Strategic Report
Edition 1.0

Governance that proves itself.

A CATEGORY-DEFINING REPORT

The Agent Boundary

Accountability, not capability, is the bottleneck to enterprise agent deployment. How a new layer of infrastructure turns four unanswerable questions into signed, offline-verifiable facts.

An industry report by **ByteVerity** — Trusted AI Governance
For CIOs · CISOs · CTOs · boards · regulators · enterprise architects · risk officers

`Identify · Control · Govern · Locate · Prove · Revoke · Posture — one signed recording`

Contents

EXECUTIVE SUMMARY

Accountability is the new bottleneck — and it is manufacturable

> *Enterprises already have capable agents. What they lack is a trustworthy way to answer who acted, what data it touched, under what rule, and where it failed.*

Capability has commoditised; the constraint on enterprise agent deployment has moved to **accountability**. When an agent acts, most enterprises cannot produce a trustworthy account of what it did — their answers live in logs that are mutable, unsigned and believable only to those who already trust them. The instant the question turns adversarial — a breach, a dispute, a regulator — that trust is exactly what is gone.

This report defines the category that closes the gap: **agent accountability infrastructure**, the Agent Boundary — a runtime control point at the agent's edge where three obligations run with no change to the agent's code:

Identify — bind a signed identity to every action, one that survives revocation, offline.
Control — govern what data crosses, field-level and purpose-bound, before it leaves.
Govern — bind every action to the rule it ran under and make drift measurable.

These converge into **one signed recording** — deterministic, signed, tamper-evident, offline-verifiable, and verifiable even with the producing engine gone. That single artifact pays off in two directions: **Verify** (audit, compliance, liability transfer — the outward, security payoff) and **Locate** (root-cause and MTTR — the inward, engineering payoff). One recording, two budgets, one chain of custody.

The report establishes the inevitability of the category before naming any product. A reference implementation exists today — ByteVerity, organised as a family of point products — Identity, the Data Gateway, Cleanroom, Browser Runtime, Control and the governors, Bisect, Lineage, the Capability Gateway and Atlas, on a shared Kernel substrate — whose distinguishing claims — offline fail-closed revocation, engine-absent verification, first-bad-step localisation — are independently re-verifiable, which is the entire spirit of the category. Every figure in the economic model is explicitly illustrative and re-derivable with the buyer's own data.

THE ONE-LINE THESIS

The enterprises that learn to manufacture agent accountability — signed facts an outsider can verify with your systems switched off — will deploy agents the others cannot.

The Problem

Why capable agents stall before production — and why the bottleneck is accountability, not capability.

The Agent Accountability Crisis

> *Enterprises are deploying agents faster than they can prove what those agents did — and regulators, auditors and boards have started to ask.*

Across the Fortune 500, autonomous and semi-autonomous agents have crossed from demo to production. They draft, decide, transact, retrieve and act on live systems. The capability is no longer in doubt. What is in doubt is something more primitive and more dangerous: when an agent acts, the enterprise frequently cannot produce a *trustworthy* account of what happened.

Three forces now converge. Agents are moving into revenue and risk paths, so the blast radius of a wrong action is material. Regulation is arriving with explicit duties to keep records, preserve traceability and evidence human oversight. And the first agent incidents — a leaked field, a wrongful refund, a silent policy drift after a model upgrade — are reaching audit committees. The question being asked in the boardroom is not "can the agent do the work?" It is "can you prove what it did?"

This paper argues that the answer requires a new layer of enterprise infrastructure, that the layer is becoming inevitable, and that it can be built today. We define the category first; only later do we point to a reference implementation as proof that it is real.

EXHIBIT 1

The accountability question is now a board-level question

ERA	DOMINANT QUESTION	OWNER
2021–2023	Can the model do the task?	Data science
2023–2024	Can we deploy it safely enough to try?	Platform eng
2025 onward	Can we prove what it did, to an outsider?	CISO / board / regulator

> **"When an agent acts and no one can prove what it did, the enterprise has not deployed a capability — it has accepted an unbounded, unobservable liability.**

EXECUTIVE CALLOUT

The bottleneck has moved. Capability is procured; accountability is not yet manufacturable. The enterprises that solve accountability first will deploy agents the others cannot.

Four lenses

CIO

Your agent roadmap is gated less by model quality than by your ability to answer for agent behaviour to people who do not trust your dashboards.

CISO

You are being asked to underwrite agent risk with logs that any insider could edit. That is not a control; it is a narrative.

AUDITOR / RISK

The control you will be asked for is record-keeping and traceability that survives independent challenge — not a screenshot.

DEVELOPER / SRE

Every incident review starts with reconstructing what the agent actually saw and did. Today that reconstruction is archaeology.

ADOPTION IMPLICATION Treat agent accountability as a deployment prerequisite, not a post-incident clean-up. It determines how many agents you can safely put into production this year.

Why Capability Is Not The Constraint

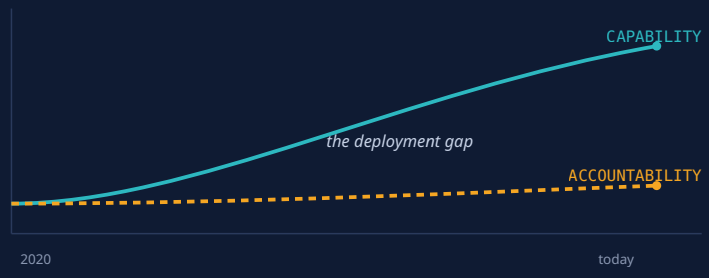
> *The models got good enough; the accountability never showed up — and the gap between the two is exactly where deployments stall.*

For three years the industry optimised one curve: capability. Context windows grew, tool use matured, agentic frameworks proliferated. That curve has flattened as a *blocker* — for most enterprise tasks, the agent is already good enough to be useful. Meanwhile a second curve barely moved: the enterprise's ability to hold an agent accountable after the fact.

The result is a widening scissor. The pilot succeeds; the production rollout stalls in risk review. Not because the agent cannot do the work, but because no one can sign their name under what it will have done. We call this the **Capability–Accountability Scissors**, and it is the structural reason agent programs plateau at "impressive pilot."

