AUSPICA

AI Governance Advisory

The Declared Behavior Framework

An AI Governance Audit Methodology

Governing premise

Governance is the assurance that what happens is what was supposed to happen, with no side effects and no unintended consequences.

Version 1.2 | July 2026

PART 0

The premise, and why it produces a better audit

Most AI governance frameworks are organized around *risk categories*; bias, privacy, security, transparency; and produce checklists that ask whether a policy exists. They are useful for demonstrating diligence and poor at detecting whether anything is actually true.

This framework starts somewhere else. It takes governance to mean a single assurance:

> **What happens is what was supposed to happen, with no side effects and no unintended consequences.**

That sentence is not a slogan. It is a specification, and it decomposes into four questions that can be tested against evidence rather than asserted in a policy document; plus a fifth question about whether the answers can be proven to someone who does not trust you.

Layer	The question	What it establishes
1. Declaration	What is it supposed to do?	Whether an intended-behavior specification exists at all, and whether it is specific enough to test against. Without this, nothing below can be assessed.
2. Conformance	Did it do that?	Whether observed behavior matches the declaration: the core comparison.
3. Containment	Did it do anything else?	Side effects: undeclared actions occurring in-band, concurrent with the intended work.
4. Consequence	What followed from it?	Unintended consequences: downstream, delayed, aggregate, and emergent effects that no single action reveals.
5. Evidence	Can you prove it?	Whether the answers above are supported by a complete, tamper-evident record a third party can verify.

Two properties that make this framework work

It is ordered. Each layer depends on the one before it. You cannot test conformance without a declaration; you cannot assess consequence without first establishing containment. This dependency produces the remediation roadmap automatically. An organization weak at Layer 1 should not be sold Layer 4 tooling, and telling them so is what makes the assessment credible.

It is falsifiable. Every layer asks for evidence rather than for policy. “Show me the declaration” and “show me the record of what actually happened” are questions with observable answers. A framework that can fail is a framework worth running.

PART I

Policy and the declaration

Most clients encountering this framework already have an AI policy, and often a substantial investment in one. The correct response is not to position against that work. Policy comes first, and it is the input to everything that follows.

Where policy fits

An organization's AI policy states what *should* be true. This framework establishes whether it *is*. Every dollar spent on policy created an obligation the organization currently cannot discharge, and closing that gap is the entire purpose of the assessment.

The relationship is hierarchical rather than parallel:

- **Policy** states the rule in human language, for human judgment.
- **The declaration** translates the rule into a machine-checkable constraint on a specific system.
- **The layers** test whether the constraint actually held.

Why policy alone cannot govern an agent

Most documents called an AI policy are, on inspection, acceptable-use policies. They govern what *people* may do with AI tools, and they are enforced through training, supervision and accountability. Every one of those enforcement mechanisms acts on a human being.

An autonomous agent does not read the policy, cannot be trained on it, and cannot be held accountable to it.

> **Policy is a human control. An agent can only be constrained by mechanism.**

This is not a criticism of any organization's policy work. It is a category distinction the market has not yet made explicit, and stating it plainly is frequently the most useful thing an assessor does in a first meeting. The line that lands:

"Your AI policy governs your people's use of AI. It does not govern the AI."

The traceability principle

Every declared constraint should trace to a policy clause. Every policy clause that constrains system behavior should produce at least one declared constraint. That bidirectional mapping produces three findings:

- **Unenforceable policy.** A clause with no corresponding declared constraint on any system. The policy asserts it; nothing checks it.
- **Orphan constraint.** A declared limit with no policy basis. Either an undocumented requirement or an unnecessary restriction, and both are worth surfacing.

- **Coverage.** The proportion of policy assertions that have been mechanized.

The policy coverage map

This is the strongest artifact available in a first meeting, because the source material is the client's own document and the finding is arithmetic rather than opinion.

