PUBLIC TECHNICAL NOTE · V0.5 · 2026-04-28

Auctor

Governed AI memory and auditable release: a public note on what the framework guarantees, what was measured under those guarantees, and what is intentionally not disclosed.

Author Angad Bank Disclosure claim, contract & evidence only Mechanism licensed

Abstract

Large language models are powerful interfaces for language, reasoning, summarization, and planning. They are not, by themselves, reliable systems of record. In regulated environments the operative question is not whether a model can produce a plausible answer; it is whether the organisation can prove which answer is allowed to be used.

That proof has many parts. Which source did the answer come from? Was the source current at the time of the request? Was the requester authorised? Was restricted content redacted? Were unsupported claims refused rather than guessed? Can the entire decision path be reconstructed from an audit trail? A conventional language model does not natively answer any of those questions. Conventional retrieval answers some of them, weakly, and only for sources it happens to find relevant. Auctor is built so that an organisation can answer every one of them, mechanically, for every release.

This document is a deliberately narrow public account. It describes the *claim* Auctor makes, the *contract* it enforces against every fact-bearing output, and the *evidence* gathered to date under that contract. It does not describe the implementation, the underlying research, or the architecture by which the contract is satisfied. Those remain licensed and outside the public record by design.

The core claim is bounded. Auctor is not a claim of universal AI truth, universal model correctness, or unrestricted reasoning accuracy. In current local evaluation gates Auctor has demonstrated exact source-bound performance across very large real public scientific catalogues, across diverse enterprise document surfaces, across a full enterprise-runtime control contract, and across open-ended natural-language requests evaluated through more than one local model family. These are bounded transaction pass rates under explicit contracts. They are not claims that the underlying model is universally correct. The model is the interface. The governance contract is the authority. What leaves the system is what the contract permits.

1 The problem: fluency is not authority

Modern AI is often evaluated on whether an answer sounds plausible, useful, or semantically close to the requested information. That is sufficient for general consumer assistants. It is not sufficient for banks, insurers, public agencies, defense organisations, pharmaceutical companies, legal teams, healthcare providers, regulated operators, or critical infrastructure providers. In those settings an answer is not acceptable merely because it sounds right; it must be traceable, authorised, current, policy-compliant, and defensible. The gap between “plausible” and “defensible” is where most enterprise AI deployments stall.

A conventional language model has no native institutional authority. It may summarise a document well without knowing whether the document is superseded; it may retrieve a relevant source without knowing whether the requester is permitted to see it; it may cite something without that citation being the correct authority; it may answer with confidence without that answer being auditable. None of those failure modes are visible at the surface, which is precisely why they are dangerous. A model that confidently cites the wrong policy version is more harmful than a model that says it does not know.

This explains the pattern that organisations describe to us repeatedly: many AI pilots, fewer AI systems that employees can safely rely on for high-stakes decisions. The missing layer is not another chatbot, not a longer context window, and not ordinary semantic search. The missing layer is governed memory and release: an explicit decision about what the system is allowed to know, what it is allowed to cite, what it is allowed to release, what it must refuse, what it must redact, what it must update, what it must revoke, and what it must be able to defend on the record. Auctor takes that operational gap as its starting point.

2 What Auctor provides

Auctor is a governed AI memory and release framework. The remainder of this section describes what the framework guarantees in operational terms. It deliberately does not describe how those guarantees are implemented. The mechanism — the internal representation, the validation pathway, the decision logic, the update procedure, and the verification design — is licensed and is not part of this public note.

Auctor's surface to an operator can be reduced to five guarantees. Each guarantee is a property of the contract, not a description of any internal component. Each one is independently measurable, and each one becomes a control point that an organisation can audit, exercise, and stress.

TABLE 1. THE FIVE OPERATIONAL GUARANTEES AUCTOR COMMITS TO AGAINST EVERY FACT-BEARING OUTPUT.

