line below.

This is the failure mode a prudential regulator is looking for, and it is invisible in a workflow tool, because a workflow tool records that each step completed. It has no concept of whether the step should have been trusted. The status is green at every stage. The assurance is worthless.

Adding AI makes this worse before it makes it better. An agent that recommends reliance is a fourth party in the chain, and it is the one nobody can question afterwards. If the recommendation is followed and the control later fails, the honest answer to "who decided" becomes "the model suggested it and the analyst clicked accept" — which is not an answer any regulator, or any board, will find acceptable.

THE FAILURE

Three lines collapse into one. Each line relies on the line below. Every status is green. No independent assurance exists anywhere in the chain.

THE ANSWER

Reliance must be earned against a quality floor and an independence test, and revoked automatically the moment its basis changes.

THE ARTIFACT

A confirmation record naming the human, the decision, the reasoning and the moment — written before the state changed, not reconstructed after.

WORKSTREAM 01

Evidence Quality Scoring

The single most important design decision in the platform: evidence is scored, and the score gates what the second line is permitted to do. Not advises. Gates.



THE PLATFORM CANNOT BE TALKED INTO RELIANCE IT HAS NOT EARNED

WHAT WE ACTUALLY DID

Every piece of evidence submitted by the first line is scored across independent dimensions — how recent it is, how complete it is against the requirement, and critically whether its source is independent of the person asserting the control works. The dimensions are scored separately and surfaced separately, because an aggregate score hides the thing a reviewer most needs to see: `which` dimension failed.

The total score is compared against a floor. Above the floor, the second line may place reliance. Below it, the reliance action is disabled in the interface and the analyst is offered supplementary testing or escalation instead. The score does not produce a warning that a busy analyst can click past. It removes the option.

THE CHALLENGES WE ENCOUNTERED

- **A single composite score destroys the information.** An aggregate that averages timeliness, completeness and independence produces a number that looks actionable and tells the reviewer nothing about what to fix. Worse, it allows a catastrophic failure in one dimension to be masked by strong performance in the others.
- **Source independence is categorically different from the other dimensions.** If the person who performed the control also produced the evidence that it worked, no amount of timeliness or completeness redeems it. It is not a low score. It is a disqualification.
- **Scores drift as evidence ages.** A control assessed as adequate in March is not adequate in June on the strength of the same evidence, but no human is going to re-examine it unprompted.
- **The model is confident when it should not be.** An LLM scoring evidence will produce a plausible score for material it has fundamentally misread, and it will present that score with the same fluency as a correct one.

HOW WE RESOLVED THEM

- Sub-scores are computed, stored and displayed independently. The reviewer sees which dimension failed and by how much, and the interface leads with the failing dimension rather than the aggregate.
- Independence is implemented as a prohibition rather than a weighting. When it is violated, the reliance action is not rendered at all — not greyed, not warned against, absent — and escalation to the third line is forced. There is no override, because an override would defeat the control it exists to enforce.
- Freshness thresholds are evaluated continuously rather than at submission. Evidence that ages past its threshold degrades its own score and re-enters the queue without anyone remembering to look at it.
- Every score carries an explicit confidence value, and low-confidence scores are surfaced as such rather than presented with the same authority as high-confidence ones. The model is permitted to be uncertain, and the interface is required to say so.

HOW WE PROVED IT WORKED

We tested the gate rather than the score. Evidence sets were constructed to fail each dimension individually and in combination, and the platform was checked to confirm that the reliance action was genuinely unavailable in every failing case rather than merely discouraged. The independence prohibition was tested by attempting to place reliance through every available route — interface, direct interaction with the API, and state manipulation — to confirm that the prohibition lived in the domain logic and not in the presentation layer. **A control that only exists in the interface is not a control.**

RISK AVOIDED

Reliance placed on evidence produced by the person whose work it attests to — the precise circular assurance that the three lines of defence model exists to prevent, arrived at by a chain of individually reasonable decisions.

WHAT A US BANK SHOULD TAKE FROM THIS

Ask where your controls are enforced. If the answer is that a warning appears and a user is expected to heed it, you have documented an intention rather than implemented a control. The difference becomes apparent during an examination, which is the worst moment to discover it.

WORKSTREAM 02

Reliance and Automatic Suspension

Reliance is the mechanism by which three lines of defence become genuinely three, or silently become one. Getting the state machine right is the difference between an assurance platform and an expensive status board.