Method

- **1. Decompose.** Break the policy into discrete assertions, one testable claim per row. A clause containing three obligations becomes three rows.
- **2. Classify.** Does the assertion constrain human behavior, system behavior, or both? Only system-behavior assertions can be mechanized. Human ones are correctly enforced by policy and should be marked out of scope, not counted as failures.
- **3. Locate.** For each system-behavior assertion, find the declared constraint that implements it, if one exists.
- **4. Score enforceability.** 0 not declared; 1 declared but never tested; 2 declared and tested manually; 3 declared and continuously compared; 4 declared, compared and evidenced.
- **5. Compute coverage.** System-behavior assertions scoring 3 or above, divided by total system-behavior assertions.

Output format

Policy ref	Assertion	Constrains	Declared where	Enforceability
4.2	PII must not leave controlled systems	System	DECL-0041 sec. E	3 compared
4.5	Staff must complete AI training annually	Human	*out of scope*	n/a
6.1	AI decisions affecting customers require human review	System	not declared	0 gap

The headline the client remembers: **"Of your 34 policy assertions that constrain system behavior, 6 are currently verifiable."** That single sentence reframes the engagement from "you need governance" to "you have already decided what should be true; here is what is provable."

Conversations with policy-mature clients

What they say	What to say
"We already have an AI policy."	"Good, that is the difficult part. Can you show me the check that proves any agent is following it?"
"We completed a governance assessment last year."	"Against what? Did it review documents, or did it test behavior?"
"Legal owns AI governance here."	"They own the policy. Who owns verification?"
"Isn't this just monitoring?"	"Monitoring detects change. Governance detects wrong. The difference is the reference point: history versus intent."

What they say	What to say
“We are implementing NIST AI RMF / ISO 42001.”	“Both require demonstrating the system operates as intended. That is the layer this tests. What is your current evidence?”
“Our vendor handles this.”	“Which of the five layers does their evidence cover, and can you verify it without trusting them?”

Never position against the policy. The person who commissioned it, whether general counsel, chief privacy officer or compliance lead, is frequently the buyer or the champion in the room. Criticizing the policy criticizes them, and the meeting closes. The framing is always “you have decided what should be true; the gap is proving it,” never “your policy does not work.”

When there is no policy

If a prospect has no AI policy, Layer 1 scores zero and the leading finding is that there is nothing to govern against. That is legitimate, but decide the response in advance: either scope a lightweight policy-alignment step into the first week, or partner with a governance and risk firm that does policy work and exchange referrals. The ideal client is not the one with no policy. It is the one with a mature policy and no verification.

PART II

The standards landscape, and what it leaves open

Part I addressed what the client wrote for itself. This part addresses what the client is measured against. An assessment that cannot say precisely what the existing standards do and do not cover will lose the room to any compliance lead who has read them, and the compliance lead has usually read them.

The purpose here is not to disparage the frameworks. Several of them are excellent at what they set out to do, and the absences identified below are gaps of scope rather than gaps of rigor. The point is to locate the boundary of that scope accurately, because the five layers in Part III sit exactly on the other side of it.

What exists

Six families of instrument make up the field as of July 2026. A serious client is likely to hold two or three of them and to be asked about a fourth.

- **Risk management frameworks.** NIST AI RMF 1.0, published January 2023 and currently under revision, is the reference instrument, structured as Govern, Map, Measure and Manage. ISO/IEC 23894 provides parallel guidance in the ISO idiom.
- **Management system standards.** ISO/IEC 42001:2023 is the spine of the field: 38 Annex A controls, certified by accredited bodies on a three-year cycle. ISO/IEC 42005:2025 covers AI system impact assessment and ISO/IEC 42006 sets requirements on the certification bodies themselves.
- **Law and regulation.** The EU AI Act, Regulation 2024/1689, is the horizontal framework, with obligations tiered by risk class and phased over several years. United States state law is currently retrenching rather than expanding.
- **Assurance and attestation.** SOC 2 remains what enterprise buyers ask for. AIUC-1, with 51 requirements and 130 controls across six pillars, is the first auditable certification standard built specifically for AI agents, and it is coupled to insurance capacity.
- **Security threat taxonomies.** The OWASP Top 10 for Agentic Applications, announced in December 2025 and designated ASI01 to ASI10, is the first systematic classification of agentic risk. MITRE ATLAS and NIST AI 100-2 sit alongside it.
- **Agent-specific standards work.** NIST announced its AI Agent Standards Initiative in February 2026 and has indicated an AI Agent Interoperability Profile for the fourth quarter of 2026. The gap is recognized at the level of national standards bodies, which also means the window for a proprietary answer is bounded.