Guarantee	What the framework commits to	Operator's control point
Source-binding	A fact-bearing output is traceable to an identified, authorised, and current source, or it is refused. There is no path by which a plausible answer reaches the user without an attached source decision.	source register
Policy compliance	Every release respects role, tenant, jurisdiction, retention, redaction, legal hold, and exception rules defined by the operator. The contract enforces the rules; the operator authors them.	policy register
Verification	A release is conditional on a verification step that owns final correctness. If verification does not hold, the system corrects, redacts, refuses, or abstains rather than guess.	verifier signal
Updateability	New information, corrections, deletions, and revocations are reflected in releases immediately, without retraining the underlying model. Institutional knowledge moves at institutional speed.	update log
Audit reconstruction	Every release, refusal, redaction, and abstention is recorded so that the decision can be reproduced and inspected after the fact. The record is what makes the system defensible, not the answer.	audit chain

These properties are contractual: they define what the deployment must deliver. They are also independent. An operator can stress source-binding without touching policy compliance; can review policy compliance without touching the verifier; can rotate the audit chain without disturbing the update log. That independence matters because real governance never lands in one team. Source authority lives with data owners, policy with legal and compliance, verification with the model and operations teams, updateability with the data engineering team, and the audit chain with security and risk. Auctor's contract is shaped so that each of those teams gets a control point that belongs to them, in language they already use.

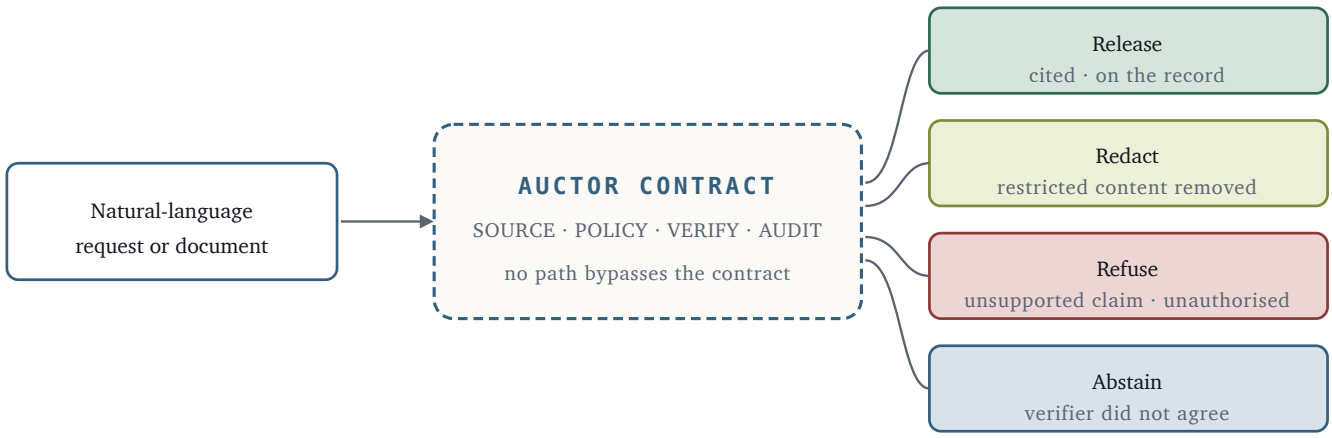

Figure 1. The four outcomes the contract permits to leave the system. Every fact-bearing request resolves to exactly one of them, with the chosen outcome and its justification recorded on the audit chain. The internal decision logic that selects the outcome is licensed and not depicted here; what is shown is only the surface visible to operators and auditors.

3 Why retrieval alone is not enough

Retrieval-augmented generation is useful. It can improve grounding by placing relevant documents near the model. It does not, by itself, solve institutional correctness. A semantically close source may still be the wrong source: it may be stale, superseded, from the wrong tenant, from the wrong jurisdiction, unavailable to the requester, not citeable, under legal hold, or insufficient for the decision being made. In high-stakes work those distinctions are not edge cases; they are the work itself.

The right way to phrase the difference is operational rather than architectural. Retrieval asks *what seems relevant?* Governed memory asks *what is authorised, current, source-bound, and safe to release?* Those are not the same question, and a system that optimises the first question well does not automatically answer the second.