EXHIBIT 2

The Capability–Accountability Scissors



Framework F1. Capability rose steeply and commoditised; accountability stayed flat. The opening gap is where governance, audit and risk teams halt the rollout.

> **Capability is now a commodity input. Accountability is the scarce, defensible asset — and it is the one no model vendor ships.**

EXECUTIVE CALLOUT

You cannot close the scissors by buying more capability. The marginal model upgrade does not produce a single additional provable fact about what the agent did.

Four lenses

CIO

More capable models will not unstick the stalled rollouts; the constraint is on the accountability axis, and that is where to invest.

CISO

Each capability increase widens the gap you must cover, because a more capable agent touches more systems with less supervision.

AUDITOR / RISK

The pilot-to-production cliff is an auditability cliff. Close it and the cliff disappears.

DEVELOPER / SRE

Shipping faster agents without accountability just means faster, harder-to-diagnose incidents.

ADOPTION IMPLICATION Re-baseline your agent investment thesis: spend on the accountability axis, where the constraint actually binds, not only on capability.

The Four Questions Every Enterprise Must Answer

> *Who acted, what it touched, under what rule, where it failed — today no enterprise can answer all four with proof, and most cannot answer any.*

Strip agent governance to its irreducible core and four questions remain. They are the questions a regulator, an auditor, an insurer and an incident commander all ask, in different words, about the same event.

The four questions are not aspirational; they are the minimum an enterprise must answer to operate an agent in a regulated or high-stakes context. The trouble is that today they are answered, if at all, with self-asserted logs. We therefore propose a maturity model that grades not whether you *have* an answer, but whether an outsider can *trust* it without trusting you.

EXHIBIT 3A

The Four Questions of Agent Accountability

#	QUESTION	WHAT A TRUSTWORTHY ANSWER REQUIRES
1	**WHO acted?**	A signed identity bound to every action — that survives revocation.
2	**WHAT data did it touch?**	A field-level record of what crossed the boundary, bound to purpose.
3	**UNDER WHAT RULE?**	Each action bound to the exact policy it ran under, effective-vs-intended.
4	**WHERE did it fail?**	The first bad step / version / input, reproducible on replay.

Framework F2. The four questions organise the entire category; every later chapter answers one of them.

> **"An answer you can only defend by saying 'trust our logs' is not an answer a regulator, an insurer or a plaintiff will accept.**

EXECUTIVE CALLOUT

Maturity is not how much you log. It is whether a stranger can verify your answer with the system that produced it switched off. Most enterprises sit at Level 0–1; regulation is pricing in Level 3–4.

EXHIBIT 3B

The Agent Accountability Maturity Model

LEVEL	POSTURE	WHAT "THE ANSWER" IS	THIRD-PARTY PROVABLE?
L0	**Trust**	Plain logs; truth depends on trusting the operator and the vendor.	No
L1	**Centralized**	Aggregated observability; correlated traces, still mutable and self-asserted.	No
L2	**Attested**	Signed records inside the vendor's system; verifiable only while the engine is up.	Partial
L3	**Portable**	Signed, deterministic records that re-verify offline with an open verifier.	Yes
L4	**Accountable**	Engine-absent, fail-closed proof: identity, rule and disclosure re-derivable from the artifact alone, even after revocation.	Yes, vendor absent

Framework F3. A five-level ladder from ‘trust our logs’ (L0) to engine-absent, fail-closed proof (L4). Regulation and insurers are beginning to price in L3–L4 for high-stakes agents.

Four lenses

CIO

Place each agent program on the maturity ladder. The gap between your level and Level 3 is your real agent-governance backlog.

CISO

Level 2 — signed but only while our engine runs — fails the exact test that matters in litigation and breach response.

AUDITOR / RISK

This ladder is a self-assessment you can run today and re-run each quarter; it is the spine of an agent-controls program.

DEVELOPER / SRE

Level 4 is the only level where post-incident reconstruction stops being a matter of opinion.

ADOPTION IMPLICATION Adopt the maturity model as a board-reportable metric. Set a target level per agent tier and track movement, the way you track patch latency.

Why Logging Is Not Evidence

> *A log is a story the system tells about itself; evidence is a fact anyone can re-check without trusting the storyteller — or the vendor.*

Enterprises object: "we already have observability." Observability answers an operational question — what is happening now — and answers it for people who already trust the system. Accountability answers a different question — what happened, provably — for people who do not. The two are not the same control, and one cannot be substituted for the other.

A log is mutable, unsigned and self-asserted. It is believable exactly to the degree you trust the operator who produced it and the vendor who stores it. The moment the question becomes adversarial — a breach, a dispute, a regulator, a subpoena — that trust is precisely what is unavailable. What survives an adversarial question is **evidence**: a fact that re-checks independently. We define the standard. And the standard holds against an active adversary, not merely a careless one: a record *forged with a valid signing key* — an insider fabricating blame, or an operator exonerating themselves — still fails verification, because attribution is pinned to an anchored signer rather than asserted inside the record. Forged blame never renders 'proven'.

EXHIBIT 4

The Evidence Standard — five properties that separate evidence from a log

PROPERTY	A LOG	EVIDENCE
Deterministic	Re-running may differ	Byte-identical on replay
Signed	Unsigned, editable	Cryptographically signed
Tamper-evident	Silent edits	Any change is detectable
Offline-verifiable	Needs the live system	Re-checks with no network
Engine-absent	Trust the producer	Verifies with the producer gone

Framework F4. The fifth property — engine-absent verification — is the one no observability tool offers and the one that matters in an adversarial setting.

> **"Observability is for the people who trust you. Evidence is for everyone who does not — and in an incident, those are the only people who matter.**

EXECUTIVE CALLOUT

The test is simple and brutal: can someone who distrusts you, using software you did not write, with your system switched off, confirm what your agent did? If not, you have logs, not evidence.

Four lenses

CIO

You are not replacing observability; you are adding the layer that holds up when trust is withdrawn. Budget it separately.

CISO

'We have logs' is the sentence that ends careers in a breach post-mortem. Evidence is the sentence that ends the post-mortem.

AUDITOR / RISK

This five-property test is a clean acceptance criterion for any agent-accountability control you evaluate.

DEVELOPER / SRE

Determinism is not bureaucracy; it is what makes an incident reproducible instead of a debate.

