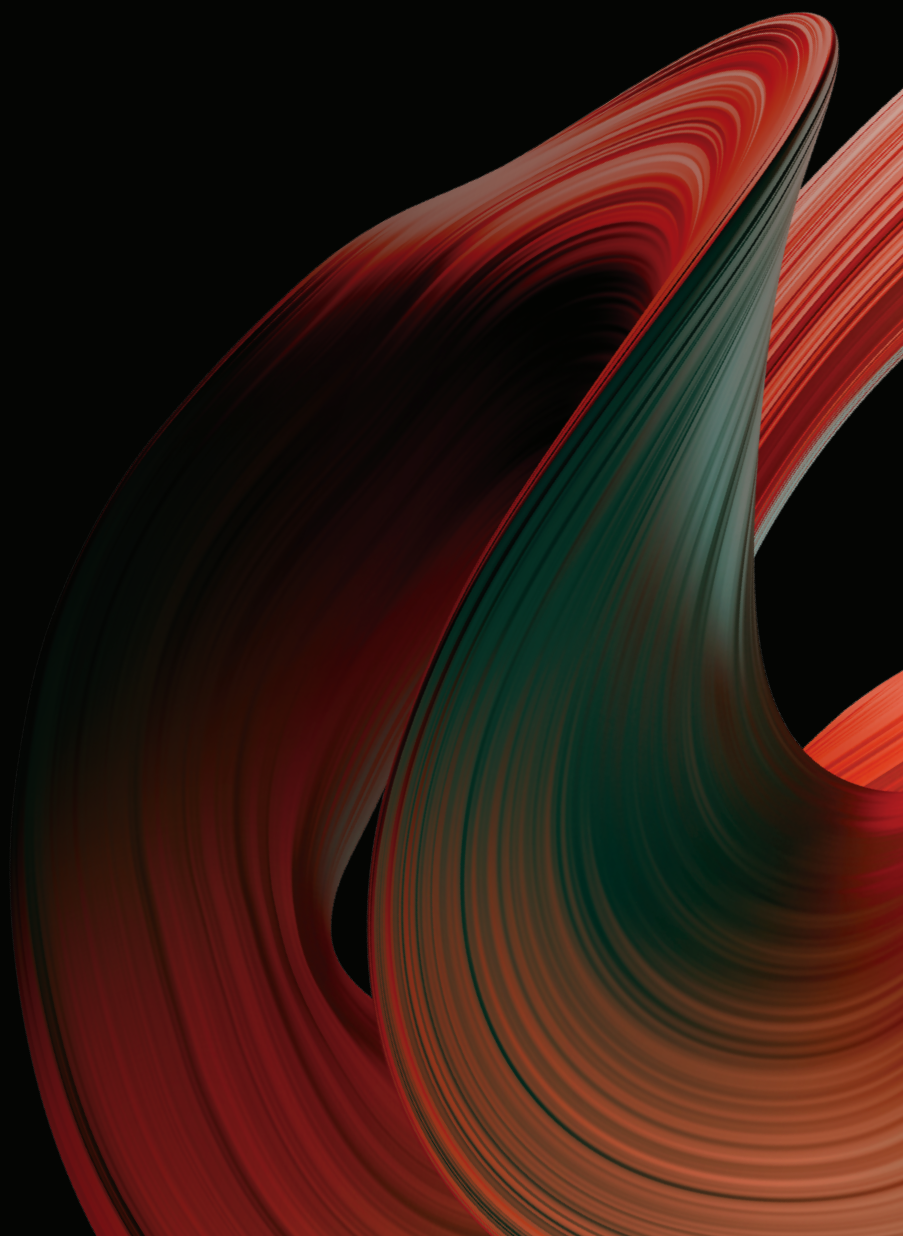
seekr | WHITE PAPER

QA for AI

What Software Quality Assurance Taught Us, and What AI Now Requires



Executive summary

Quality assurance for AI has to answer three questions:

01 Does the output pass?

02 Why did the system produce it?

03 Can you prove both after the fact?

Traditional software QA answers the first question well because conventional software is deterministic. The same input produces the same output, so a test passes or fails.

AI systems are probabilistic, shaped by data rather than only by code, and increasingly trusted to act and decide on their own: who gets an offer, what a customer pays, whether an identity check passes. A pass rate tells you how often a system was right during testing. It doesn't tell you why it was right, whether it will stay right after the next model update, or what went wrong on the occasions it failed.