Recent comparative evaluation makes the boundary concrete. Against a 100M+ row real public scientific catalogue with tens of thousands of probes, a heavily engineered metadata-filtered retrieval system matches Auctor's exact-source result only when it adopts the same kind of typed source-selection contract that Auctor enforces by construction. Lighter retrieval variants either fail the regulated exact-source task by orders of magnitude, or remain accurate only by falling back to raw text-body scanning that runs orders of magnitude too slow to use. The interpretation is not that retrieval cannot be made to work; it is that *exact governed output requires a typed source-selection contract, regardless of which framework name is over the top of it*. Auctor treats that contract as the primary object. Retrieval can participate, but it does not certify.

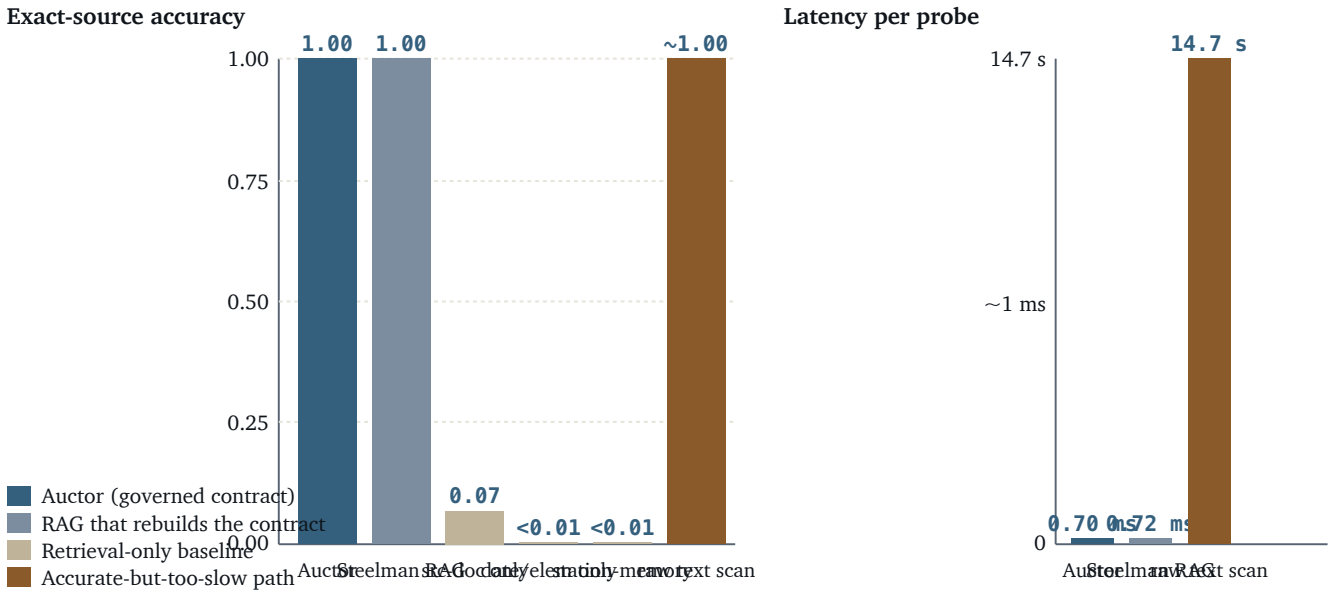

Figure 2. Comparative exact-source accuracy and latency over a 100M+ row real public scientific catalogue, evaluated with tens of thousands of probes. Auctor and a heavily engineered metadata-filtered retrieval system ("steelman RAG") tie at exact-source accuracy 1.00 and sub-millisecond latency only because the steelman path internally rebuilds a typed source-selection contract equivalent to the one Auctor enforces by construction. Retrieval-only baselines fall to single-digit-percent accuracy at the same scale; a raw text-body scan can be exact but is roughly four orders of magnitude too slow to use. The lesson is operational, not architectural: exact governed output requires a typed source-selection contract.