ADOPTION IMPLICATION Make the five-property Evidence Standard the acceptance test for every agent-governance tool you buy or build. Anything that fails 'engine-absent' is observability, priced as accountability.

II

The Category

The control point forming at the agent’s edge, and the three obligations that run there.

The Emergence Of The Agent Boundary

> *A new control point is forming between the agent and everything it touches — where every action becomes a signed fact your stack can trust.*

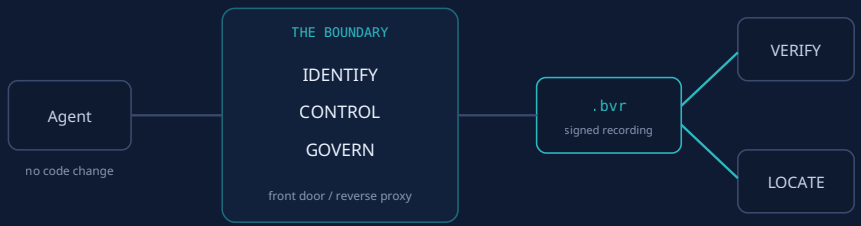
Every prior platform shift created a new control point: the firewall at the network edge, IAM at the identity edge, the API gateway at the service edge. Agents create a new edge — the boundary between an agent's intent and the data, tools and systems it reaches. That boundary is where accountability must be manufactured, because it is the one place every agent action necessarily crosses.

We name the category **agent accountability infrastructure** — in shorthand, the Agent Boundary. Its definition: *a runtime control point that captures every agent action, binds it to a signed identity and the rule it ran under, and emits a signed, offline-verifiable record of what data crossed and what happened.* Three obligations run at this boundary — Identify, Control, Govern — and converge into one artifact that pays off in two directions.

Such a layer exists today. We refer to one reference implementation, the **Agent Boundary Platform**, throughout the rest of this paper — not as the point of the argument, but as proof that the category is buildable now. Its distinguishing claims are independently re-verifiable, which is the entire spirit of the category. ByteVerity — the reference implementation referenced throughout — organises the boundary into a family of point products on a shared proof substrate: **Identity** (who acts); the **Data Gateway**, **Cleanroom** and **Browser Runtime** (what an agent may see, may not see, and what staff may paste); **Control** and the governor family (what is allowed); **Bisect** (where it broke); **Lineage** (the proof); the **Capability Gateway** (revocation); and **Atlas** (posture) — each emitting the same signed record, so cryptographic proof is produced as a byproduct of operation, not bolted on after.

EXHIBIT 5

The Agent Boundary Model



Framework F5. Any agent, unchanged, is captured at a front door; Identify / Control / Govern converge into one signed recording; the recording pays off outward (Verify) and inward (Locate).

> **"The firewall made the network accountable. IAM made access accountable. The Agent Boundary makes the agent accountable — and nothing else is positioned to.**

EXECUTIVE CALLOUT

Categories are defined by control points, not features. The Agent Boundary is the control point at the agent's edge — the natural home of accountability the way the firewall was the natural home of network control.

Four lenses

CIO

This is an infrastructure layer, not a tool — budget, own and standardise it the way you did IAM.

CISO

A single control point at the boundary is far more defensible than instrumenting every agent framework you will ever adopt.

AUDITOR / RISK

One boundary means one evidence pipeline, regardless of which model or framework engineering chooses next quarter.

DEVELOPER / SRE

No code change is the adoption unlock: the boundary captures agents you did not write and cannot modify.

ADOPTION IMPLICATION Designate the Agent Boundary as an architectural standard now, before each team instruments its own agents in incompatible ways.

Runtime Identity

> *Every action carries a signed identity that re-verifies even after the agent is revoked — offline, no network, fail-closed.*

The first obligation is **Identify**: bind a signed identity — a Principal — to every action the agent takes, at the moment it takes it. The boundary either issues that identity from a built-in registry or accepts it from your existing OIDC / Okta- or Entra-compatible IdP, validating at the door and stamping it into the record. Issue, rotate and revoke run at fleet scale.

The differentiating test is revocation. In a credential system that matters, a record made *after* an agent is revoked must fail to verify — and must fail even with no network call at all, from cached signed state, fail-closed. *[Reference implementation: shipped and hand-verified.]* This is the property that separates real identity governance from a login screen, and it is the one a breach responder will reach for first.

EXHIBIT 6

Identity that survives revocation — offline

MOMENT	ACTION	OFFLINE VERDICT
T0	Agent registered → signed Principal + short-TTL credential	Valid
T1	Agent records a governed run	Verifies — not revoked
T2	Agent revoked	status list updated, signed
T3	Same agent records again	Fails — "revoked as of T2", no network

The verdict at T3 is reached with the issuing system absent — the headline property of runtime identity.

> **"An identity you can only verify while the vendor's servers are up is an identity you cannot use in the one moment you need it.**

EXECUTIVE CALLOUT

‘Who acted?’ is only answered if the answer holds after you have fired the agent. Identity that evaporates the moment the engine is offline is not identity governance.

Four lenses

CIO

Reuse your existing IdP — the boundary consumes Okta/Entra rather than replacing them, so there is no identity migration.

CISO

Offline, fail-closed revocation is the primitive that makes agent identity defensible in incident response.

AUDITOR / RISK

Each action carries a verifiable attribution; ‘who acted’ stops being a matter of correlation and becomes a matter of signature.

DEVELOPER / SRE

Short-TTL credentials bound at record time mean a leaked token cannot be replayed into a valid-looking history.

ADOPTION IMPLICATION Wire the boundary to your IdP and make signed, revocable agent identity a precondition for any agent reaching production data.