WHAT WE ACTUALLY DID

Reliance is modelled as an explicit state with explicit transitions, attached to a specific version of the evidence rather than to the control in the abstract. The second line places reliance on a particular evidence package, not on a control's general reputation. This distinction sounds academic and is the entire mechanism.

Because reliance is bound to an evidence version, the platform knows when its basis has changed. New evidence submitted against a control that already carries reliance suspends that reliance automatically and returns the control to the review queue. Nobody has to notice. Nobody has to remember.

THE CHALLENGES WE ENCOUNTERED

- **Not every change should suspend reliance.** A corrected typo in a filename is not new evidence. Suspending on every write produces alert fatigue and trains the second line to dismiss suspensions without reading them — which is worse than not suspending at all.
- **Withheld is not the same as suspended.** A deliberate decision to withhold reliance carries entirely different governance meaning from an automatic suspension pending re-review, and conflating them destroys the reporting.
- **Planned assurance is not assurance.** The third line adding a control to its audit plan tells you about intent, not about coverage. A model that treats a plan as a contribution systematically overstates how well the institution is assured.
- **Accepted risk is not resolved risk.** When the CRO accepts a risk, the natural engineering instinct is to treat the item as closed and remove it from the view. That instinct is precisely wrong.

HOW WE RESOLVED THEM

- Suspension triggers on material change to the evidence package rather than on any write. Materiality is defined explicitly in the domain rules rather than inferred, so the behaviour is predictable and can be explained to an auditor.
- Suspension and withholding are separate states with separate transitions. Withholding requires a written justification of substantive length — long enough to constitute a reason rather than a formality.
- The third line's states distinguish rigorously between trajectory and contribution. Planned and in-progress states are visible in the interface but contribute nothing to assurance coverage. Only an issued opinion counts, and the coverage calculation reflects that.
- Accepted risk is a visible terminal state, not a closure. It remains on the Board's view until the underlying condition is genuinely resolved. The CRO can accept a risk; the CRO cannot make it disappear.

HOW WE PROVED IT WORKED

The state machine was tested transition by transition, including every transition that should be impossible. We attempted to reach reliance from states that must not permit it, to close a governance signal that must not close, and to have planned assurance contribute to coverage. Each attempt was required to fail in the domain logic. Suspension timing was verified under load to confirm that a control returns to the queue promptly enough to matter operationally, because a suspension that arrives after the period closes is a log entry rather than a control.

RISK AVOIDED

Reliance surviving the disappearance of its own justification — the second line resting on a conclusion drawn from evidence that has since been replaced, with the interface showing green throughout.