Auctor is therefore not "RAG but better." It is a governance and release framework. Retrieval can nominate; governance certifies. That phrasing matters when comparing options: a procurement team should not be choosing between Auctor and a retrieval system, because they answer different questions. They should be choosing whether to operate their AI under a governance contract or without one, and if under one, which contract.

4 The governance contract

The public form of Auctor is best described as a contract rather than as an algorithm. A governed transaction commits the system to a small set of obligations against every fact-bearing output: a request must be parsed and classified by what it actually asks for, distinguishing information from policy from action; candidate knowledge must be sourced only from approved memory; a release must be permitted only when source, time, tenant, access, policy, and validity checks all hold; if any check does not hold, the release must be corrected, redacted, refused, or abstained on rather than guessed; and every outcome must be recorded so that the decision can be inspected later.

This is intentionally framed as obligations rather than as a sequence of steps. The order in which Auctor satisfies the obligations, the internal structure of each check, and the decision logic that resolves conflicts among them are not part of

this public note. What is public is what the operator can demand and inspect from the outside.

One word in the contract deserves explicit clarification because it is widely misused in vendor literature. In Auctor, “canonical” does not mean “automatically true.” It means in the form the contract can check. Truth and authority come from source authority, policy, timestamp, tenant scope, legal hold, verifier agreement, and audit replay — not from canonicalisation itself. Canonicalisation makes a record auditable; the surrounding contract makes it authoritative. This distinction is what allows Auctor's perfect transaction pass rates in evaluation to coexist honestly with the well-known probabilistic nature of language models. The model and any interpretation step upstream of the contract remain statistical; what changes after canonicalisation is that release becomes a deterministic decision against the contract.

The contract changes what AI is allowed to do. The model is no longer free to present plausible language as institutional fact. If the system cannot verify the answer, the correct behaviour is not to guess. The correct behaviour is to abstain, refuse, redact, or escalate — in each case, on the record. That is the entire point of the contract, and it is the reason a released answer in Auctor is meaningfully different from a confident answer in an ungoverned system.

---

5 Evidence summary

The evidence below is summarised in public-safe language. It does not reproduce internal track structure, internal codes, or the internal experiment ledger; those are licensed. The numbers and pass criteria reported here are taken from the same internal record that licensed partners receive in full, but the public table is organised by what was tested, not by which experiment did the testing. The table is followed by a scale figure to make the numbers concrete.

TABLE 2. PUBLIC-SAFE SUMMARY OF THE VALIDATION EVIDENCE ACCUMULATED TO DATE.

Evidence area	What was tested	Public result
REAL-DATA SCALE		
Public scientific catalogue at scale	Exact source-bound retrieval over a real, public, 100M+ row scientific catalogue, evaluated with tens of thousands of probes against the typed source-selection contract.	Field, retrieval, source, and value accuracy **1.00** in the local gate.
Multi-model front end on the same catalogue	Two independent local model families used as the natural-language front end into the governed contract on the same catalogue, with a tuned cross-model bridge.	Exact field, retrieval, source, and value accuracy **1.00** across both model families in the tuned gate.
Natural-language multi-step workflow	A normal-language request producing source selection, calculation, intermediate notes, a rendered report, and verification — end-to-end through the governed contract.	Workflow and report-verification rate **1.00** in the local gate.
COMPARISON VS. RETRIEVAL-ONLY SYSTEMS		
Same-catalogue, multi-system comparison	Tens of thousands of probes against the same 100M+ row catalogue, comparing Auctor with a steelman metadata-filtered retrieval system and several retrieval-only baselines.	Auctor and the steelman both reach **1.00** only where the typed contract is present; retrieval-only baselines either fail exact source selection or become orders of magnitude too slow.
ENTERPRISE CONTENT SURFACES		
Diverse post-extraction document surfaces	PDF text, OCR text, email, contracts, tables, tickets, chats, and logs — sampled across many enterprise-shaped variants.	Exact canonical accuracy, source-binding accuracy, and rejection accuracy **1.00** in the local gate; **no raw document text** recorded in the result artifact.
PRODUCTION-STYLE RUNTIME CONTRACT		