Instrument	What it governs	Where it stops for agents
NIST AI RMF 1.0	Organizational AI risk, via Govern, Map, Measure and Manage.	No categories for delegation chains, sub-agent spawning, or the interval between an action and its observation.

Instrument	What it governs	Where it stops for agents
ISO/IEC 42001:2023	The AI management system: policy, roles, process, continual improvement.	Certifies that you decide responsibly, not that any particular agent behaved.
ISO/IEC 42005 and 42006	Impact assessment; requirements on certification bodies.	Assessment is prospective and periodic. No runtime observation.
EU AI Act	Legal obligations tiered by risk class.	High-risk regime deferred to December 2027 and August 2028. Conformity is documentary.
Colorado SB 26-189	Disclosure around automated decision-making technology.	Duty of care, risk programs and impact assessments removed. Effective January 2027.
SOC 2	Service organization controls against five Trust Services Criteria.	No AI module exists. Auditors interpret the 2017 criteria against machine learning failure modes.
AIUC-1	Agent security, safety, reliability and accountability, certified and insured.	Point-in-time adversarial testing, self-accredited, with agent scope set by the vendor.
OWASP Agentic Top 10	The ten most critical agentic security risks.	A threat catalog. Silent on what a given agent was permitted to do.
NIST Agent Initiative	Voluntary guidelines for autonomous systems.	Not yet published. The shape of the answer is still open.

The four assumptions the field shares

The absences below are not oversights. They follow from four assumptions that almost every instrument above shares, each of which held firmly for machine learning systems and holds poorly for agents.

- **The unit of governance is a system, not an actor.** Frameworks are built around an AI system with a boundary: inputs arrive, a model produces outputs, a deployment context determines consequences. An agent selects its own actions, mutates external state, persists memory across sessions and may delegate to other agents. Governance designed for a bounded system has no natural place to record what an unbounded actor may touch.
- **Accountability is discharged by naming a person.** Every serious instrument requires an accountable owner, and enterprise procurement now asks for one by name. This is sound governance and it is where the chain breaks, for the reason set out in Part I: naming an owner without giving that owner an instrument produces accountability without control.
- **Documentation is evidence.** The artifact that discharges an obligation is a document: a risk assessment, an impact assessment, a model card, a technical file, an attestation report. These are assertions produced by the party being governed. For an agent taking thousands of consequential actions a day, a document describing intent is a weak proxy for a record of behavior.

- **Assurance is a point in time.** Certification runs on a three-year cycle, attestation covers a review period and is issued afterward, impact assessments are performed before deployment, adversarial testing is a battery run on a build. All establish a state at a moment. Agent behavior is not a state; it is a stream.

Taken singly, each assumption is defensible. Taken together they produce a governance program in which a named human is accountable, on the basis of documents they wrote, for the behavior of a non-human actor they cannot observe, assessed at a moment that has already passed. That is not a weak program. It is a program aimed at the wrong object.

The seven absences

Seven capabilities that no instrument above provides. They are ordered by dependency: each is difficult to close until the one before it is closed.