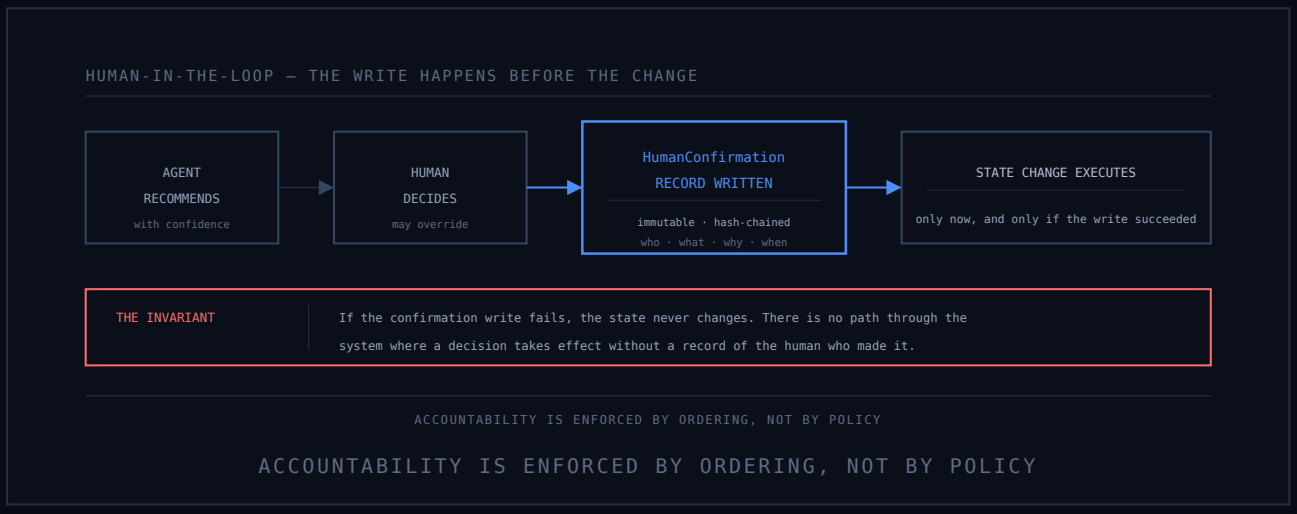
WHAT A US BANK SHOULD TAKE FROM THIS

Ask what happens in your environment when evidence underlying a completed assessment is replaced. If the answer depends on somebody noticing, that is not a process. That is a hope with a workflow tool attached.

WORKSTREAM 03

Human-in-the-Loop as an Enforced Invariant

Every vendor selling AI into a regulated process claims a human in the loop. Almost none of them can tell you what happens when the human's confirmation fails to record. That question is the whole of the design.



ACCOUNTABILITY IS ENFORCED BY ORDERING, NOT BY POLICY

WHAT WE ACTUALLY DID

The confirmation record is written before the state change executes. Not concurrently, not asynchronously, not in a log the application appends to afterwards on a best-effort basis. The write happens first, and the state change is conditional on it succeeding. If the confirmation cannot be recorded, the decision does not take effect.

This inverts the ordinary engineering instinct, which is to perform the valuable work and log it afterwards. Under that ordering, the log is a description of what happened. Under ours, the record is a precondition of it happening. The difference only becomes visible during a failure, which is exactly when it matters and exactly when it will be examined.

THE CHALLENGES WE ENCOUNTERED

- **The natural ordering is the wrong ordering.** Every framework, every tutorial and every engineer's instinct produces act-then-log. Under that ordering a logging failure yields a state change with no record of who authorised it — the exact artefact a regulator is looking for and the exact one that cannot be reconstructed afterwards.
- **Overrides are the most important events and the easiest to lose.** When a human disagrees with the agent, that disagreement is the most governance-relevant thing the system will record all quarter. It is also, in a naive implementation, just another status update.
- **Confirmation fatigue is a real failure mode.** A system that demands confirmation for everything trains users to confirm reflexively, at which point the human in the loop is a rubber stamp with a database record attached — arguably worse than no human at all, because it manufactures evidence of oversight that did not occur.
- **A tamper-evident trail must survive its own operators.** An audit log that a sufficiently privileged administrator can quietly edit is not evidence. It is a document.

HOW WE RESOLVED THEM

- The ordering is enforced in the domain layer rather than left to convention, so that no future engineer can accidentally reintroduce act-then-log by writing perfectly ordinary code. The invariant does not depend on anyone remembering it.
- Overrides are first-class records. When a human departs from the agent's recommendation, the system captures the recommendation, the decision, the divergence and the stated reason, and surfaces that divergence to the next line as a signal in its own right rather than burying it in history.
- Confirmation is reserved for decisions that genuinely change the institution's risk position — placing reliance, withholding it, accepting risk, issuing an opinion. Routine progress does not demand it. The gates are rare enough to still mean something when they appear.
- The audit trail is append-only with a cryptographic hash chain, so that any modification breaks the chain detectably. Trust in the record does not rest on trust in the administrator.

HOW WE PROVED IT WORKED

We proved the invariant by breaking it deliberately. Confirmation writes were made to fail — through induced database errors and through failure injection at the persistence layer — and the system was required to leave the state entirely unchanged in every case. A partial application, where the decision took effect and the record did not, was treated as a critical defect rather than a race condition to be tuned. The hash chain was verified by attempting modification at the storage layer and confirming that verification failed loudly rather than silently.

RISK AVOIDED

A state change with no attributable author. The regulator asks who decided, the system shows that something was decided, and the institution cannot name the person — not because anyone acted improperly, but because a log write failed on a Tuesday and nobody noticed.

WHAT A US BANK SHOULD TAKE FROM THIS

Ask your AI vendor what happens when the confirmation write fails. If the answer is that the action proceeds and the failure is logged, then your human-in-the-loop control is a user interface convention, and it will be discovered as such at the least convenient possible moment.

WORKSTREAM 04

Explainable AI and Grounded Retrieval

An AI that recommends a regulatory decision and cannot explain itself is not an asset. It is a liability with a subscription cost, and it will be identified as one by the first reviewer who asks a hard question.