Evidence area	What was tested	Public result
Local enterprise-runtime integration	Identity and access control (IAM/RBAC/ABAC), tenant envelope encryption, cryptographic key rotation, durable queues with dead-letter handling, observability, tamper-evident audit, legal holds, deletion-SLA enforcement, and backup expiry — exercised together as one harness.	Zero false allows, zero false denies, full encrypted-payload rate, full trace coverage, deletion SLA met where permitted, legal-hold deletions blocked, backup expiry completed, audit chain validated.
OPEN-ENDED NATURAL-LANGUAGE REQUESTS		
Held-out enterprise reasoning, two model front ends	Held-out, normal-language enterprise cases evaluated through more than one local model family, including unseen tenant values and messy unseen document layouts.	Governed field decode and open-ended reasoning accuracy **1.00** across the tested local cases; **zero** false authoritative answers; **zero** false abstentions; one-sided 95% lower confidence bound **> 0.99** for open-ended reasoning under the contract.

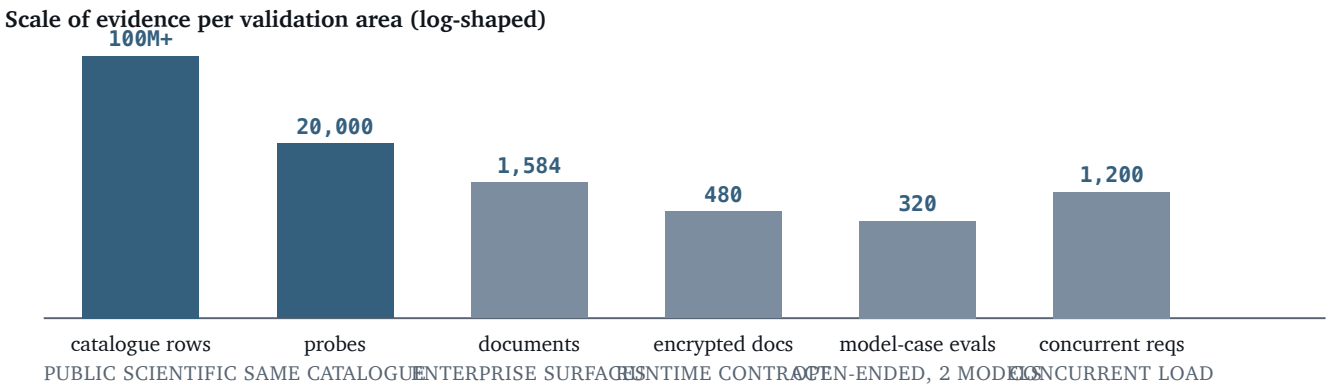

Figure 3. Approximate scale of the validation evidence behind each public claim. Bar heights are illustrative (log-shaped) so that all gates remain visible on one axis; the printed numbers are the reported quantities. The combination matters more than any single bar: scale on real public data, breadth on enterprise surfaces, depth on the production runtime contract, and adversarial reach on open-ended natural-language requests.

These results are bounded local transaction evidence. They do not prove universal truth. They prove that, under explicit contracts, the framework separates model interpretation from institutional release authority, and that the separation can

be measured. The next section interprets the four operational claims that the table supports and states the boundary that goes with each one.

---

6 What the evidence verifies

6.1 Governed source selection at real-data scale

A 100M+ row real public scientific catalogue is exercised end-to-end with exact field, source, and value selection under the contract. A normal-language request to the same store can produce a verified multi-step report — that is, the system can take one ordinary request, drive a model-backed front end into the governed contract, retrieve the right rows, perform the calculations, write intermediate notes, render a human-language report, and verify the report against the underlying records, all without leaving the contract. The supported public claim is that Auctor can support model-facing natural-language workflows while keeping source selection, calculation, citation, and release inside a governed transaction path. The boundary is that this is real public scientific data; private mixtures with conflicting authorities and human exception routing remain a separate validation layer.