That gap has moved from a quality problem to a defensibility problem. When an AI decision affects someone's income, price or access, the enterprise may have to show a customer, regulator or court why the model decided as it did and that it was tested for unfair outcomes. A pass rate can't do that.

A pass rate tells you how often a system was right during testing. It doesn't tell you why it was right.

QA has always required traceable requirements, defined expected results, reproducible defects, regression control, independent verification, and an audit trail. Each of these disciplines carries over to AI, and most of the QA playbook still applies. What changes is the evidence. AI QA has to capture where the data behind an output came from, which data, model and steps produced it, how that data changed along the way, and where it was held while that happened.

This paper is written for the QA, defect management, legal, privacy, and security leaders who now share responsibility for AI quality. It closes with a way to run AI QA as an ongoing operating rhythm and a checklist to scope an Enterprise AI Assurance program or evaluate AI QA vendors.

What QA has always required

Every mature QA discipline, in software or in physical goods, rests on the same eight practices. None of them is new, and enterprises already pay for all of them.

Requirements traceability. Every test traces to a requirement, and every requirement has a test. A traceability matrix shows what was specified, what was checked and what was missed.

A known expected result. A test needs an oracle: for input X, the correct output is Y. That is what makes pass and fail meaningful.

Reproducibility and root-cause analysis. A defect you can reproduce is a defect you can fix. Teams trace failure back to the line of code, the configuration or the dependency that caused it.

Regression and change control. Nothing ships unless what worked before still works. Builds are versioned, changes are approved and every release is retested.

Non-functional testing. Quality covers more than correctness: performance, reliability, security, usability, and safety, as catalogued in quality models such as ISO/IEC 25010.

Independent verification and validation. The team that builds a system doesn't grade it. Independence is why auditors, IV&V contractors and separate QA organizations exist.

Production monitoring and defect triage. QA continues after release. Escaped defects are detected from production signals, isolated, triaged and fixed, and each one becomes a new test.

Evidence. Test records, signoffs and change tickets let an auditor, regulator or executive confirm that the controls ran. If it was not recorded, it did not happen.

Physical goods quality adds one more lesson that matters for AI: chain of custody. When a product is recalled, a retailer or manufacturer has to identify the lot, the supplier, the facility and the date. Inspection tells you a product failed. Traceability tells you which other products are affected and where the failure entered. AI needs the second capability as much as the first.

Inspection tells you a product failed. Traceability tells you which other products are affected and where the failure entered.

Why AI breaks the old assumptions

AI doesn't make QA obsolete. It breaks seven assumptions that traditional QA quietly depends on, and quality is now the gate on getting AI into production. Cisco reported in March 2026 that 85% of its major enterprise customers were experimenting with AI agents but only 5% had moved them into production (Security Boulevard). LangChain's 2026 State of AI Agents report found 32% of organizations name quality as the single biggest barrier to deployment (as reported by Testriq).

Outputs are probabilistic. The same prompt can produce different answers, and a model reviewing the same image twice can return opposite verdicts. Pass or fail becomes a rate, and a rate needs a sample size, a threshold, and a definition of "good enough" that the business owns.

Behavior comes from data, not only code. You can't read an AI system's logic in its source. The requirement "answer return-policy questions correctly" maps to the documents, model and prompts behind the answer, not to a function.

The invisible middle is the most expensive break, because it makes every other one harder to fix.

The system changes without a code change. A provider updates a model, a retrieval index picks up new documents, or a prompt is edited. Behavior shifts and no release ticket records it.

The correct answer is often a judgment. A summary, a recommendation or a product comparison has no single correct output. Someone has to define the criteria, and they differ by use case.

Agents act, not just answer. An agent that issues a refund, updates an order or calls another system has to be tested for what it did and whether it was authorized to do it, not just for what it said.

The middle is invisible. Input goes in and output comes out. What happens in between, including which evidence the system used and which it ignored, is a gray area unless the system is designed to show it. Without that view, root-cause analysis turns into guesswork.

Some defects look like success. A model can meet its accuracy target while its errors fall on one group of users more than others, or drift toward favoring one group over time. No single answer is wrong, so a test that checks answers one at a time won't see it. Outcome fairness is a quality attribute, and decision models need it tested.

The invisible middle is the most expensive break, because it makes every other one harder to fix. A team that can measure a failure but can't explain it can only retry, re-prompt or swap models, and hope the next run goes better. When the failure is a skewed outcome, it also can't show that the skew wasn't intended.