- **No machine-checkable declaration of permitted behavior.** Every framework asks an organization to describe its AI systems, and every description is prose. There is no standard artifact stating, in a form a machine can evaluate, which tools an agent may call, which data it may reach, what requires approval and what it must never do. Note that AIUC-1, the one standard built for agents, does not define what an agent is. The absence begins that early. Layer 1.
- **Conformance is asserted, never measured.** No framework requires an organization to establish continuously that an agent stayed inside its specification, and most provide no mechanism to establish it at all. Adversarial testing answers whether an agent can be induced to misbehave under test. Drift monitoring answers whether outputs shifted statistically. Neither answers whether yesterday's production behavior matched what was approved. Layer 2.
- **Side effects are outside the frame.** Frameworks evaluate whether the task was completed correctly and are largely silent on what else happened along the way. For a model, side effects are close to meaningless. For an actor with tool access they are most of the risk. The OWASP list names excessive agency without supplying a governance instrument for it. Layer 3.
- **Consequence is unowned.** Downstream, delayed, aggregate and emergent effects have no assigned owner in any framework reviewed. OWASP names cascading failures as a top-ten risk; nothing specifies how the question would be evidenced. This is the least mature area in the entire landscape. Layer 4.
- **Evidence is vendor-vouched, not verifiable.** Every evidentiary artifact in the field is produced by, or on behalf of, the party being governed, and rests on the reputation of that party or its auditor. Nothing produces evidence a skeptical third party can verify independently of the producer. The absence of tamper-evidence is invisible until the moment of dispute, which is the only moment it matters. Layer 5.
- **The accountable owner has no instrument.** Procurement asks for the named owner of AI risk and the frameworks require one. None supply the thing that would let that person do

the job: a live view of which agents exist, what each is permitted to do, where behavior has diverged and what was done about it. The named owner attests on the basis of documents produced by the teams they are supposed to be overseeing.

- **Assurance decays between assessments.** Between two assessments an agent can acquire a new tool, be pointed at new data, receive an updated model, or begin interacting with an agent deployed by another team. Each changes the behavior without triggering a governance event.

The first five absences are the five layers of this framework, in order. That correspondence is not a coincidence and it is the strongest available argument for the structure: the layers were derived from asking what the field leaves unanswered, and the last two are answered by the shape of the instrument rather than by any single layer.

Where this leaves AIUC-1

AIUC-1 warrants specific treatment because it is the nearest thing in the market to a competing answer, and a well-read client may raise it. It should be taken seriously rather than dismissed: it was developed with substantial institutional participation, it publishes crosswalks to ISO 42001, MITRE ATLAS, the EU AI Act, NIST AI RMF and the OWASP Top 10, and it updates quarterly rather than annually.

Its constraints are structural rather than incidental. AIUC authors the standard, runs the technical evaluations, issues the certificates, accredits the auditors and sells the insurance that the certificate enables, so the standard rests on its own authority rather than on an external accreditation body. It does not define what counts as an AI agent, which leaves the scope of certification to the vendor. And its instrument is adversarial testing: a large battery of simulations run at a point in time. That establishes what an agent did under test. It does not establish what the agent did on Tuesday. Layer 2 asks the same question continuously, in production, against a declaration rather than a test suite.

A correction on timing

Through 2025 the case for AI governance investment was carried largely by regulatory deadlines. That case has weakened, and any material still resting on it is now inaccurate.

- **The EU high-risk deadline has moved.** The Digital Omnibus on AI, endorsed by the European Parliament on 16 June 2026 and approved by the Council on 29 June 2026, defers stand-alone Annex III high-risk obligations from 2 August 2026 to 2 December 2027, and obligations for AI embedded in regulated products under Annex I to 2 August 2028. Most transparency obligations were left in place. The deferral happened because national authorities and harmonized technical standards were not ready, not because the requirements were withdrawn.
- **Colorado has retrenched further.** SB 26-189, signed on 14 May 2026, moves the effective date to 1 January 2027 and replaces the original risk-based regime with disclosure requirements around automated decision-making technology. The duty of care against

algorithmic discrimination is gone, along with deployer risk management programs and impact assessments.

- **There is no SOC 2 module for AI.** The AICPA has issued none. What exists is AI-fluent auditors interpreting the 2017 Trust Services Criteria against machine learning failure modes. Describing this as SOC 2 AI criteria overstates the position, and it is the kind of imprecision a well-briefed auditor will notice.

The correct response is not to find a replacement deadline. It is to change the argument from compliance timing to operational exposure. Agents are in production and taking consequential actions now. The instruments aimed directly at them are months old, which means no organization is behind and every organization is exposed. And the insurance market has begun to price agent governance directly, which turns evidence into a commercial input rather than a compliance artifact. That argument is true, and unlike a deadline it does not expire.

Part III sets out the five layers that answer these absences, and the crosswalk at the end of this document translates them back into the obligations a client is actually subject to.

PART III

The five layers

Layer 1: Declaration

The question: What is this system supposed to do?