6.2 Enterprise document surface coverage

Diverse post-extraction enterprise surfaces — PDF text, OCR text, email, contracts, tables, tickets, chats, and logs — are bound to source-bound records or rejected, with rejection itself measured and recorded. Auctor can therefore bind diverse enterprise document surfaces to governed records in a local evaluation gate. The careful boundary is that this is post-extraction document-surface validation: it does not claim that every raw enterprise file in every customer environment can be ingested without exception handling. Raw binary PDF extraction at scale, conflicting source authorities, low-confidence routing, and human exception workflow are all real; they just sit in the next validation layer rather than this one.

6.3 Local enterprise runtime semantics

A production-shaped runtime contract — identity, encryption, key rotation, durable queues, observability, tamper-evident audit, legal holds, deletion SLA, backup expiry — is enforced inside one integrated harness rather than as a checklist of optional features. Auctor has therefore been tested as a local governed AI runtime contract, not only as a memory concept. The careful boundary is that this is not external cloud certification: it is a local integration gate that defines what must be preserved when the same contract is moved into managed enterprise infrastructure. Customers should expect the contract to translate, but the certification itself belongs in their environment, not in this evaluation.

6.4 Open-ended enterprise requests are corrected by the contract

This is the result that most directly addresses the commercial concern. Across more than one local model family, the raw model output was deliberately imperfect in the underlying decode step — the framework did not pretend the model was

always right. What it did instead was correct it. Governed release reached **1.00** on held-out field decode and on open-ended reasoning under the contract, with zero false authoritative answers and zero false abstentions in the tested local cases, even when the raw model output below the contract was significantly worse. The model can be imperfect; the contract corrects, verifies, refuses, or abstains before any answer is used. This is the core commercial claim, and the next figure summarises the confidence bounds Auctor has reported around it.

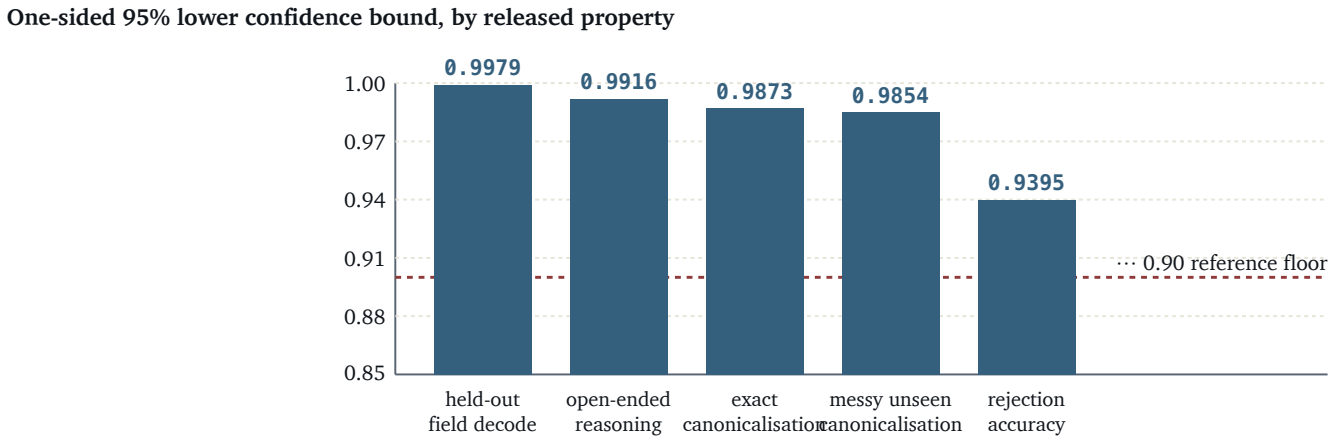

Figure 4. One-sided 95% lower confidence bounds on released-output properties measured under the open-ended natural-language enterprise contract, evaluated through more than one local model family. All five released properties remain above 0.93 even at the lower confidence bound; four of five remain above 0.98. Reading these alongside the zero false-authoritative-answer and zero false-abstention counts is the intended summary: when the contract releases an answer, the answer is the recorded quality; when it cannot, it refuses or abstains rather than guess.