WHAT WE ACTUALLY DID

Every agent decision persists its reasoning chain, its confidence, and the specific sources it drew upon. Not a post-hoc summary generated when someone asks — the actual reasoning, captured at the moment of the decision, retrievable months later by a reviewer who was not there and has no reason to take the delivery team's word for anything.

The model is grounded twice over. A knowledge graph holds the structured relationships between regulatory obligations, controls, evidence requirements and the institution's own operating model. Retrieval over the institution's uploaded policies and procedures supplies the unstructured context. Every prompt carries both. The model answers from the bank's reality rather than from its training data's general impression of what banks do.

THE CHALLENGES WE ENCOUNTERED

- **A generated explanation is not the reasoning.** Asking a model to explain a decision it already made produces a plausible narrative constructed after the fact. It is a story about the decision, not a record of it, and it will be persuasive precisely when it is wrong.
- **Retrieval without a graph loses the structure.** Regulatory obligations are a web of relationships — this control satisfies that obligation, which requires this evidence type, which the institution assigns to that business unit. Flat similarity search over documents cannot represent that, and it will confidently return a semantically similar but structurally irrelevant passage.
- **A graph without retrieval loses the institution.** The graph knows the regulation. It does not know that this particular bank's disaster recovery procedure says something unusual on page forty.
- **Confidence is not accuracy.** A model's stated confidence correlates with fluency more reliably than with correctness, and a well-designed interface will make a confident wrong answer look exactly like a confident right one.

HOW WE RESOLVED THEM

- Reasoning is captured as the decision is made rather than reconstructed on demand, and it is stored immutably alongside the decision. What the reviewer reads is what the system used.
- The knowledge graph and the retrieval layer are used together on every inference. The graph supplies structure and constrains the retrieval to what is actually relevant to the obligation at hand; retrieval supplies the institution's own language and specifics.
- Every response cites its sources explicitly, and the citations are inspectable rather than decorative. A reviewer can follow a claim back to the paragraph in the bank's own policy that produced it.
- Confidence is displayed but never load-bearing. No gate, no state transition and no prohibition depends on the model's self-assessment. Confidence informs the human; it does not act.

HOW WE PROVED IT WORKED

Reasoning chains were tested for reconstruction rather than for plausibility: a reviewer with no involvement in the original decision was given the stored chain and asked to determine whether the decision was sound. Where they could not, the chain was insufficient regardless of how articulate it read. Grounding was tested adversarially by asking questions whose correct answers existed only in the institution's own documents and contradicted the general regulatory position — a model answering from training data rather than from the retrieved context gets these confidently wrong, which makes them the only questions worth asking.

RISK AVOIDED

An articulate, confident, entirely unfounded recommendation entering a governance process and being acted upon because it read well — and being indistinguishable, at review, from the recommendations that were sound.

WHAT A US BANK SHOULD TAKE FROM THIS

Ask whether the explanation was captured or generated. If it is produced when you ask for it, you are reading a story about a decision rather than a record of one, and the distinction is the entire basis on which your model risk function will either accept or reject the system.

WORKSTREAM 05

Governance Signals and Board Visibility

The purpose of the entire apparatus is that the right thing reaches the Board at the right time, in a form the Board can act on. Everything below this layer is plumbing in service of that.

WHAT WE ACTUALLY DID

Conditions that matter escalate as governance signals with their own explicit lifecycle: raised, acknowledged by a named executive, assigned to an owner with a due date, and eventually either genuinely resolved or formally accepted as risk. The lifecycle is modelled in the domain rather than implied by a status field, because the difference between acknowledged and accepted is the difference between two entirely distinct governance postures.

THE CHALLENGES WE ENCOUNTERED

- **Everything looks important from below.** A system that escalates every anomaly produces a Board pack nobody reads, and the one signal that mattered arrives indistinguishable from forty that did not.
- **Aggregation hides the thing that matters.** A portfolio-level assurance percentage is exactly the number an executive wants and exactly the number that conceals a single catastrophic control failure inside a comfortable average.
- **Accepted risk tends to disappear.** Once accepted, an item has been dealt with in every practical sense, and every instinct in the interface design pushes it off the screen.
- **The Board sees the output and inherits the assumptions.** A dashboard is a claim about the institution's position. If the basis of that claim is not visible, the Board is being asked to approve a conclusion whose derivation it cannot inspect.