Runtime Data Control

> *See and constrain what data crosses the boundary — before it leaves, not after the incident report.*

The second obligation is **Control**: govern what data crosses the boundary, at the field level, bound to the requesting identity and its declared purpose. The boundary can run in three escalating postures — **observe-first** (flag and sign boundary crossings without blocking), **deny-before-release** (refuse a disclosure that exceeds policy before any data leaves), and full enforcement. Each decision produces a signed receipt of what was released, to whom, and for what purpose — and that receipt cannot misreport which safeguard fired: hide a transform, downgrade a matched class, or re-seal what was released, and re-derivation from the signed record fails the verification. The recording proves the guardrail, not merely its own existence. In the reference implementation, Control spans three point products — the **Data Gateway** (what an agent may see: field-level data disclosure and context-boundary enforcement), **Cleanroom** (what it may not see: secrets and sensitive source sealed to a harmless stub), and the **Browser Runtime** (what staff may paste into AI tools, governed at the moment of send). *[Reference implementation: field-level disclosure, observe and deny-before-release, and typed prompt-injection scanning.]*

This is the answer to “what data did it touch?” — and it is answered prospectively, before exfiltration, rather than reconstructed afterward from logs that may already be compromised.

EXHIBIT 7

Three postures, one signed receipt

POSTURE	BEHAVIOUR	WHEN TO USE
Observe-first	Flag & sign off-policy crossings; do not block	Rollout & baselining; build evidence before enforcing
Deny-before-release	Refuse a disclosure that exceeds policy, before data leaves	Sensitive fields, regulated data
Enforce	Transform / withhold per policy; sign the decision	Production, high-stakes agents

Posture is per-agent and reversible; every outcome is a signed receipt, so even observe-first produces admissible evidence.

> **The cheapest data-loss incident is the one the boundary refused before a single field crossed the line.**

EXECUTIVE CALLOUT

‘Deny-before-release’ is the difference between a control and a chronicle. A log tells you the data left; a boundary stops it leaving and signs the fact that it did.

Four lenses

CIO

Observe-first lets you deploy the control without blocking the business on day one, then tighten with evidence in hand.

CISO

Field-level, purpose-bound disclosure is DLP semantics applied at the agent edge — where your existing DLP cannot see.

AUDITOR / RISK

Each release carries a signed record of what data, to whom, for what purpose — the substance of a records-of-processing obligation.

DEVELOPER / SRE

Purpose binding means the same agent gets different data for different declared tasks, enforced, not by convention.

ADOPTION IMPLICATION Start every sensitive agent in observe-first to accumulate signed evidence, then graduate to deny-before-release once the policy is proven against real traffic.

Runtime Governance

> *Every action is bound to the exact rule it ran under — and effective-vs-intended is provable per agent, not asserted in a policy document.*

The third obligation is **Govern**: bind every action to the specific policy version it executed under, and make the gap between the policy you *intended* and the policy that was *effective* computable for each agent. A policy document on a wiki is an intention; a signed binding between an action and a policy hash is a fact. In the reference implementation, **Control** and the governor family add temporal-sequence governance over these bindings — cooldowns, prerequisites and approval gates. *[Reference implementation: per-agent policy, the three postures, and effective-vs-intended conformance are shipped.]*

This answers “under what rule?” in a way that withstands the most common governance failure: drift. A model upgrade, a prompt change or a config edit silently moves behaviour away from policy. Because each decision is bound to the exact policy hash it ran under, drift is not a surprise discovered months later — it is a quantity you can measure per agent, continuously.

EXHIBIT 8

Intended vs effective — governance as a measurable quantity

CONSTRUCT	TRADITIONAL GOVERNANCE	BOUNDARY GOVERNANCE
The rule	A document	A signed, content-addressed policy bundle
Binding to action	Assumed	Each decision bound to a policy hash
Drift detection	Periodic review	Effective-vs-intended, computed per agent
Proof	Attestation by staff	Re-verifiable from the record

Governance stops being a quarterly assertion and becomes a continuous, signed measurement.

> **Policy drift is the silent killer of agent governance — and the only defence is binding each action to the rule it actually ran under.**

EXECUTIVE CALLOUT

‘We have a policy’ and ‘every action provably ran under that policy’ are different claims. Only the second one is governance; the first is documentation.

Four lenses

CIO

Conformance becomes a number you can put on a board slide, per agent, instead of a colour on a heat map.

CISO

You can prove that a given decision ran under the approved policy version — the question regulators actually ask.

AUDITOR / RISK

Effective-vs-intended is the audit you have always wanted and never had: did the control actually operate?

DEVELOPER / SRE

A model bump that changes behaviour shows up as conformance drift immediately, not as a customer complaint.

ADOPTION IMPLICATION Make per-agent conformance a release gate: an agent whose effective policy diverges from intended does not ship until the gap is explained.

One Signed Recording

> *One artifact, five properties — and it still verifies with the engine that made it gone.*

Identify, Control and Govern converge into a single artifact: **one signed recording**. It captures the agent's run byte-for-byte and carries the identity, the policy bindings and the disclosure receipts with it. Because the capture is deterministic, the recording can be replayed byte-identically; because it is signed and tamper-evident, any change is detectable; because the verifier is open and engine-absent, anyone can re-check it offline. *[Reference implementation: all five properties shipped, with a dependency-light open verifier; the tamper-evident ledger detects omission, not just alteration.]*

The recording sits on a tamper-evident ledger that catches not only an altered row but a *deleted* one — detection a conventional audit log cannot give. This is what lets a single artifact serve as both courtroom-grade evidence and an engineering-grade reproduction.

EXHIBIT 9

The signed recording — one artifact, five guarantees

GUARANTEE	WHAT IT GIVES THE ENTERPRISE
Deterministic	Replay the run byte-identically — reproduction, not reconstruction
Cryptographically signed	Origin and integrity bound cryptographically
Tamper-evident	Any edit — or deletion — is detectable
Offline-verifiable	Re-checks with no network and no vendor service
Engine-absent	Verifies even with the producing system permanently gone

The artifact is the product of the category. Everything outward (Verify) and inward (Locate) is a read of this one record.

> **The signed recording is the unit of agent accountability — the byte-exact, self-verifying fact that every stakeholder can read in their own language.**

EXECUTIVE CALLOUT

One artifact, two readers. The CISO reads it as evidence; the engineer reads it as a reproduction. They are reading the same signed bytes — which is why one record can settle a dispute and fix a bug.

Four lenses

CIO

One artifact format, owned by you, that two organisations depend on — that is leverage and standardisation in a single object.

CISO

Engine-absent verification means your evidence outlives the vendor, the contract and the outage.

AUDITOR / RISK

Omission detection closes the oldest gap in log-based audit: the record that was quietly removed.

