

Agent Observability: Observe, Evaluate, and Guardrail Your Entire AI Stack

Mitigate AI errors and risks. Track costs
and performance. Build trust and ROI.

The new rules of observability in the AI era

Changing from “What broke?” to “Can I trust my AI systems?”

For the past decade, observability was defined by the transition to cloud-native architectures. As organizations moved from monoliths to microservices, the industry standard focused on understanding system health through the “Golden Signals” of latency, traffic, errors, and saturation. These tools were designed to tell us what broke and where, providing a reliable map for the traditional software lifecycle. However, the rise of AI has fundamentally rewritten this operating model. We are no longer just monitoring static code — we are overseeing autonomous agents that reason, plan, and take multi-step actions to complete tasks. This shift from deterministic, rule-based applications to complex, adaptive agentic systems has created a trust gap that traditional observability was never designed to bridge.

As enterprises rush to invest in AI to enhance automation and improve decision-making, they face an impossible choice: speed vs. control. However, this speed has given rise to “slop,” which is inauthentic, inaccurate, and potentially harmful content that threatens the integrity of business workflows. Because AI agents are inherently non-deterministic, they introduce new failure modes, including hallucinations, flawed reasoning, and policy drift. Without proper governance, organizations remain vulnerable to prompt attacks and data leaks. The sheer complexity of AI infrastructure — spanning GPUs, vector databases, and LLMs — often leaves root causes buried. AI is also expensive to run. A single interaction can consume tens of thousands of tokens, and without visibility into tokenomics, runaway costs can quickly destroy the business case for AI.

The core question of observability has shifted from “Is my infrastructure healthy?” to “Can I trust the systems and agents that are helping me efficiently run the business?” To scale AI from experimental pilots to enterprise-grade production, teams must adopt a new framework that moves beyond traditional telemetry. This requires a holistic approach that integrates:

- **Performance metrics** to track agent and infrastructure uptime and overall health
- **Quality metrics** (or evals) to define and measure what “good” looks like in an output; allowing you to benchmark whether additional AI spend produces a better output or simply a more expensive one
- **Security metrics** to mitigate risks like prompt injection and personally identifiable information (PII) leakage
- **Cost metrics** to ensure clear ROI

Scaling AI is an operational hurdle. The cost of inaction is far too high — ranging from production incidents and brand damage to regulatory exposure. By implementing real-time guardrails and high-accuracy evaluations, organizations can move from reactive monitoring to proactive control. This approach ensures that agents behave as expected and optimizes resource allocation, preventing the inefficiencies that drain budgets.

Observability is the key to ensuring that your business remains secure, compliant, and profitable as your AI agents scale in this new era. It's time to move beyond traditional observability and build a leading observability practice that prioritizes reliability, safety, and trust.

The four pillars of agent observability

To navigate the complexities of the agentic era, organizations need a unified strategy that moves beyond traditional monitoring. A unified agent observability strategy can be broken down into four pillars:

1. Observe the entire AI stack

The AI infrastructure stack is far more complex than traditional cloud-native architectures, primarily because it introduces volatile failure modes that are difficult to isolate. Managing GPUs, vector databases, and LLMs means accounting for hardware-level events such as voltage dips on a chip, which serve as early warning signs of an impending crash. Detecting these signals, correlating them to specific agent failures, and responding at machine speed is significantly more challenging than managing traditional servers. To maintain reliability, you need end-to-end visibility that traces dependencies across every layer. This allows you to track SLOs, diagnose failures in real time, and ensure infrastructure bottlenecks do not compromise the performance of your agents.

2. Evaluate and improve agent behavior

AI systems are very different from traditional software. Traditional software errors usually come from edge cases and race conditions, which are hard to reproduce but can be fixed once found. AI, on the other hand, acts like a permanent race condition. It's designed to be creative and autonomous, so running the same AI agent multiple times rarely produces the same result or follows the same path. This makes debugging AI agents much more challenging — it's like software development on hard mode.