HOW WE RESOLVED THEM

- Escalation is driven by explicit, configurable thresholds tied to the institution's stated risk appetite rather than by a model's judgement about salience. What reaches the Board is what the institution decided should reach the Board, and that decision is itself inspectable.
- Aggregate figures are drillable to the specific control, the specific evidence and the specific decision beneath them. No number appears on a governance surface without a path to its basis.
- Accepted risk remains visible until the underlying condition resolves. It is a terminal state, not a closure, and it stays on the Board's view precisely because it is comfortable to forget.
- Every governance surface carries the reasoning and the sources behind its figures, so that the Board can interrogate the claim rather than only receive it.

HOW WE PROVED IT WORKED

We tested the path rather than the pixels. A control failure was introduced at the first line and traced through scoring, reliance suspension, second-line escalation, third-line scope inclusion and the resulting governance signal, to confirm that it surfaced to the CRO with its reasoning intact and its basis inspectable at every step. We also tested the inverse, which is the test most systems fail: a routine anomaly was introduced and confirmed `not` to reach the Board. **A signal that always fires is not a signal.**

RISK AVOIDED

A Board approving a risk position on the strength of an aggregate that concealed the one control failure that mattered — and doing so with a complete, unbroken record of its own diligence.

WHAT A US BANK SHOULD TAKE FROM THIS

Ask whether your Board can drill from any number on its pack to the evidence beneath it. If the answer requires an analyst and a week, then your Board is receiving conclusions rather than information, and its oversight is only as good as the pipeline it cannot see.

WHAT A US BANK SHOULD TAKE FROM THIS

The prudential regime here is Australian, and no US institution is subject to it. That is not the point. **The point is that the machinery a US bank now needs for AI governance is the machinery this platform already implements** — and it was built to satisfy a regulator, not a procurement checklist.

YOUR REQUIREMENT	THE MECHANISM THAT ANSWERS IT	MAPS TO
Model risk validation	Captured reasoning chains, explicit confidence, cited sources, and immutable decision records — a model whose outputs can be independently reviewed rather than only observed.	SR 11-7
Human accountability for AI	Confirmation written before state change. No decision takes effect without a named human, enforced in the domain rather than the interface.	SR 11-7 / BOARD
Third-party risk management	Evidence quality scoring and reliance gating applied to vendor assurance rather than accepted on attestation.	INTERAGENCY TPRM 2023
Operational resilience	Control-to-obligation mapping with continuous evidence freshness evaluation, so coverage reflects the present rather than the last review.	OCC OP RESILIENCE
Audit defensibility	Append-only hash-chained trail. Evidence of what happened that does not depend on trusting the operators of the system.	INTERNAL AUDIT
Independent assurance	Reliance earned against a quality floor and an independence prohibition, suspended automatically when its basis changes.	THREE LINES

Built for a regulator

Not for a procurement checklist. The distinction shows up in the details: prohibitions that cannot be overridden, confirmations that gate rather than log, aggregates that drill to evidence, and accepted risk that refuses to disappear. The engagement is bound by non-disclosure and the platform is a third party's product, so neither is named here. The engineering is ours, and the engineering is what we are offering to walk your team through.

THE QUESTIONS WORTH ASKING ANY AI GOVERNANCE VENDOR

- What happens when the human confirmation write fails? If the action still proceeds, your human-in-the-loop is a user interface convention.
- Is the explanation captured at decision time or generated when I ask for it? Only one of those is a record.
- Where is the prohibition enforced — in the interface, or in the domain? Try it through the API and find out.
- What happens to a completed assessment when the evidence beneath it is replaced? If the answer depends on someone noticing, it is not a control.
- Does an accepted risk leave the Board's view? If it does, who decided that the Board had seen enough of it?
- Can I drill from any figure on the Board pack to the evidence that produced it, without an analyst and a week?

WHERE THIS GOES NEXT

Every bank deploying AI into a governance process will eventually be asked to prove that a human was genuinely accountable for what the system decided. Most will discover at that moment that their human-in-the-loop was a screen rather than an invariant.

We have built the version that holds. We would rather show you how it works than tell you that it does.

TECH4BIZ SOLUTIONS | CODENUK | AUDITTRAX LABS

tech4bizsolutions.com codenuk.com auditraxlabs.com

contact@tech4biz.io