Governance begins with a statement of intended behavior. If no such statement exists, there is nothing to govern against, only behavior to observe and rationalize after the fact. This layer is where most organizations discover they have deployed systems nobody ever specified.

The declaration is where policy becomes testable. Each constraint should trace back to a policy clause, and the coverage map in Part I is the instrument that makes that traceability visible.

What good looks like

- A declaration exists for every deployed agent or AI system, maintained as an artifact rather than as institutional memory.
- It is specific enough to generate a test: named tools and capabilities, data scopes, egress destinations, identity and privilege, resource and rate bounds, and required human approval gates.
- It is versioned, with an owner and an approval record, who authorized this system to do these things, and when.
- It is machine-readable, so comparison can be automated rather than performed by a human reading logs.
- Each constraint traces to the policy clause that requires it.

How to test it

Ask for the declaration. Then apply the **test-generation check**: can you write a specific pass or fail test from this document? “Agents must handle data responsibly” generates no test and is not a declaration. “This agent may call tools A, B and C; may read from store D; may not transmit to any endpoint outside allow list E” generates several.

Failure modes to look for

- **No declaration.** The system was deployed by a team that understood its purpose; nothing was written down.
- **Prose masquerading as specification.** A policy document exists but nothing in it is testable.
- **Stale declaration.** The specification describes version one; the system is on version nine.
- **Circular declaration.** The specification was generated from observed behavior. This is the subtle and serious one: a baseline learned from what the system does cannot detect that the system is doing the wrong thing. It can only detect change. An organization with learned baselines and no declared intent has monitoring, not governance.

> **Auditor's note.** Layer 1 is where the majority of assessments will score lowest, and it is the finding clients find most uncomfortable and most useful. Deliver it plainly: you cannot be told whether your agents behaved correctly, because no one ever recorded what correct was.

Layer 2: Conformance

The question: Did it do what it was supposed to do?

With a declaration in hand, conformance is the direct comparison: for each declared capability, is there evidence the system operated within it? This is the layer people imagine when they say "monitoring", and it is necessary but not sufficient. It tests only the surfaces you already thought to declare.

What good looks like

- Telemetry captures the actions the declaration constrains; tool calls, data reads, egress, identity assertions, resource consumption.
- Comparison against the declaration is automated and continuous, not a periodic manual sample.
- Divergences produce a finding with enough context to act on: what was declared, what was observed, when, by which system, and under whose authority.
- Findings route to an owner and have a defined disposition path. A finding nobody acts on is a log line.

How to test it

Select three to five declared constraints and trace each end to end. Ask to be shown the telemetry that would demonstrate compliance, and, more revealing, ask what would have happened had the constraint been violated. If nobody can describe the detection path, the constraint is aspirational.

Failure modes to look for

- **Telemetry gaps.** Constraints exist over actions that are not instrumented at all.
- **Sampling.** Comparison happens on a subset, so a violation is detected probabilistically. For governance assertions this is usually inadequate.
- **Detection without disposition.** Findings accumulate in a dashboard nobody owns.
- **Alert-threshold drift.** Rules were tuned down to reduce noise until they stopped firing. Ask when each rule last produced a finding: a rule that has never fired is either perfectly obeyed or broken, and the difference matters.

Layer 3: Containment (side effects)

The question: Did anything else happen?

A side effect is an action that occurred as part of the work but was never intended: the agent asked to summarize a document also transmitted it to an external service; the agent granted read access used a credential to write; the agent performing a lookup cached results in an unmanaged store. Side effects are in-band and concurrent: they appear in the same execution trace as the intended work, if anyone is looking at the whole trace.

> **The structural trap in this layer.** You cannot detect undeclared behavior by monitoring only declared surfaces. An organization that instruments exactly the tools its agents are supposed to

use will never observe a call to a tool they are not supposed to use. Containment requires monitoring the whole available surface and treating anything outside the declaration as a finding; the inverse of an allow list tested only against itself.

