EXPLAINER NO. 01 · AGENT OBSERVABILITY

The Agent Observability Gap

Why agents that look healthy in a playground go sideways the moment they meet real users, and what a useful observability stack for agent-shaped systems has to cover.

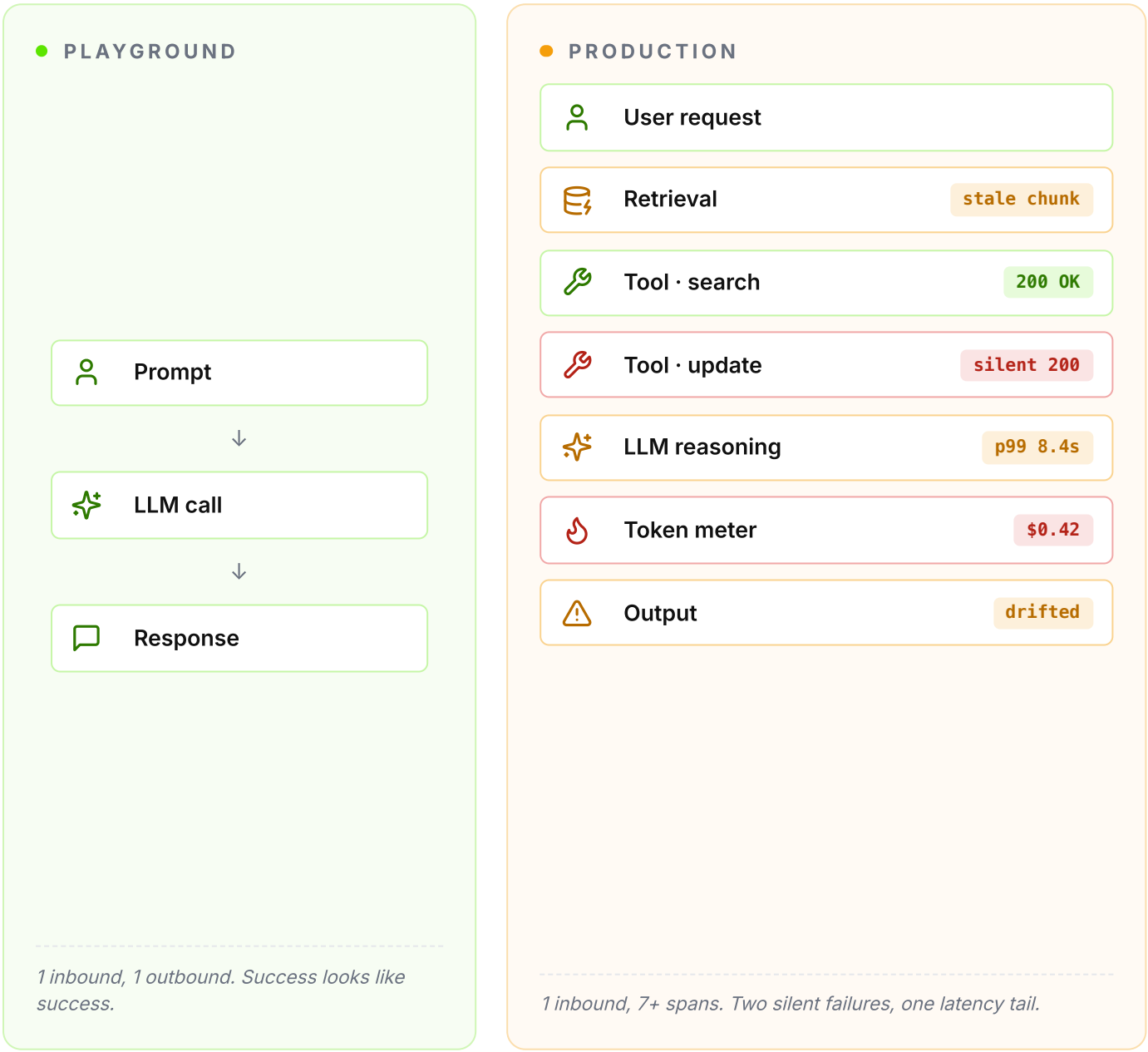
51% of teams already run AI agents in production.[5]

Most inherited observability built for request-and-response web apps, a shape their agent stopped having a long time ago.