DEVELOPER / SRE

A deterministic recording is a perfect repro — the incident replays exactly, every time.

ADOPTION IMPLICATION Standardise on the signed recording as your agent system-of-record; route both audit and incident workflows to the same artifact.

III

The Payoff

One signed recording, read two ways: outward as evidence, inward as diagnosis.

Verify: Evidence, Audit, Liability

> *Turn 'trust us' into 're-check it yourself' — and map every signed fact to the control framework your auditor already uses.*

The outward payoff is **Verify**. The same recording that the boundary produces is what your auditor, your regulator, your insurer and, in the worst case, opposing counsel re-check — independently, with your systems offline. Audit stops being a quarterly archaeology of log-pulls and narrative and becomes an export of signed records anyone can confirm. Liability becomes transferable, because the chain of custody is cryptographic rather than testimonial.

Crucially, the evidence survives the vendor. If the platform vendor disappears, the records still verify with the open verifier — the property that lets a CISO sign off without betting the enterprise's legal posture on a supplier's longevity.

One handling note follows from the design: because a record *cites* the upstream receipts it composes, it can carry the very data it attests to. The boundary adds no secrets of its own and redacts them on display, but the exported bundle should be classified and shared under the same controls as the data it describes — it is evidence, and evidence is sensitive.

EXHIBIT 10

From log-pull to signed export — the audit collapse

AUDIT TASK	LOG-BASED	EVIDENCE-BASED
Establish what happened	Reconstruct from mutable logs	Replay the signed recording
Establish who	Correlate identities	Read the signed Principal
Establish the rule	Find the policy in effect	Read the bound policy hash
Convince a sceptic	Ask them to trust you	Hand them the open verifier

Chapter 16 maps these records to specific obligation classes (EU AI Act, NIST AI RMF, ISO 42001, GDPR, SOC 2).

> **"Audit is expensive because trust is manual. Make the facts self-verifying and the cost of proving them collapses toward the cost of an export.**

EXECUTIVE CALLOUT

Liability transfer requires a chain of custody. A log is testimony — only as strong as your credibility under challenge. A signed, offline-verifiable record is evidence — as strong as the mathematics, regardless of who is asking.

Four lenses

CIO

Audit-cycle time and external-audit fees are a measurable line you can attack with self-verifying evidence.

CISO

You can hand a regulator the verifier and let them confirm the facts themselves — the strongest possible posture.

AUDITOR / RISK

Sampling gives way to verification: you can check the population, not a sample, because checking is cheap.

DEVELOPER / SRE

The evidence you produce for audit is the same artifact you use to debug — no separate compliance tax.

ADOPTION IMPLICATION Re-engineer the agent audit process around signed-record export and independent verification; retire the manual log-narrative workflow.

Locate: Diagnosis, Root Cause, MTTR

> *The same recording that proves accountability also finds the first step that broke — so safe rollback no longer means re-run-and-hope.*

The inward payoff is **Locate**. Because the recording is deterministic and replayable, the boundary can do for an agent what `git bisect` does for a codebase: given a recording and a pass/fail check, it localises the **first bad step** — or the first bad version, or the minimal input — that produced the wrong outcome, by replaying spliced variants. *[Reference implementation: first-bad-step localisation via deterministic replay is shipped.]*

This is the second budget's payoff. The CISO funds Verify; engineering funds Locate; both are reads of the same artifact. Mean-time-to-resolution on agent incidents — today dominated by reconstructing what the agent saw — collapses when the run is a perfect, replayable reproduction and the bad step is found automatically.

EXHIBIT 11

git bisect, for agents

GIT BISECT	AGENT BOUNDARY LOCATE
Searches commit history	Searches the agent's own axes: step, version, input, policy
Needs a deterministic checkout	Needs a deterministic recording — which the boundary guarantees
Finds the breaking commit	Finds the first bad step / version / input
Output: a SHA	Output: a signed verdict, re-verifiable offline

Framework F6 in action: one recording, two payoffs, two budgets — Verify (security) and Locate (engineering).

> **"Determinism is the bridge: it makes the same artifact admissible to a regulator and reproducible to an engineer.**

EXECUTIVE CALLOUT

The recording you keep for the auditor is the recording that finds the bug for the engineer. You are not paying twice; you are paying once and being served twice.

Four lenses

CIO

One artifact serves two organisations and two budgets — the rare control that pays for itself on the engineering side alone.

CISO

Faster, evidence-grounded incident response is itself a security outcome, not just an engineering one.

AUDITOR / RISK

‘Where did it fail?’ gets a reproducible answer, not a plausible story.

DEVELOPER / SRE

First-bad-step localisation turns an afternoon of bisecting prompts into a single command.

ADOPTION IMPLICATION Route agent incident response through Locate; measure the MTTR delta and attribute the engineering savings to the same artifact security funded.

IV

The Business & Market Case

Economics, competitive coverage, architecture, deployment and regulatory alignment.

The Economics Of Accountability

> *Accountability is not a compliance tax — it transfers liability, compresses MTTR, and lets you put more agents into production safely.*

The economic case rests on a structural fact established in Chapter 11: one artifact is funded by two budgets. Security pays for Verify; engineering pays for Locate. That alone changes the buying calculus, because the control does not have to justify itself on the compliance line alone.

Five value lines drive the return. We present them as a model, not a forecast: the figures below are **illustrative** and exist to show the *shape* of the case. Every number must be re-derived with the buyer's own loaded rates, incident history and the backlog of agents currently stuck in governance review. The largest term is almost always the last one — the agents you cannot deploy today because you cannot account for them.

EXHIBIT 12A

Five drivers of the risk-adjusted accountability case

#	DRIVER	BUDGET LINE
1	Liability transfer — cryptographic chain of custody	Risk / legal
2	Audit & compliance hours collapsed	Security / GRC
3	MTTR / incident & downtime cost	Platform eng
4	Regulatory penalty & standard-of-care exposure avoided	Risk / board
5	Agent-deployment velocity — revenue unlocked from agents stuck in review	Business / P&L

Framework F8. Drivers 1, 2 and 4 are cost-avoidance; 3 is operational; 5 is the growth term and usually the largest.