From QA to AI QA: the mapping

Each traditional discipline carries over to AI, but the evidence it requires changes. The right-hand column is what an Enterprise AI Assurance program should be able to produce on demand.

QA discipline	Traditional practice	What AI changes	What AI QA must require
Requirements traceability	Test cases map to written requirements	Behavior is driven by data, models and prompts, not code alone	Provenance and attribution: for any output, show the source data and which data, model and steps produced it
Expected result	Deterministic oracle: input X gives output Y	Outputs vary and quality is often a judgment	Business-defined evaluation criteria scored across a representative test set, with pass thresholds the business owns
Root-cause analysis	Reproduce the defect and trace it to code or configuration	The reasoning between input and output is not visible by default	Explainability: why the system produced that specific output, including evidence retrieved versus evidence actually used, and the ability to reconstruct a single decision after the fact
Regression and change control	Versioned builds; retest every release	Models, data and prompts change outside the release cycle	Continuous re-evaluation against the same criteria whenever a model, dataset or prompt changes, with results versioned
Release gate and acceptance	Sign-off against acceptance criteria	Many candidate models could do the job at very different costs	Evidence-based model selection: compare models side by side on the same criteria and choose the smallest one that reliably meets them
Security testing	Penetration testing and vulnerability scans	Prompt injection, data leakage and manipulation through inputs	Adversarial evaluation included in the standard test set, not run as a one-off exercise
Fairness and non-discrimination	Equal-treatment and accessibility requirements	Decision models can skew outcomes across groups with no single answer wrong	Outcome fairness testing: measure results across groups and segments, explain or correct any disparity, and re-test after every change

QA discipline	Traditional practice	What AI changes	What AI QA must require
Authorization testing	Verify users can only do what their role allows	Agents take actions across systems on a user's behalf	Agent behavior controls: record what an agent did and confirm it stayed within its authorized boundaries
Defect management	Log, triage, fix, verify	Users and reviewers need a path to challenge an output	Contestability and defect triage: detect drift, regression and inconsistent verdicts in production, give reviewers a way to flag, override and correct an output, with the correction recorded
Independence and evidence	IV&V; test records for auditors	Vendors often evaluate their own models; evidence may sit in a third-party cloud	Sovereign evidence under your control: evaluation and audit records generated and held where the enterprise controls them, whether in SaaS or on-premises
Chain of custody	Lot and supplier traceability for recalls	Data is transformed between source and output	Lineage: how the data changed from its source to the output, so every affected output can be found when a source turns out to be wrong

The three questions AI QA must answer

The mapping reduces to three questions. An AI QA program that answers only the first is a testing program, not an assurance program.

Does it pass? Evaluation against criteria the business defines, across the models and configurations under consideration, at a scale that makes the pass rate meaningful. This is where most AI testing stops.

An AI QA program that answers only the first is a testing program, not an assurance program.

Why did it produce this output? Explainability: where the data came from, which data, model and steps produced the output, and how the data changed along the way. A score tells you an answer was wrong. An explanation tells you what to fix: the source document, the retrieval step, the model or the prompt.

Can you prove it later? Evidence that stands up when a customer, auditor, regulator, executive, or court asks. That means a complete record of what was evaluated, what was decided and why, stored where the enterprise controls it, with a path to contest and correct an output.

The second question is what separates assurance from testing. Testing without explanation produces a growing list of failures and no reliable way to prevent the next one.

What legal, privacy, and security teams need from AI QA

AI quality is no longer only an engineering question. When an AI decision is challenged, counsel needs an evidence record, not a dashboard.

Consider a lender using an AI model to recommend mortgage approvals. Fair lending rules prohibit decisions that discriminate by race or religion. If an applicant challenges a denial, the lender has to show what the model considered and that its outcomes were tested across groups. If the decision logic can't be traced, the lender may be unable to prove the model was fair, even if it was.

When an AI decision is challenged, counsel needs an evidence record, not a dashboard.

For any single AI decision, a defensible record should show:

- The model and version that made the decision, and when.
- The input, and the data the decision relied on, with its source.
- The criteria the model was evaluated against before it went live, and the results.
- How outcomes were tested across relevant groups, and what was done about any disparity.
- Why the model produced this output.
- Who reviewed, challenged or overrode it, and what changed as a result.

Advice needs explaining too. Some AI outputs suggest rather than decide: a recommended route, a best time to work, a next action. When someone follows the suggestion and the result disappoints, the organization may still be asked how the system reached it. Advisory AI needs the same explainability as AI that decides.