PLAYGROUND VS PRODUCTION

Why the demo lies

A single user request in a playground is a one-shot prompt-and-response. The same request in production fans out into retrieval, tool calls, and retries, and most of those steps never surface in traditional logs.



Takeaway. The same agent that looked healthy in the playground now depends on retrieval quality, tool correctness, retry loops, and spend per request. Any one of those can degrade without throwing an error, and without traces you'll find out from a user.

FIVE FAILURE MODES

Failures that only appear in production

Agent failures rarely raise exceptions. These five are the ones most teams meet first, and the five that traditional logging is least equipped to spot.

Retrieval Failure

Your RAG pipeline returns stale, wrong, or irrelevant chunks, the model answers confidently anyway.

APM sees a successful vector query. It can't judge whether the retrieved context was actually right.

Tool-Call Failure

The agent picks the wrong tool, passes bad parameters, or the tool fails silently mid-chain.

HTTP 200 hides semantic errors. Logs don't know which tool the agent should have called.

Latency Spikes

p50 looks fine. p99 is 12 seconds on the exact flows your power users hit most.

Request-level metrics average away the tail. Per-span latency across an agent chain is invisible.

Cost Blowouts

A reasoning loop burns 40k tokens on a single request. One user racks up $200 before lunch.

Infra dashboards track CPU, not tokens. Cost per trace, per user, per model is not a native concept.

Output Drift

Same input, different output. Quality silently degrades after a model upgrade or prompt tweak.

There's no 'error' to log. You need evals and historical replay to catch behavioral regressions.

WHAT EACH TIER ACTUALLY SEES

Why standard logging breaks down

APM and log aggregation were designed for stateless request-and-response apps. Inside one agent run there are seven or more spans, and the useful signal lives between them. Here is what each tier can observe.

Traditional APM

Built for stateless HTTP

20% of an agent run

SEES

- HTTP request and response
- Top-level status codes
- Process CPU and memory

MISSES

- Per-step reasoning
- Tool inputs, outputs, decisions
- Silent semantic failures on HTTP 200

Log aggregation

print() plus aggregation

50% of an agent run

SEES

- Whatever you remembered to log
- Error messages and stack traces
- Free-text breadcrumbs

MISSES

- Causal chain between steps
- Prompts, completions, token cost
- Drift over time

Agent observability

Built for stateful graphs

100% of an agent run

SEES

- Full span tree with parent and child causality
- Prompts, completions, tool I/O, token cost per step
- Eval scores, drift over time, replay

MISSES

Nothing in the intended scope.

Takeaway. APM and logs are not wrong, they are just scoped for a different shape. Stateful, multi-step agents need span-level visibility, eval signal, and replay as primitives, not add-ons.

SIX LAYERS, NOT ONE

What a working observability stack covers

A useful stack for agent-shaped systems has to do all six at once: traces, evals, cost, latency, drift, and replay. Miss any one and you're debugging blindfolded on the others.

1 Traces

Full reasoning path and tool chain for every agent run.

89% of agent teams have adopted dedicated observability.[6]

2 Evals

LLM-as-a-judge scores, golden datasets, regression checks.

#1 enterprise blocker is quality, evals make it measurable.[8]

3 Cost

Token spend attributed per model, per app, per user.

$3.5B to $8.4B LLM API spend jump between late 2024 and mid 2025.[13]

4 Latency

p50, p95, p99 broken down by span across the whole chain.

42% to 54% YoY jump in AI monitoring adoption (New Relic 2025 Forecast).[9]

5 Drift

Behavioral change detection over time and across versions.

5.5% of orgs see real financial return from AI, drift is a silent killer.[7]

6 Replay & Debug

PRINT, TICK, SHARE

Is your LLM app production-ready?

Five yes-or-no questions, each tied to one of the pillars. Tick the boxes with a pen, then count the yeses and compare against the rubric.