> **"Frame accountability as a deployment accelerator, not a cost centre: it is the control that lets you say yes to the agents you are currently forced to delay.**

EXECUTIVE CALLOUT

The decisive number is not what accountability saves — it is what it unlocks. Every agent stalled in risk review is value you have already built and cannot ship. Accountability is the key that releases it.

EXHIBIT 12B

Illustrative worked example — mid-size deployment

VALUE LINE	BASIS (ILLUSTRATIVE)	ANNUAL
Audit & evidence assembly Hours of manual log-pull & narrative per audit, replaced by export of signed records	1,200 hrs/yr × $120 loaded	$144,000
One avoided Sev-1 agent incident Mean cost of a single material data-exposure / wrong-action event (illustrative, breach-cost basis)	1 event	$300,000
MTTR on agent regressions Engineer-days per incident saved by first-bad-step localisation	260 days × $900	$234,000
Agents unblocked from risk review Value of agents stuck in governance, released because accountability is provable	8 agents × $150,000	$1,200,000
	Illustrative annual value, mid-size deployment	**≈ $1.88M**

Every figure is **illustrative** and must be re-derived with the buyer's own loaded rates, incident history and agent backlog. Breach-cost basis: public industry studies (e.g., IBM Cost of a Data Breach). The model is the deliverable — not the numbers.

Worked instance of Framework F8. Figures are illustrative placeholders; the arithmetic is the point. A real model is built from the buyer's loaded rates and agent backlog.

Four lenses

CIO

The growth term — agents released from review — typically dwarfs the cost-avoidance terms; lead the business case with it.

CISO

Two budgets share the cost, which is the easiest internal-funding story you will ever run for a security control.

AUDITOR / RISK

Penalty-and-standard-of-care avoidance is real and growing; price it with counsel rather than guessing.

DEVELOPER / SRE

The MTTR line is yours to claim; instrument it before and after to make the savings undeniable.

ADOPTION IMPLICATION Build the business case bottom-up from your own numbers using this model; lead with deployment velocity, not compliance savings.

Competitive Landscape

> *You already run IAM, DLP, observability and a SIEM — none of them produce a signed fact about what the agent did.*

The instinct in any new category is to ask "don't I already own something that does this?" The honest answer is that you own adjacent controls that each cover part of one question and none of which produce signed, offline-verifiable accountability across all four. The Agent Boundary is a **complement** to these controls, not a replacement: it consumes your IdP and feeds your SIEM and GRC.

The coverage map makes the gap explicit. IAM knows who, but not what data or under what rule, and signs nothing portable. DLP sees some data, but not agent identity or policy binding. Observability sees activity, self-asserted and engine-bound. Guardrails screen content, but produce no evidence. The boundary is the only layer that answers all four questions with a signed, independently verifiable record.

EXHIBIT 13

The Control-Stack Coverage Map

CONTROL YOU ALREADY RUN	WHO	WHAT DATA	UNDER WHAT RULE	WHERE IT FAILED	SIGNED & OFFLINE-PROVABLE
IAM / workload identity	○	–	–	–	–
DLP / CASB	–	○	–	–	–
AI observability	○	○	–	○	–
AI guardrails	–	○	○	–	–
SIEM / GRC	○	–	○	–	–
Agent Boundary layer	●	●	●	●	●

● covered & provable ○ partial / self-asserted – out of scope. Incumbents are complements the boundary consumes (IdP) or feeds (SIEM/GRC).

Framework F7. The boundary fills the row no incumbent occupies: signed and offline-provable across who, what, rule and failure. It is positioned as a complement, integrating with the stack you run.

> **Every tool in your stack watches the agent. None of them make the watching provable. That empty cell is the category.**

EXECUTIVE CALLOUT

This is not a displacement play. The boundary makes your existing IAM, DLP and SIEM investments *accountable* — it is the layer that turns their telemetry into evidence.

Four lenses

CIO

No rip-and-replace: the boundary slots in beside your controls and raises the evidentiary value of all of them.

CISO

Map your current coverage against the four questions; the blank column is your exposure and your buying thesis.

AUDITOR / RISK

A single layer that spans all four questions simplifies your control narrative dramatically.

DEVELOPER / SRE

It consumes the IdP you run and exports to the SIEM you run — integration, not migration.

ADOPTION IMPLICATION Run the coverage map against your installed tools; the cells none of them fill define the requirement for an Agent Boundary layer.

Reference Architecture

> *A front door, not a forklift — capture any agent in an afternoon and keep every recording on your own infrastructure.*

Architecturally, the boundary is a reverse-proxy front door. The agent's base URL is pointed at it; no SDK change, no agent rewrite. From there the layered design separates concerns cleanly: a frozen, canonical-bytes **substrate** defines the deterministic vocabulary every record speaks; an **identity** capability issues or validates Principals; a **governance / disclosure** plane makes field-level, purpose-bound decisions and signs receipts; and the **boundary platform** binds it all into the signed recording and the Verify / Locate reads.

The trust posture is deliberate: recordings stay on the customer's infrastructure, the verifier is open and dependency-light, and telemetry is off by default. There is nothing the customer is forced to log into and no place the evidence is held hostage.

EXHIBIT 14

Reference architecture — layered, local-first

Capture	Reverse-proxy front door; point one base-URL env var at it. No SDK change, no agent rewrite.
Identity in	Consumes your existing OIDC / Okta- or Entra-compatible IdP, or a built-in registry for agents without one.
Evidence out	Signed records export to your SIEM and GRC tooling as the audit system-of-record.
Residency	Local-first: recordings stay on your infrastructure. Open, dependency-light verifier. Telemetry off by default.
Lock-in	The artifact format is open and offline-verifiable; you can re-check every record with the vendor gone.

Capture is non-invasive; identity flows in from your IdP; evidence flows out to your SIEM/GRC; everything is re-verifiable with the vendor absent.

> **"The architecture that wins is the one that captures the agents you already run, unchanged, and keeps the evidence on your side of the line.**

EXECUTIVE CALLOUT

‘A front door, not a forklift’ is the adoption test. If accountability requires re-instrumenting every agent, it will not happen. If it requires pointing one URL, it will.