Additionally, AI agents and models are adaptive and unpredictable. To ensure they remain reliable, you must evaluate their behavior continuously — from initial development through to production. This means testing prompts and responses for hallucinations, bias, toxicity, and relevance. By integrating high-accuracy evaluations into your workflow, you can move from hoping your agents work correctly to knowing they are performing as intended. This bridges the AI trust gap.

Continuous evaluation is key because AI adds complexity beyond traditional infrastructure monitoring. You need observability that tracks not only uptime and latency but also output quality, security risks, and resource use. This visibility helps teams catch issues early, maintain trust, and confidently scale AI projects that deliver real business value.

3. Optimize token usage and costs

In the age of AI, token consumption is the new currency of operational efficiency. A single agent interaction can consume tens of thousands of tokens, and without granular visibility, costs can spiral out of control. But not all tokens are created equal — some are pure waste. This happens when agents read about tools they will never use, or when irrelevant context is stuffed into requests. Sometimes agents get stuck in loops trying to solve problems they have no chance of fixing, like chasing unrealistic goals set by operators. These common missteps drive runaway costs that many teams face daily when working with AI.

It's critical to have deep insights into token usage so you can attribute spend to specific agents, workflows, or business outcomes. By correlating token consumption with evals, you can optimize your resource allocation, ensure a clear ROI, and prevent the runaway costs that often stall AI projects before they reach full-scale production.

4. Block harmful and inaccurate actions with guardrails

Governance is the final, critical piece of the puzzle. Without real-time guardrails, organizations are vulnerable to security and privacy risks like prompt injections, PII leakage, and tool misuse that can erode customer trust and trigger regulatory non-compliance. Dynamic, runtime guardrails that block inaccurate or harmful content before it ever reaches the end user are essential. By training custom business logic into a small language model (SLM), you can enforce safety and compliance policies at the speed of your agents, ensuring that your agents remain a secure and productive asset.

Reflection questions

- How are you currently monitoring the “hidden” costs of your AI stack, such as token consumption and GPU idle time?
- Do you have a unified framework that tracks quality (hallucinations/bias) alongside infrastructure performance, or are these metrics siloed in different tools?
- What is your current process for updating guardrails when a new security or compliance risk is identified?

Observing the entire AI stack

Modern generative AI applications have evolved from simple, deterministic tasks into complex, adaptive, and interdependent agentic workflows. To succeed, you cannot rely on monitoring the application layer alone. A truly resilient AI strategy requires holistic visibility across the entire stack — from your agents and models to your vector databases, proxies, and other specialized infrastructure.

From fragmented signals to full-stack intelligence

When a user request triggers a multi-agent workflow, it initiates a chain of dependencies: orchestration logic, vector database queries, model-serving endpoints, and tool calls. If one link falters, the entire process slows or fails. While traditional infrastructure monitoring excels at tracking core systems, the real challenge lies in monitoring and correlating new types of hardware and software signals unique to AI technologies — like vector databases, orchestration frameworks, and model-serving layers. To maintain reliability, you need APM-like performance metrics that allow you to trace interactions, map dependencies, and analyze workflows across the entire AI stack. With 100% visibility into agent interactions, you can pinpoint the root cause of errors in complex, multi-agent flows in real time, rather than discovering them through user complaints.

Dynamic assessment: The new standard for quality

Unlike traditional machine learning (ML) models evaluated on static metrics like accuracy or F1-scores, agents require more dynamic, multi-step assessments — such as end-to-end task completion rates, hallucination detection, and safety checks. Without efficient tools, teams face a tough choice: rely on manual reviews that don't scale or use costly LLM-as-judge evaluations.

LLM-as-judge refers to using a foundation model to automatically evaluate the outputs and actions of an AI agent. This approach transforms raw, unstructured telemetry data from agent interactions into concrete, quantifiable metrics. These metrics provide a scalable way to observe agent behavior and output quality across thousands of daily interactions, without the overhead of manual review or expensive human-in-the-loop processes. This method is well proven, but teams often struggle with the cost of scanning so much data with an LLM, and the multi-second latencies which prevent them from acting on these signals in real time.