7 Why this matters for regulated AI

Regulated organisations do not need AI that merely sounds intelligent; they need AI that can be controlled. The relevant operational questions are familiar to anyone who has sat through a model-risk review or a data-protection audit. Can the system show where an answer came from? Can it distinguish authorised sources from merely relevant ones? Can it refuse unsupported claims? Can it enforce user access? Can it redact restricted information? Can it respect deletion, retention, and legal hold? Can it produce an audit trail after the fact? Can it continue to use a stable model while institutional knowledge changes? Auctor is built around those questions, and each one corresponds to an explicit guarantee that the contract enforces and the audit chain records.

The framework's posture aligns with the direction of contemporary AI governance. NIST's AI Risk Management Framework is a voluntary framework for managing AI risk and incorporating trustworthiness into AI design, development, use, and evaluation. The OECD AI Principles promote trustworthy AI and explicitly include transparency, explainability, and human oversight. The EU AI Act establishes a risk-based legal framework for AI, including obligations

for risk management, transparency, record-keeping, human oversight, accuracy, robustness, and cybersecurity for high-risk AI systems. Auctor does not replace any of those processes; it gives them a technical object to govern. The contract becomes inspectable: what sources exist, who may access them, what policies apply, what outputs were allowed, which outputs were refused, and what proof is attached. The figure below summarises how the public guarantees in §2 map to those frameworks. It is a coverage map rather than a certification claim.

Auctor guarantees against contemporary AI governance frameworks

AUCTOR GUARANTEE	NIST AI RMF	OECD AI	EU AI ACT
Source-binding	MAP / MEASURE	transparency	record-keeping
Policy compliance	GOVERN	accountability	risk management
Verification	MEASURE / MANAGE	robustness	accuracy / robustness
Updateability	MANAGE	human oversight	human oversight
Audit reconstruction	MEASURE	transparency	record-keeping

Figure 5. Coverage map between the five Auctor guarantees and three contemporary AI-governance frameworks. The pills indicate where each guarantee provides direct technical evidence in support of a published framework requirement; they do not constitute a formal certification of conformity. Certification, where required, belongs in the operator's environment under their auditor's review.

8 What Auctor is not

Auctor is not a claim that language models are universally truthful, that every open-world question can be answered perfectly, or that all tasks can be reduced to deterministic transactions. It is not a replacement for security review, legal review, independent audit, customer compliance assessment, or human accountability. The framework provides a technical object that those processes can govern; it does not absorb them.

Auctor is also not ordinary semantic retrieval. Retrieval can nominate candidates, but a candidate is not an authority. Whether a candidate becomes a release is decided by the contract, not by the similarity score. By the same token, Auctor is not “more context.” A longer context window may expose the model to more text without deciding whether that text is current, authorised, tenant-specific, citeable, or safe to release. Bigger windows do not produce smaller risk.

Auctor is not a model provider. It can be operated with different model front ends, and the multi-model evidence in §5 is intentionally there to show that the contract does not depend on any single model being correct. The goal is not to make

any one model the institutional system of record. The goal is to let an organisation use AI through a governance contract that the organisation actually controls.

Finally, Auctor is not a full disclosure of mechanism. The implementation, the underlying research, the internal representation, and the decision logic are licensed, and this document is intentionally limited to the contract and the evidence. A reader who wants to understand how the framework satisfies the contract should request a licensed technical briefing rather than try to reconstruct it from this note.

---

9 What remains to be validated

The current evidence is strong enough to support a serious technical trust position, but it is not the end of validation, and it would be dishonest to present it as such. The next layers fall into roughly six bands, and the description below is what an organisation should expect to see covered before placing Auctor on a critical workflow.

The first band is independent replication. External parties should reproduce the bounded transaction results using documented evaluation contracts. This is the standard scientific check, and it is the cleanest way to retire the “internal local gate” qualifier from the public claim.

The second band is customer-environment pilots. The local Y-class harness uses local IAM/KMS-style components; managed Postgres or object storage, real enterprise IAM, managed KMS or HSM, service queues, SIEM and observability, backup systems, and incident-response workflows all live in the customer's environment. Each of those substitutions is its own validation step, and Auctor's pilot scope is built to make those substitutions one at a time so that any regression has an obvious cause.