EXHIBIT 14B

The point-product family on the seven-verb spine

VERB	POINT PRODUCT	WHAT IT GOVERNS AT THE BOUNDARY
IDENTIFY	**Identity**	A short-lived, revocable identity bound to every action as it acts — minted in-platform or carried from your IdP (Okta ID-JAG, SPIFFE / SPIRE).
CONTROL	**Data Gateway**	What an agent may **see** — field-level disclosure; redacts secrets, quarantines prompt-injection, releases only what policy allows.
CONTROL	**Cleanroom**	What it may **not** see — secrets and sensitive source sealed to a harmless stub; the real value never reaches the agent.
CONTROL	**Browser Runtime**	What staff **paste** into AI tools — governed at the moment of send; holds no keys of its own.
GOVERN	**Control + governors**	Deny a risky action *before* it happens and seal the decision to the policy it ran under — with release & underwriting editions.
LOCATE	**Bisect**	The exact bad step — or the exact poisoned knowledge write, who made it and when.
PROVE	**Lineage**	One signed record across every plane, composed from the receipts the others emit and re-verified offline by anyone.
REVOKE	**Capability Gateway**	Revoke a capability — or an identity in your IdP — and every action under it fails closed, re-derived offline.
POSTURE	**Atlas**	A searchable, verifiable index over every signed record: what agents knew, did and disclosed, and when.

Identity (WHO) and the signed recording are the cross-cutting substrate every product emits — cryptographic proof as a byproduct of operation, feeding VERIFY and LOCATE. Product scope here is positioning; capabilities are stated at the level of what the platform does today.

Four lenses

CIO

No agent rewrite and no data egress means the architecture clears both the engineering and the procurement bar.

CISO

Local-first, open-verifier, telemetry-off is the posture that survives a vendor-risk and data-residency review.

AUDITOR / RISK

An open verification path means your evidence is not contingent on the vendor's continued existence.

DEVELOPER / SRE

Reverse-proxy capture works with subscription tokens and any OpenAI/ Anthropic-compatible endpoint.

ADOPTION IMPLICATION Pilot with the front-door capture pattern against one existing agent; validate that no code change and no data egress are required before scaling.

Deployment Models

> *From a single developer binary today to a governed fleet — with no lock-in, because the evidence format is open.*

The boundary deploys along a spectrum. At the small end it is a local developer binary: capture one agent, see the run, locate a bad step, verify offline. At the enterprise end it is a fleet control plane: many agents, signed policy distribution, a tamper-evident ledger and posture views. Critically, the artifact is identical across the spectrum, so evidence captured by a developer on day one is the same evidence an auditor reads in year three.

Lock-in is addressed structurally, not contractually. Because the recording format is open and the verifier is independent, the buyer can re-check every record with the vendor gone. That is the strongest possible answer to the build-vs-buy question: you buy the manufacturing of evidence, but you own the evidence and can always verify it yourself.

EXHIBIT 15

Deployment spectrum

STAGE	SHAPE	PRIMARY PAYOFF
Developer	Local binary, one agent	Locate — debug & reproduce
Team	Shared capture, per-agent policy	Govern — conformance per agent
Fleet	Control plane: distribution, ledger, posture	Verify — fleet-scale evidence

The artifact is invariant across stages; you adopt incrementally without re-capturing or re-formatting evidence.

> **"Adopt small, keep the artifact, scale the deployment — the evidence you capture as a developer is admissible as an enterprise.**

EXECUTIVE CALLOUT

Open format plus independent verifier equals no lock-in by construction. You are not trusting a promise not to hold your evidence hostage; you are holding the verifier yourself.

Four lenses

CIO

Incremental adoption with an invariant artifact de-risks the program: no big-bang, no re-platform.

CISO

An open, independently verifiable format is your exit guarantee — vendor risk is bounded.

AUDITOR / RISK

Evidence captured years apart, by different teams, remains comparable because the format is frozen.

DEVELOPER / SRE

Start on your laptop today; the recording you make is the one the enterprise will trust later.

ADOPTION IMPLICATION Begin at the developer tier on a real agent; standardise the artifact now so later fleet adoption is configuration, not migration.

Regulatory Alignment

> *The signed recording supplies the evidence behind the obligation classes regulators are now codifying — record-keeping, traceability, oversight.*

Regulation is converging on a small set of obligation classes for high-stakes AI: keep records, preserve traceability, evidence human oversight, be transparent, monitor post-deployment, and remain accountable for what data is processed. The Agent Boundary does not *satisfy* these obligations — an organisation does — but it manufactures the **evidence** that satisfying them requires, in a form an external party can verify.

The mapping below is by obligation class, deliberately. Article numbers and applicability shift; the underlying duties are stable. **This is not legal advice** — confirm specifics with counsel — but the structural alignment is strong precisely because the category was designed around the same questions regulators ask.

EXHIBIT 16

Obligation-class mapping

OBLIGATION CLASS	FRAMEWORK	WHAT THE SIGNED RECORDING SUPPLIES
Record-keeping & logging	EU AI Act (high-risk record-keeping obligations)	The signed recording is a tamper-evident, time-bound record of each agent action and the rule it ran under.
Traceability	NIST AI RMF (Measure / Manage)	Every decision is bound to a policy hash and a signed identity; effective-vs-intended is re-derivable.
Human oversight & transparency	EU AI Act oversight & transparency duties	Observe-first and deny-before-release postures, plus a readable per-step trace, evidence the override existed.
AI management system	ISO/IEC 42001	Posture tiles and the append-only ledger supply the operational evidence an AIMS audit asks for.
Accountability & records of processing	GDPR accountability principle; records of processing	Field-level disclosure decisions bound to purpose produce a signed record of what data was released and why.
Security evidence	ISO/IEC 27001; SOC 2	Offline-verifiable receipts and omission-detecting ledger serve as control evidence.

Not legal advice. Frameworks are referenced by obligation class; verify current article numbers and applicability with counsel. The platform produces *evidence that supports* these obligations — it is not a compliance guarantee.

The recording is the connective evidence beneath multiple frameworks at once — one artifact, many obligations.

> **When the obligation is 'keep a record that an outsider can trust,' a signed, offline-verifiable recording is not one option — it is the literal answer.**

EXECUTIVE CALLOUT

Regulators are not asking for better dashboards. They are asking for records that survive independent scrutiny. That is the exact artifact the boundary produces — which is why alignment is structural, not retrofitted.