Using a fine-tuned small language model as judge can replace these inefficient LLM-driven processes by providing scalable, high-accuracy evaluations. By integrating automated evals for hallucinations, bias, toxicity, and sentiment directly into your observability pipeline, you gain deep input/output visibility that ensures your agents remain reliable without breaking your budget or user experience.

Optimizing the “secure AI factory”

Infrastructure underpins every successful agentic AI deployment. High-availability setups — relying on GPUs for inference, vector databases for RAG, and orchestration for multi-agent logic — are resource-intensive. Without deep visibility, teams often face significant waste due to idle resources and over-provisioning, which can lead to unnecessary costs. This is especially true of owned infrastructure, where attribution challenges can be even more nuanced. Teams struggle to know which agents are taking the most resources and decide which should be continued or throttled. These questions are front and center when you don’t have the convenience of a API-key based attributions in your monthly AI bill.

The key is to bridge the gap between hardware and software. Whether you are utilizing a “secure AI factory” — combining hosted hardware with an observability solution — or managing a multi-cloud environment, it’s important to have a unified view of your infrastructure without vendor lock-in or surprise overage charges. This integration allows you to:

- **Develop AI applications:** Manage the massive amounts of information processing and training required to develop AI applications with a centralized platform for designing, testing, and deploying your models.

- **Gain insight into potential bottlenecks:** See what is causing slow responses and gain clear visibility into API call failures or incomplete responses.
- **Detect underutilization:** Identify idle resources like GPU capacity or inefficient KV cache performance that result in slower inferencing.
- **Optimize workloads:** Correlate technical performance with business impact to ensure your infrastructure is scaled to meet demand without burning your budget.

Deep visibility is your best defense against the operational risks of scaling AI. By linking hardware health, model performance, and agentic behavior into a single, correlated view, you ensure that your AI stack is not just operational, but optimized. Building a system that is as efficient as it is trustworthy is the path to moving from experimental pilots to enterprise-grade production.

Reflection questions

- Can you currently trace a single user request across your orchestration layer, vector database, and model-serving endpoints?
- Are you able to pinpoint whether a latency spike is caused by model inference, network congestion, or GPU resource exhaustion?
- How much of your AI budget is currently lost to over-provisioning or idle GPU capacity that goes undetected?

Mastering agentic performance with tokenomics and trust

Getting observability right in the agentic era means tracking more than just uptime. Because AI agents are non-deterministic and adaptive, you need a unified framework that balances performance, cost, and quality to ensure your agents are performing as intended for their specific business purpose.

The four pillars of agent metrics

To gain a complete picture of how your AI applications behave, you must track indicators across four critical categories:

- **Objective accuracy:** Measures of correctness, such as F1 scores or task-specific benchmarks, that validate the agent's reasoning and logic.
- **Human-centric quality:** Semantic quality, relevance, and informativeness, evaluated continuously and affordably by SLM judges. These can provide real-time, affordable insights into tone, customer sentiment, relevance, success rate, and accuracy — helping you quickly spot issues and make confident decisions on model selection and cost.

- **Operational efficiency:** Latency, throughput, error rates, and token usage. This category tracks the “tokenomics” of your agents, helping you detect spikes caused by inefficient prompts or the overuse of premium models.
- **Safety and fairness:** Indicators of toxicity, bias, or prompt injection risks, showing where agent behavior drifts into unwanted or non-compliant patterns.

Evals as a team sport: The agent development lifecycle

Evaluation is not a one-time event; it is a continuous loop. To build trust, you must integrate evals into your CI/CD pipelines, running them both before and after shipping. This allows your team to gather expert feedback on real-world performance and iterate on prompts with precision.

By leveraging “LLM-as-a-judge” frameworks, you can automate these assessments at scale. However, accuracy is paramount. Untuned, an LLM acting as a judge typically achieves around 70% accuracy — so about one in three evaluation scores may

be incorrect. Worse, you won't know which ones. The only way to improve this is through iterative training and refinement. By incorporating SME feedback to tune prompts against an annotated dataset, teams can significantly boost eval judge accuracy. Further, fine-tuning a SLM enables running evaluations at machine speed and very low cost, while maintaining the high accuracy achieved in previous tunings.