- [ ] **01 Can you trace every agent step?**
  Without a full trace, debugging is guesswork and fixes are gambles.

- [ ] **02 Can you see exactly where it fails?**
  Retrieval, tool calls, and prompts all fail differently. You need to see which one broke.

- [ ] **03 Can you inspect latency and cost per step?**
  One slow tool or one runaway loop can tank UX and burn budget. Per-span visibility prevents both.

- [ ] **04 Can you compare runs over time?**
  Prompts and models change. Evals and replay tell you whether quality held or slipped.

- [ ] **05 Can you catch drift before your users do?**
  Silent degradation is the most expensive bug. Drift detection turns it into an alert, not a postmortem.

MATURITY RUBRIC

5 / 5

Mature coverage

Traces, evals, cost, latency, drift, and replay are all in place. You can debug on evidence.

3, 4

Clear gaps

You can debug most failures, but one or two classes still rely on guesswork.

0, 2

Building from scratch

Expect regressions, silent cost leaks, and long debugging loops until the basics land.

REFERENCES (14)

What this piece stands on

Grouped by evidence class, ordered by epistemic weight. Standards and primary research sit above surveys and analyst commentary; vendor and context material frame the category but don't carry the data.

STANDARDS

[01] **OpenTelemetry**. OpenTelemetry, Semantic Conventions for Generative AI (spans, attributes, events). https://opentelemetry.io/docs/specs/semconv/gen-ai/

PRIMARY RESEARCH

[02] **Berkeley FCL (Patil et al., arXiv:2407.00121)**, ICML 2025. Berkeley Function-Calling Leaderboard, paper and live leaderboard. https://gorilla.cs.berkeley.edu/leaderboard.html

[03] **Vectara**, 2025, updated. Hallucination Evaluation Model Leaderboard (HHEM), factual-consistency rates across frontier models. https://github.com/vectara/hallucination-leaderboard

[04] **Anthropic**, 2025. Measuring AI Agent Autonomy in Practice, real usage data and auto-approve curves. https://www.anthropic.com/research/measuring-agent-autonomy

INDUSTRY SURVEYS

[05] **LangChain**, 2024. State of AI Agents 2024 (1,300+ respondents). https://www.langchain.com/stateofaiagents

[06] **LangChain**, 2025. State of Agent Engineering 2025 (1,340 responses). https://www.langchain.com/state-of-agent-engineering

[07] **McKinsey**, Nov 2025. The State of AI, Nov 2025 (1,993 companies). https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai

[08] **Langbase**. State of AI Agents, 3,400+ builders in 100+ countries (observability is the #3 cited blocker at 50%). https://langbase.com/state-of-ai-agents

[09] **New Relic**, 2025. 2025 Observability Forecast, 1,700+ practitioners on AI monitoring adoption. https://newrelic.com/resources/report/observability-forecast/2025

ANALYST REPORTS

[10] **Gartner**, Jun 25, 2025. Gartner predicts more than 40% of agentic AI projects will be canceled by end of 2027. https://www.gartner.com/en/newsroom/press-releases/2025-06-25-gartner-predicts-over-40-percent-of-agentic-ai-projects-will-be-canceled-by-end-of-2027

[11] **Forrester**, May 29, 2025. Why AI Agents Fail (And How To Fix Them), practical framework for agentic failure modes. https://www.forrester.com/report/why-ai-agents-fail-and-how-to-fix-them/RES183446

VENDOR / CATEGORY

[12] **Datadog**, 2025. Datadog expands LLM Observability with capabilities for agentic AI (category legitimacy). https://www.datadoghq.com/about/latest-news/press-releases/datadog-expands-llm-observability-with-new-capabilities-to-monitor-agentic-ai-accelerate-development-and-improve-model-performance/

[13] **MorphLLM**. LLM cost optimization, industry aggregated spend growth. https://www.morphllm.com/llm-cost-optimization

INDUSTRY CONTEXT

[14] **Andreessen Horowitz**, 2026. Big Ideas 2026, Part 1, Aubakirova on agent-native infrastructure becoming table stakes. https://a16z.com/big-ideas-in-tech-2026/