Four lenses

CIO

One evidence pipeline maps to several frameworks at once, reducing the per-regulation compliance build.

CISO

You can demonstrate the control operated, not merely that a policy existed — the distinction examiners probe.

AUDITOR / RISK

Obligation-class mapping is more durable than article-by-article checklists that change with each revision.

DEVELOPER / SRE

Compliance evidence is a by-product of the recording you already make — no separate instrumentation.

ADOPTION IMPLICATION Map your applicable frameworks to obligation classes with counsel, then point each class at the signed-recording evidence rather than building parallel compliance logging.

V

The Future & The Thesis

From developer binary to the governance layer of the agent economy.

The Future: Discovery, Replay, Control Plane

> *From a developer-grade platform today to the fleet control plane and operator surface next.*

This chapter sets out where the platform and the category are heading. The reference implementation today is a capable, developer-and-team-grade platform with an accountability core in place today: signed identity with offline revocation, the signed recording with offline verification, locate, per-agent policy and conformance, a tamper-evident ledger and posture, and live integration with a governing disclosure gateway.

The forward arc extends this core into a fleet product.

EXHIBIT 17

Platform direction — the fleet arc

DIRECTION	WHAT IT ADDS
Fleet distribution endpoint	Pull signed policy & status lists to many agents over a content-addressed channel
Operator Console	A read-mostly, attestable surface — the ten-minute onboarding and live-receipt view
Metered disclosure (GA) & structured parsers	Leakage-budget enforcement and SQL/GraphQL-aware field control
External witnessing	Publish ledger checkpoints out-of-band to close the split-view trust gap

Stated as the platform's forward direction; the core described in earlier chapters stands today.

> **"Maturity is a direction; the platform's forward arc extends a core that is real today.**

EXECUTIVE CALLOUT

Everything in earlier chapters is verifiable today; this chapter sets the platform's forward direction.

Four lenses

CIO

Buy for the core capability today; the roadmap is upside.

CISO

External witnessing is the item to watch — it closes the one trust gap the offline model cannot close alone.

AUDITOR / RISK

A vendor that names what it can prove today, and separates it from direction, is one whose claims you can weight more heavily.

DEVELOPER / SRE

The control plane and console reduce operational toil; until then the core is fully usable via the binary.

ADOPTION IMPLICATION Evaluate on capability today; treat the roadmap as upside.

The Governance Layer Of The Agent Economy

> *In 24 months, 'prove what your agent did' becomes an expectation — and the signed-fact format becomes the connective tissue regulators, insurers, auditors and engineers all read from.*

Step back and the trajectory is clear. Network traffic got a governance layer; identity got one; data got one. The agent economy will get one too, and it will form at the boundary because that is where every agent action crosses. The enterprises that adopt early do not merely reduce risk — they convert agent risk from a deployment blocker into a competitive advantage, deploying at scale while peers remain in review.

The deeper prize is standardisation. As auditors, cyber insurers, regulators and engineering teams all converge on reading the same signed artifact, that artifact becomes the connective tissue of the agent economy — the receipt everyone accepts. Owning the boundary, and the format it emits, is therefore not a point product; it is the governance layer of an entire mode of computing.

EXHIBIT 18

The accountability maturity curve, revisited

LEVEL	POSTURE	WHAT "THE ANSWER" IS	THIRD-PARTY PROVABLE?
L0	**Trust**	Plain logs; truth depends on trusting the operator and the vendor.	No
L1	**Centralized**	Aggregated observability; correlated traces, still mutable and self-asserted.	No
L2	**Attested**	Signed records inside the vendor's system; verifiable only while the engine is up.	Partial
L3	**Portable**	Signed, deterministic records that re-verify offline with an open verifier.	Yes
L4	**Accountable**	Engine-absent, fail-closed proof: identity, rule and disclosure re-derivable from the artifact alone, even after revocation.	Yes, vendor absent

Framework F3, as a destination. The standard of care is migrating toward L3–L4; early movers operate there before it is mandatory and absorb the change as advantage rather than scramble.

> **Whoever defines the signed-fact format for agent accountability defines the language the whole agent economy will use to trust itself.**

EXECUTIVE CALLOUT

The standard of care is not set by vendors; it is set by what regulators, insurers and courts come to expect as normal. That expectation is moving toward signed, offline-verifiable agent records. Early movers meet it as strategy; late movers meet it as a fire drill.

Four lenses

CIO

Being early on the maturity curve is a durable advantage: you deploy agents your competitors cannot yet account for.

CISO

You would rather operate at L4 by choice now than be forced there by an examiner during an incident.

AUDITOR / RISK

A converging evidence standard across regulators and insurers reduces your long-run compliance entropy.

DEVELOPER / SRE

A standard artifact format means tooling, talent and integrations compound instead of fragmenting.

ADOPTION IMPLICATION Make a board-level decision to operate ahead of the standard of care; the cost of being early is small and the cost of being late is set by someone else.

CONCLUSION

Start with one agent

> *Inevitability is an argument; conviction is a demonstration. The category is proven the first time your own people verify a recording with the vendor absent.*

We have argued that accountability is the binding constraint on enterprise agent deployment; that four questions define it; that logs cannot answer them because evidence requires what logs lack; that a new control point — the Agent Boundary — answers all four with one signed recording; and that the same artifact pays off outward as evidence and inward as diagnosis. We have placed the category against the existing stack as a complement, mapped it to the obligations regulators are codifying, and distinguished what the platform does today from where it is heading.

The proof is not in this document. It is in an afternoon: take one high-stakes agent already stuck in review, put it behind the boundary without changing its code, drive a governed run, and hand the signed recording to your own auditor to re-verify — offline, with our systems switched off. If it holds, you have not evaluated a product; you have located the standard of care for the agent economy, and decided to operate ahead of it.

THE FIRST STEP

One agent. One signed recording. Re-verified by your team, vendor absent. Minutes to prove; a category to own.

Methodology & honesty note: capability statements describe a reference implementation, with forward-looking platform direction set out in Chapter 17. Economic figures are illustrative and must be re-derived with the buyer's own data. Regulatory mapping is by obligation class and is not legal advice. The defining claims of the category are designed to be independently re-verifiable with the producing system absent.