Using these high-accuracy, autotuned metrics allows you to move away from expensive manual reviews or unreliable open-source setups. As the system learns and improves with SME input, your evaluation engine becomes more precise over time.

The “tokenmaxxing” problem

As agent adoption grows, token consumption often skyrockets. However, many teams struggle to explain which agent workflows, models, or infrastructure components are driving that spend. This leads to “tokenmaxxing” — a state where high activity is mistaken for high impact. Because they lack visibility into quality and business impact, most managers assume that more AI use automatically means more value, equating AI activity with success. This naive thinking stems from an inability to quantify the true impact of AI usage. Some workflows generate meaningful, production-ready code, while others consume enormous amounts of tokens without improving what ships.

Finance teams see the invoice, while engineering teams see the sprint activity, but connecting the two remains a challenge. By correlating token consumption with agent behavior, deployment speed, and business outcomes, organizations can identify which agents deliver real ROI versus those that consume resources without impact. This visibility enables optimization of inference workloads and helps prevent the “noisy neighbor” effect, where one misconfigured agent slows down the entire production environment.

Correlating token usage with agent behavior and infrastructure

Tokenmaxxing happens when AI agents consume excessive tokens without delivering proportional value, driving up costs and risking runaway spend. The key to solving tokenmaxxing is gaining real-time visibility into how tokens are used across agents, workflows, and the underlying infrastructure.

By correlating token consumption with agent behavior and infrastructure telemetry like GPU utilization, memory, and vector database performance, you can pinpoint inefficiencies and bottlenecks. For example, an agent that is allowed to search large volumes of unrelated data can quickly inflate its token consumption without any meaningful improvements.

Tracking token spend alongside quality signals such as code quality, policy adherence, and context relevance helps teams identify which agents provide true ROI and which are token drainers. This insight enables smarter routing of workloads to the most cost-effective models and hardware, preventing overuse and optimizing inference workloads.

Real-time dashboards that unify token usage, agent performance, and infrastructure metrics help stop runaway agents before they run up the bill. This approach transforms AI from a costly experiment into a reliable, enterprise-grade asset.

Reflection questions

- Are your evals static (e.g., simple accuracy), or are they dynamic and tuned to your specific business use case?
- How much is your team currently spending on manual reviews or LLM-as-a-judge setups to validate agent actions?
- Can you attribute token spend to specific agents, business workflows, or customer journeys, or is your AI spend a “black box”?

Securing the agentic frontier

AI systems introduce a new class of risk that traditional security tools were never built to handle. In an environment where agents autonomously reason, call tools, and interact with sensitive enterprise data, the attack surface has expanded dramatically. The real danger isn't just hallucinations or bias — it's that an AI agent could accidentally drop your entire database, delete critical containers, or overwrite essential files. (For example: Would you trust an AI agent with your only copy of the family photos?) Beyond reputational damage from bias and toxicity, sophisticated threats like prompt injection and data exfiltration put sensitive PII, PHI, and PCI data at risk. Left unmanaged, these vulnerabilities impact far more than system performance — they shake the very foundation of customer trust.

The complexity of the agentic stack

Modern AI apps rely on complex chains of LLM calls and retrieval-augmented generation (RAG) pipelines. Each step, from the initial prompt to final output, is a potential failure or attack point. Risks like context poisoning, where harmful inputs skew the agent's understanding, and agent hijacking, where attackers take control to cause damage, are real threats. These can lead to corrupted outputs, data leaks, or unauthorized tool use. To protect these systems, you need end-to-end visibility that tracks security signals alongside performance metrics.

You can't secure what you can't see. Monitoring the full interaction path helps spot where malicious inputs or biased responses originate and lets you act fast. Comprehensive observability also detects AI-specific risks like prompt injections, data exfiltration, and tool misuse, keeping your AI stack reliable and secure throughout its lifecycle.

This approach aligns with [OWASP's](#) agentic AI security guidelines, emphasizing continuous monitoring, threat detection, and governance to build trustworthy AI systems that scale with confidence.