Bring these teams in as co-designers. Legal, privacy and security teams are most useful when they help write the evaluation criteria rather than review results at the end. Criteria written in plain language, with business, legal and technical owners, are easier to defend because the people defending them helped write them.

Run AI QA as an operating rhythm

AI QA works best as a recurring cycle, run the way defect organizations already run, rather than a one-time project before launch. Each turn of the cycle produces evidence for the next.

1. **Define** the criteria for each use case with business, QA, legal, privacy and security owners.
2. **Evaluate and select** candidate models against those criteria, including outcome fairness.
3. **Release with evidence**, so every AI experience goes live with a record of what it was tested against.
4. **Monitor** production for drift, regression, inconsistent verdicts and skewed outcomes.
5. **Triage and explain** each escaped defect to its root cause: the data, the retrieval step, the model or the prompt.
6. **Correct and re-prove** against the same baseline, then add the case to the test set.

Treat live pilots as production. A model that touches real users or real decisions carries real risk, whatever it's called. Pilots should run through the same evaluation as everything else.

Choose an assurance layer that outlasts the model. Models will change faster than any QA program. Criteria, test sets and the evidence record should stay constant while the model underneath changes, whether it's a commercial API, an open-weight model or one built in-house.

Criteria, test sets and the evidence record should stay constant while the model underneath changes.

That consistency is also what keeps AI spend in check: when every model is scored against the same bar, teams stop paying frontier-model token prices for work a smaller model does as well.

Where the market stands

AI assurance is a crowded and still-forming category, and most offerings answer one or two of the three questions well. Analysts are still drawing its lines: Gartner published its first Magic Quadrant for AI Governance Platforms in June 2026, separate from its research on AI-augmented software testing (Kovrr). Audit firms entering the space have acknowledged that AI assurance lacks the standardization of financial audits (Financial Times, via Koltin).

Buyers will typically meet five kinds of offering.

Approach	What it does well	What it can leave open
QA and testing services firms extending into AIQA, trust and safety, and AI data services firms	Large teams of experienced reviewers, red teaming at scale, go-live readiness	Point-in-time, people-driven results; black-box review of outputs without visibility into why they were produced; evidence that lives in reports rather than in the system
Developer evaluation and observability tools	Fast scoring, tracing and monitoring for engineering teams	Scores often come from another AI model acting as judge; the explanation of a failure is left to the engineer
AI governance, risk and compliance platforms	Inventories, policies, risk assessments and regulatory mapping	Documentation of controls rather than technical evidence from the system itself
AI security validation	Red teaming, jailbreak and injection testing, runtime guardrails	A secure system can still give a wrong answer; security is one quality attribute, not all of them

Approach	What it does well	What it can leave open
Audit-firm assurance and attestation	Independence and credibility with boards and regulators	Periodic reviews that depend on the evidence the system can produce

These approaches are not mutually exclusive, and large enterprises will use more than one. The question for an assurance program is which one produces the evidence the others rely on.

How Seekr delivers AI QA

Seekr answers all three questions in one system that runs where the enterprise needs it and works with whichever model sits underneath. SeekrGuard evaluates and selects models against the enterprise's criteria. SeekrFlow makes each output explainable by recording where its evidence came from and how it was produced. Together they turn AI QA from a pre-launch checkpoint into a continuous record.

AI QA requirement	How Seekr meets it
Business-defined evaluation criteria	SeekrGuard customizable evaluation criteria: teams define what good looks like for each use case and score every candidate model against it
Decision-ready models and generative AI	The same evaluation approach covers models that rank, price, score and verify, as well as LLMs and agents
Evidence-based model selection	SeekrGuard evidence-based model selection compares frontier, open-weight and small models on the same criteria, so the choice rests on results rather than reputation
Continuous re-evaluation	SeekrGuard centralized evaluation management keeps test sets, criteria and results versioned, so a model or data change is re-scored against the same baseline
Explainability, provenance, attribution and lineage	SeekrFlow attributes each output to the data, content, model and agent behavior behind it, applying provenance and lineage throughout
Root-cause analysis	Attribution separates retrieval failures from evidence-use and reasoning failures, so teams fix the actual cause