The five containment surfaces

Surface	The side effect to detect	Evidence to request
Tool authority	Invocation of tools, functions or APIs outside the declared set.	Complete tool-call log, not filtered to the allow list.
Data access	Reads beyond declared scope; retrieval that pulls unintended context.	Query and retrieval logs with the requesting identity attached.
Data egress	Information leaving by undeclared paths, including model outputs, logs, caches and error messages.	Network egress records and output-scanning results.
Identity & privilege	Acting under a credential or on behalf of a principal beyond declared authority.	Credential inventory and per-action identity attribution.
Resource & rate	Consumption, throughput or spend outside declared bounds.	Usage and cost telemetry over time, not in aggregate.

Failure modes to look for

- **Allow list self-confirmation.** Monitoring scoped to the declaration, as above.
- **Output blindness.** Inputs are inspected; model outputs are not scanned for credentials, personal data or internal context.
- **Unattributed actions.** Actions are logged without which agent, which principal, or under what authority, making containment untestable even where data exists.

Layer 4: Consequence (unintended consequences)

The question: What followed from it?

An unintended consequence differs from a side effect in time and place. The side effect is in the trace; the consequence is downstream of it: later, in another system, in aggregate, or emerging from interaction between systems. Almost no organization assesses this layer, which makes it the most differentiating part of the audit and the section clients circulate internally.

The five consequence classes

Class	Example	How it is detected
Temporal	An action creates a cached artifact, queued job or scheduled task that acts later, outside the original approval.	Correlate created artifacts and jobs back to the originating action.
Cross-system	A permitted write triggers automation in a downstream system that was never part of the declaration.	Blast-radius mapping: what does each declared action set in motion?
Aggregate	Individually valid actions become harmful in volume: cost runaway, rate-induced outage, mass notification.	Rate and budget analysis over windows, not per-action checks.
Emergent	Two agents, each conformant alone, produce a loop or escalation neither declared.	Multi-agent interaction review; look for agents that consume each other's output.
Human	A person makes a consequential decision on agent output, believing it was reviewed when the approval was a formality.	Approval-chain integrity: sample approvals and test whether review actually occurred.

How to test it

Take one high-authority agent and perform a blast-radius walk: for each declared action, ask what that action causes, then repeat the question on the answer until you reach a system nobody in the room owns. That endpoint is your finding. Most organizations reach it in two or three steps.

For the human class, sample approval records and check the interval between presentation and approval. Approvals granted in seconds, in batches, or by an account that approves everything are rubber-stamps; the control exists on paper and not in practice. This is one of the most reliably productive tests in the entire audit.

Failure modes to look for

- **No blast-radius model.** Nobody has mapped what agent actions set in motion downstream.
- **Per-action-only controls.** Every individual action is checked; nothing evaluates the aggregate.
- **Unreviewed multi-agent topology.** Agents call other agents and no one has drawn the graph.
- **Ceremonial human oversight.** A required approval gate that is never withheld.

Layer 5: Evidence

The question: Can you prove it to someone who does not trust you?

The first four layers establish what is true. This one establishes whether it can be demonstrated: to an auditor, a regulator, a customer's security review, or a court. An organization may be genuinely well governed and still fail this layer, and in a dispute the distinction between being right and being able to show it is the whole matter.

The five evidence tests

- **Existence.** Is there a record of the actions, the declarations, and the approvals?
- **Completeness.** Can gaps be detected? A record with silent gaps cannot support an assertion that nothing happened during them.
- **Integrity.** Would alteration be detectable? Mutable application logs held by the team being audited are weak evidence.
- **Independence.** Can a third party verify the record without trusting the system that produced it, or the vendor that operates it?
- **Durability.** Does the record outlive the incident, the employee, the vendor contract, and the retention window of the regulation that applies?

Failure modes to look for

- **Mutable logs.** Records that an administrator can edit without trace.
- **Retention shorter than obligation.** Ninety-day log retention against a multi-year audit requirement.
- **Vendor-dependent proof.** Evidence that can only be produced while a subscription is active.
- **No chain of custody.** Evidence can be assembled but not shown to be the same evidence later.

PART IV

The maturity model

Each layer is scored 0 to 4 on the same scale. The scale is deliberately about *mechanism*, not effort. It asks what the organization can demonstrate, not how hard it is trying.