The third band is adversarial testing. Malicious documents, prompt injection, stale-source attacks, tenant confusion, policy conflicts, deletion pressure, and attempts to tamper with the audit chain are part of the threat model and should all be exercised by a customer's red team before the framework moves onto a critical path.

The fourth band is governance review by the legal, compliance, risk, security, and data-protection functions of the deploying organisation. Their job is not to read this note; their job is to inspect the contract and the audit trail against their own obligations. The contract is designed to be inspectable in exactly that way.

The fifth band is operational monitoring. Latency, false allow rate, false deny rate, abstention behaviour, exception workflows, drift, and user reliance are all behaviours that move once a system is in production. None of them are visible in a one-shot evaluation, and all of them belong on a dashboard that the deploying team owns.

The sixth band is boundary mapping. Some workflows are transaction-safe under the contract; others remain advisory. The work of separating those workflows is collaborative — domain experts know which decisions can be carried by the contract, and the framework's job is to make the boundary explicit. The right public claim is therefore narrow: Auctor has demonstrated a local governed AI runtime contract; the next step is external deployment validation under customer infrastructure and compliance review. That claim is strong, honest, and defensible, and it is the one we ask readers to evaluate.

10 Conclusion: governance is the authority

Auctor reframes the enterprise AI question. The question is not whether a model can produce an answer; it is whether the answer can be used. In regulated settings, usable means source-bound, policy-compliant, auditable, current, revocable, citeable, redacted where necessary, and refused when unsupported. Auctor is a framework for making those properties part of every release rather than a property of an answer that happened to be lucky.

The strongest public summary is that Auctor makes AI usable where correctness, auditability, and authorisation matter, by separating model fluency from institutional authority. The model is the interface; the contract is the source of record; the contract decides what may be used. Across current local gates, Auctor governance has demonstrated exact source-bound behaviour in large real-data workflows, diverse enterprise document-surface canonicalisation, infrastructure-style runtime controls, and open-ended enterprise request handling under more than one model front end. These are not universal truth claims; they are governed transaction results. That distinction is exactly why Auctor matters. It does not ask organisations to trust AI because it sounds confident. It gives organisations a way to trust AI only when the answer is authorised, verified, and on the record.

PUBLIC DISCLOSURE BOUNDARY

This document discloses what the contract guarantees and what was measured under those guarantees. It does not disclose the underlying research programme or its mathematical foundation, the framework's internal architecture or layer decomposition, the internal representation used by the contract, how memory is accessed, selected, updated, or corrected, how verification is performed, how new structure is admitted, how model outputs are bound to the contract, how the audit chain is constructed, the design of correction, redaction, or abstention logic, the internals of the scoring harness, the canonical record format, how evaluation cases are generated, or production deployment internals. Those are licensed.

Partners under licence receive the implementation documentation, the architecture, the underlying research, and the technical evaluation harness. Partners not under licence receive this public note and the operational claim that follows from it. A reader who wants to understand how the framework satisfies the contract should request a licensed technical briefing rather than attempt to reconstruct it from this document.

· Licensing and contact

The implementation, underlying research, and complete experimental record are licensed. Material is available to qualified partners under licence. Pilot enquiries, licensing requests, and customer-environment validation engagements all begin from the same address: `impact@angad.swiss`.

· Official references

[1] NIST. *AI Risk Management Framework.* nist.gov/itl/ai-risk-management-framework

[2] NIST. *Artificial Intelligence Risk Management Framework (AI RMF 1.0).* nist.gov/publications/artificial-intelligence-risk-management-framework-ai-rmf-10

[3] OECD. *AI Principles.* oecd.org/en/topics/ai-principles.html

[4] European Union. *Regulation (EU) 2024/1689 (AI Act).* eur-lex.europa.eu/eli/reg/2024/1689/oj

[5] NOAA / NCEI. *Global Historical Climatology Network — Daily (GHCN-Daily), Version 3.* ncei.noaa.gov / GHCN-Daily v3