AI QA requirement	How Seekr meets it
Outcome fairness testing	SeekrGuard customizable evaluation criteria include outcome measures across groups and segments, scored alongside accuracy and re-run whenever the model or data changes
Adversarial evaluation	Evaluation datasets that include adversarial examples run as part of the standard test set
Contestability and review	The SeekrGuard explainable AI assistant helps reviewers interrogate results, and outputs can be challenged and corrected with the decision recorded
Audit evidence	SeekrGuard keeps a complete audit trail of what was evaluated, the results and the decisions made
Sovereign evidence under your control	SeekrGuard deploys as SaaS or on-premises, and in AWS GovCloud for government; SeekrFlow runs in cloud, on-premises, edge and air-gapped environments. Teams can start in SaaS and move on-premises without changing criteria or losing the evidence record

Right-sizing is part of quality. The cheapest model that reliably meets the criteria is the right one, and that answer changes as models and workloads change. Seekr's approach is to find the best-fit model with SeekrGuard, fine-tune a smaller model to that standard with SeekrFlow if it makes sense, then prove the result with SeekrGuard again. A model is not chosen once. It's proven and re-proven. That's also the answer to runaway token spend.

A model is not chosen once. It's proven and re-proven.

Root-cause analysis in practice. Seekr research on grounded enterprise AI showed that retrieving the right evidence doesn't guarantee a system uses it, breaking a 57.5-point accuracy gap into retrieval, evidence-use and reasoning failures. Output-only testing can't see that difference. Attribution can. [Editor: confirm release status of "Retrieved Doesn't Mean Used" before citing figures.]

Evaluation checklist for AI QA vendors

Use these questions to scope an Enterprise AI Assurance program or to compare AI QA vendors. Each one asks for evidence the vendor can show, not a capability it can describe.

Evaluation: Does it pass?

- ☐ Can we define our own evaluation criteria per use case, or are we limited to a vendor's fixed metrics?
- ☐ Can you evaluate decision models that rank, price, score, and verify, not only LLMs and agents?
- ☐ Can you test for differences in outcomes across groups, and show what was done about them?
- ☐ Can we compare several models, including frontier, open-weight and small models, against the same criteria in one place?
- ☐ When a score comes from an AI judge, can we see and validate how the judge reached it?
- ☐ Are adversarial cases part of the standard test set, or a separate engagement?

Explanation: Why did it produce this?

- ☐ For a single output, can you show the source data, which data, model and steps produced it, and how the data changed along the way?
- ☐ Can you distinguish a retrieval failure from a failure to use retrieved evidence, or from a reasoning failure?
- ☐ Can you reconstruct why one specific decision was made, months after it happened?
- ☐ Can you explain inconsistent results, such as two different verdicts on the same input?
- ☐ For agents, can you show every action taken and confirm it was within authorized boundaries?

Evidence: Can we prove it later?

- ☐ Is there a complete, exportable audit trail of evaluations, results, and decisions?
- ☐ Can a reviewer challenge, override, and correct an output, with the correction recorded?
- ☐ Where are evaluation data, outputs, and audit records stored, and who can access them?
- ☐ Would the evidence hold up to review by our legal, privacy, and security teams?

Control and lifecycle

- ☐ Can we start in SaaS and move on-premises or to a private cloud later, without changing criteria or losing evidence?
- ☐ Do evaluations run in software, so results are repeatable, or do they depend on the availability of a review team?
- ☐ What happens when a provider updates a model we depend on? How quickly can we re-score against our baseline?
- ☐ Is the vendor independent of the models being evaluated?
- ☐ Does the program cover the full lifecycle, from model selection through production, or a single phase?

Conclusion

Enterprises do not need to invent AI QA from scratch. The disciplines that made software and physical products dependable still apply: traceability, defined results, root-cause analysis, regression control, independence, and evidence. What AI adds is the need to show why an output was produced, where its evidence came from, and who held the data along the way.

An assurance program that can answer "does Does it pass, why, and can you prove it?" is one that lets AI move from pilot to production and stay there, and lets the people who have to defend it do so. Some things in life are hard to explain. Your AI shouldn't be one of them.

To see how Seekr evaluates, explains, and documents AI in your environment, visit seekr.com.