Level	Name	Definition
0	**Absent**	The capability does not exist. No declaration, no telemetry, no record.
1	**Asserted**	It exists as prose or intention. Nothing testable, nothing enforced. Most policy documents live here.
2	**Structured**	Specific and documented in a form that could be tested, but comparison is manual, periodic or sampled.
3	**Observed**	Automated and continuous. Divergences are detected and produce owned findings.
4	**Assured**	Continuous, with signed tamper-evident evidence and a defined enforcement response. Independently verifiable.

Scoring rules

- **Score against evidence, not intent.** If it cannot be demonstrated during the engagement, it does not score.
- **A layer cannot exceed the layer above it by more than one.** Conformance at 4 with Declaration at 1 is not possible: you cannot continuously verify against a specification that does not exist. Where the interview suggests otherwise, the higher claim is almost always monitoring mislabeled as governance.
- **Score per system, then roll up.** Organizational averages hide the one ungoverned high-authority agent that constitutes the actual risk. Report the minimum alongside the mean.

Present the result as a grid: systems on one axis, the five layers on the other, colored by level. This single page is what circulates to the board, and it is the reason the engagement gets talked about internally.

PART V

The audit methodology

A four-week engagement. The sequence matters: inventory before declaration, declaration before testing, testing before scoring. Compressing the front end is the most common way these assessments go wrong. An audit of systems you have not finished discovering produces a confident report about the wrong scope.

Week	Activity	Output
1	Scope and inventory. Identify every deployed AI system and agent, its owner, its authority, and its data reach. Obtain the current AI policy.	The agent register, frequently the first complete list the client has ever had, and often the single most valued artifact.
2	Policy decomposition and declaration review. Build the coverage map. Gather specifications, telemetry samples, approval records and log configurations.	The policy coverage map; Layer 1 findings; the evidence inventory; identified gaps in what can be tested at all.
3	Testing. Trace selected constraints end to end. Run the containment surfaces, the blast-radius walk, and the approval-interval sample.	Layer 2 to 4 findings with supporting evidence references.
4	Scoring, roadmap and readout. Score the grid, sequence remediation, prepare and deliver the executive presentation.	The leave-behind package and a live executive readout.

Prioritizing findings

Rank by **authority multiplied by exposure**: how much a system is permitted to do, multiplied by how far its reach extends. A low-authority agent with a broad blast radius and a high-authority agent in a sealed environment are both real risks; an agent with high authority and wide reach and no declaration is the finding that leads the report.

Sequence remediation by layer dependency, not by severity alone. Recommending continuous monitoring to an organization at Declaration level 0 sells them a system that will compare observed behavior against nothing. Fix the declaration first, and say so even though the larger engagement is downstream. That restraint is the thing that earns the second engagement.

PART VI

The leave-behind

The leave-behind is the engagement's real product. A report that is read once and filed generates no referral; a set of artifacts the client keeps using generates both renewal and reputation. Build it so that half of it remains in daily use after you have gone.

Package contents

Artifact	Audience	Purpose
1. Executive summary	Board / exec	One page. The maturity grid, the three findings that matter, and the recommended sequence. Written so a non-technical director can act on it.
2. Agent register	Ongoing	The living inventory: every AI system, its owner, authority, data reach and declaration status. Handed over as a maintainable template.
3. Policy coverage map	Compliance	Every policy assertion, whether it constrains humans or systems, where it is declared, and its enforceability score. The artifact that connects existing policy investment to verification.
4. Layer findings	Technical	Findings by layer, each with observed evidence, why it matters, and a specific remediation. No finding without evidence.
5. Maturity scoring	Board / exec	The grid, the method, and the per-system scores, so the next assessment is comparable to this one.
6. Remediation roadmap	Program	Sequenced by layer dependency, with owners, effort estimates and the maturity level each step reaches.
7. Declaration template	Ongoing	A reusable specification format for declaring a new agent's intended behavior. The artifact that changes how they deploy.
8. Control set	Compliance	The governance controls implied by the findings, written in a form an auditor will accept.
9. Framework crosswalk	Compliance	How the five layers map to the obligations they are actually subject to.