Real-time mitigation with runtime guardrails

Observability is the necessary precursor to effective protection. By integrating real-time guardrails, you can move from reactive logging to active defense. By using high-accuracy, low-latency guardrails — powered by specialized SLMs — you can intercept and block harmful actions in milliseconds. This allows you to:

- **Detect risks in real time:** Automatically identify prompt injections, tool misuse, and data leakage before they reach the end user.
- **Mitigate threats instantly:** Enforce centralized policy controls that block undesirable agent behavior without hard-coding safety logic into every individual agent.
- **Ensure compliance:** Provide the audit trails and traceable interactions that compliance teams need to verify that models adhere to legal, regulatory, and internal safety standards.

Connecting safeguards to business value

Governance is more than just a security requirement — it is a business enabler. By preserving trust and validating the safety of your AI actions, you provide the evidence needed to move from pilot to production. Observability connects these safeguards back to business impact, demonstrating how reliability, accuracy, and security shape the user experience. When you can prove that your agents are protected against attacks and compliant with safety standards, you gain the confidence to scale your AI services and turn your security infrastructure into a competitive advantage.

Reflection questions

- How are you currently validating that your agents are not leaking PII, PHI, or PCI data during multi-turn conversations?
- If an agent were to be compromised via prompt injection today, how long would it take for your team to detect and block the attack?
- Do you have a centralized policy server that allows you to update security guardrails across all agents instantly, or is safety logic hard coded into your applications?

The future of AI observability: Proactive, unified, and trustworthy

AI leaders face an existential choice: slow down development and risk falling behind, or cede control and risk failure. But you shouldn't have to choose. By creating high-accuracy, custom evals that become runtime guardrails, you can build and enforce high standards of performance at machine speed — without sacrificing ROI or speed.

Proactive: Speed without compromise

True speed means having a platform that connects development and production while keeping performance, uptime, and security steady. Separate evaluation systems and security checks block the flow of data needed to turn pre-production insights into real-time protections. A unified platform fixes this by applying the same safety and quality rules from development to live environments, ensuring reliability and compliance without compromise. This helps you deliver faster, safer, and more reliable deployments with confidence.

Unified: Aligning teams to business outcomes

Effective AI observability is a team sport. From AI leaders (CIOs, CTOs) and business partners (CISOs, architects) to end users (data scientists, engineers), everyone must work from a shared source of truth.

When you correlate technical telemetry — like GPU spikes or storage bottlenecks — directly with business KPIs like transaction success rates and user retention, you break down silos. This shared context allows teams to prioritize fixes based on business impact, manage production costs, and accelerate delivery. When your visibility is tied to business outcomes, you stop guessing and start shipping.

Trustworthy: Scaling with confidence

As AI environments scale, blind spots become liabilities. Small deployments may survive with fragmented views, but enterprise-grade AI demands a unified strategy. By weaving security, privacy, and compliance guardrails into the same fabric that tracks uptime and cost, you eliminate the gaps that lead to failure.

Reliable infrastructure visibility is the operational anchor for enterprise AI. A trusted approach ensures that as your AI footprint grows, your business remains secure, compliant, and profitable. This method is how you scale — not by sacrificing performance for safety, but by building a system that is as efficient as it is trustworthy.

Reflection questions

- Do you have agents in production today, and what is your current strategy for moving from pilot to enterprise-grade scale?
- Are your evaluation, monitoring, and security teams working from a shared source of truth, or are they operating in silos?
- What is the primary business outcome (cost reduction, compliance, or user experience) that is driving your need for better AI visibility right now?

Ready to turn your AI data into doing?

Stop guessing about your AI performance and start scaling with certainty. Discover how [Splunk Agent Observability](#) helps you bridge the trust gap, optimize your costs, and secure your agentic workflows.



Splunk, Splunk>, Data-to-Everything, and Turn Data Into Doing are trademarks or registered trademarks of Splunk Inc. in the United States and other countries. All other brand names, product names, or trademarks belong to their respective owners. © 2026 Splunk LLC. All rights reserved.

26_CMP_ebook_Agent-observability-Observe-evaluate-and-guardrail-your-entire-AI-stack_v4