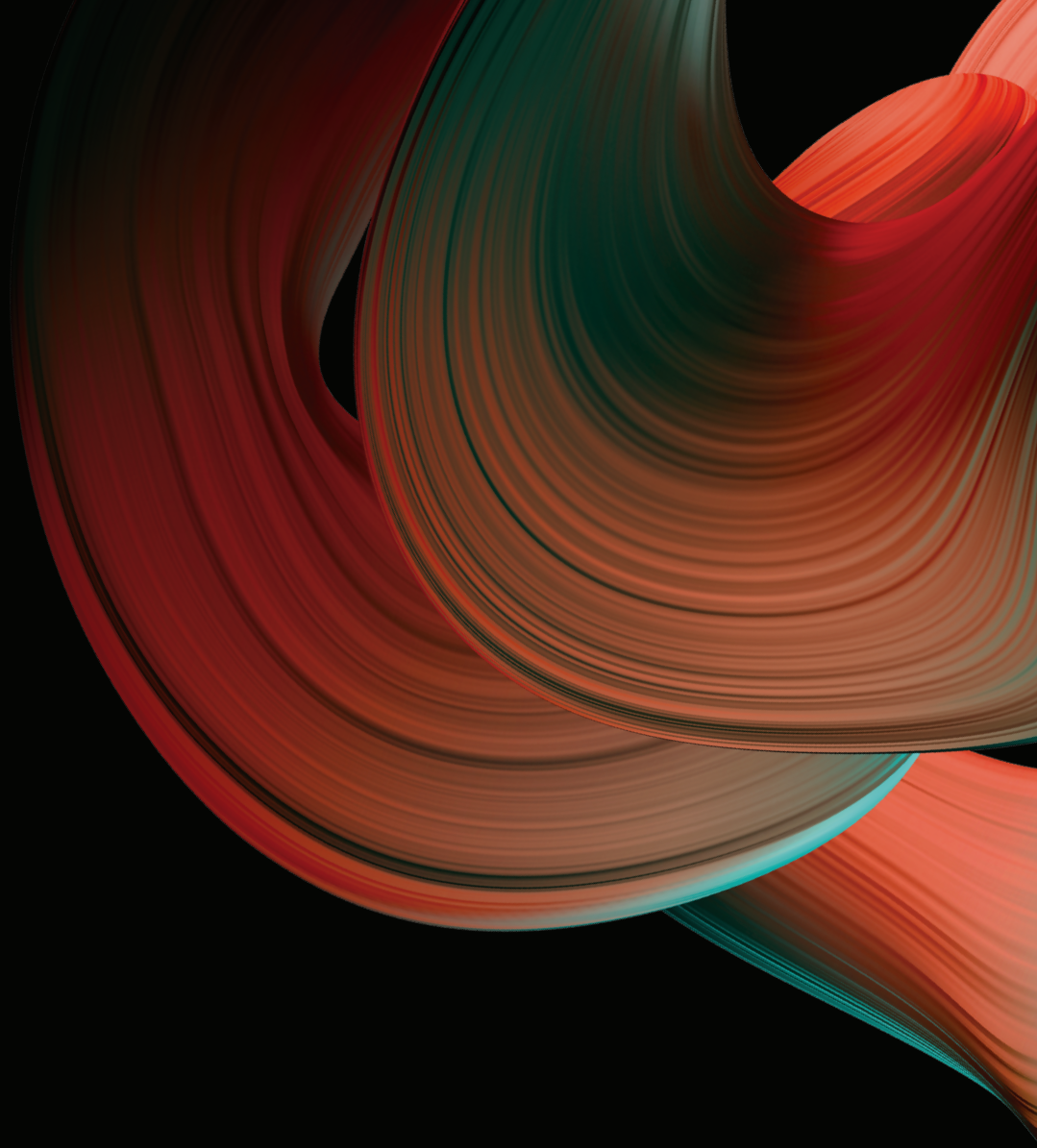
Sources

Cisco Ships Zero Trust for AI Agents, Self-Service Red Teaming, and Agentic SOC Tools at RSAC 2026, Security Boulevard, March 2026

AI Agent & LLM Testing in 2026, Testriq, citing LangChain's 2026 State of AI Agents report

The Top AI Governance Platforms in 2026, Kovrr, August 2026

Big Four Firms Race to Develop Audits for AI Products, Financial Times, June 2025



About Seekr

Seekr is the leader in explainable, defensible AI built for critical decisions in environments that demand accuracy and accountability. Seekr's technology and products help enterprises and government agencies develop and deploy domain-specific large language models (LLMs), vision language models (VLMs) that understand the physical world, edge-based systems and AI agents across any infrastructure, including sovereign deployments. Backed by robust verification and validation tools that surface the provenance, lineage, and intent behind every model decision, Seekr delivers AI that organizations can audit and trust across all modalities. Learn more at **seekr.com.**

© Seekr. All Rights Rese