Writing rules for the leave-behind

- **Every finding cites evidence.** “We observed X in system Y on date Z”, not “practices appear immature”. An audit that asserts is no better than the policy documents it is auditing.
- **Separate fact from recommendation.** What is true, then what to do about it. Clients need to be able to accept the first while disagreeing with the second.
- **Name what is working.** An all-negative report reads as a sales document and gets discounted accordingly.
- **Make four artifacts reusable.** Register, coverage map, declaration template, control set. These are what keep you present after the engagement ends.

Framework crosswalk

Clients buy governance assessments because a framework obliges them to. The crosswalk translates the five layers into the language of the obligation they face, which is what makes the report usable in their compliance program rather than adjacent to it.

Layer	Maps conceptually to
Declaration	NIST AI RMF: GOVERN and MAP (context, intended purpose, documented system characteristics). ISO/IEC 42001: AI system objectives and documented intended use. EU AI Act: technical documentation and intended-purpose declaration.
Conformance	NIST AI RMF: MEASURE (evaluating whether the system performs as intended). ISO/IEC 42001: performance evaluation and monitoring. EU AI Act: accuracy and robustness obligations for higher-risk systems.
Containment	NIST AI RMF: MANAGE (risk treatment and control of system behavior). SOC 2: logical access and change-control criteria applied to agent authority. Data-protection regimes where egress and access scope are the substance of the obligation.
Consequence	NIST AI RMF: MEASURE and MANAGE (impact assessment, downstream effects). EU AI Act: human oversight and post-market monitoring. Sector duty-of-care regimes concerned with harm to individuals rather than with system correctness.
Evidence	EU AI Act: record-keeping and automatic logging obligations. ISO/IEC 42001: documented information and internal audit. SOC 2: the evidentiary basis of every criterion. This layer is what makes the other four auditable.

Use the crosswalk conceptually, verify it specifically. The mappings above are directionally sound and sufficient to structure an engagement. Clause-level assertions to a client's auditor should be confirmed against the current text of the relevant framework and, where the client is regulated, with their counsel. Never let a crosswalk in a slide deck become a compliance opinion.

APPENDIX

Evidence request checklist

Issue at kickoff. What arrives, and what does not, is itself the first finding of the assessment.

Policy and scope

- The current AI policy, acceptable-use policy, and any related standards or guidelines.
- Any prior AI governance assessment and its remediation status.
- The regulatory or framework obligations the organization considers applicable.

Layer 1: Declaration

- Inventory of deployed AI systems and agents, with business owner and technical owner.
- For each: the specification of intended behavior, in whatever form it exists.
- Approval records: who authorized deployment, on what basis, and when.
- Change history for both the systems and their specifications.

Layer 2: Conformance

- Telemetry samples covering a representative period, not a curated window.
- The monitoring rule set, with the date each rule last produced a finding.
- Finding disposition records: raised, owned, resolved.

Layer 3: Containment

- Complete tool and function call logs, unfiltered by allow list.
- Data access and retrieval logs with identity attribution.
- Network egress records for agent-executing environments.
- Credential inventory: what each agent can authenticate as.
- Output scanning configuration and results, if any exist.

Layer 4: Consequence

- Architecture diagrams showing what agent actions trigger downstream.
- Multi-agent topology: which agents invoke or consume the output of others.
- Rate, cost and consumption telemetry over time.
- Human approval records, with timestamps for presentation and decision.

Layer 5: Evidence

- Log retention configuration and policy.
- Log integrity controls: immutability, signing, access restrictions.
- Prior audit or assessment reports, and their remediation status.
- The applicable regulatory record-keeping obligations, as the client understands them.

Notes and disclaimer

This framework is a professional methodology for assessing AI governance maturity. It is not a certification scheme, and an assessment conducted under it does not constitute a compliance opinion, a legal opinion, or an attestation under any statutory or industry framework. Regulatory mappings are conceptual and should be verified against current framework text; clients in regulated sectors should involve qualified counsel. Findings reflect evidence available during the engagement window and the scope agreed at kickoff.

The Declared Behavior Framework, version 1.2. Prepared by Auspica LLC.