

Deep Research: Multi-Agent Deployment Auditing for EU AI Act Compliance

1. Executive Summary

The transition from single-model, monolithic artificial intelligence applications to multi-agent, compound AI systems represents a fundamental architectural paradigm shift. As enterprises increasingly deploy orchestrations using frameworks such as LangGraph, CrewAI, AutoGen, and Google's Agent Development Kit (ADK), the complexity of ensuring and proving regulatory compliance has grown exponentially. The "AI Trace Auditor" project currently serves a critical function for single-agent systems, successfully mapping linear execution traces to the European Union Artificial Intelligence Act (EU AI Act) and the NIST AI Risk Management Framework (RMF). However, the market signal indicates that this linear approach is rapidly becoming obsolete.

Recent industry analysis, including the Sequoia Capital State of AI Report (2025/2026) and McKinsey surveys, reveals that 79% of organizations have adopted generative AI, and multi-agent systems operating "under the hood" are projected to become the dominant deployment pattern.¹ Despite this widespread adoption, a profound governance gap exists: while 88% of organizations regularly use AI, a mere 7% report that AI is fully deployed and integrated across the enterprise, largely due to trust, security, and governance bottlenecks.⁵ Compliance and legal teams are actively issuing Requests for Proposals (RFPs) for "Agentic AI Compliance" solutions capable of verifying behavioral safety, outcome integrity, and adherence to the EU AI Act across autonomous multi-step workflows.⁷ The market demands solutions that can parse non-deterministic agent routing, tool usage, and inter-agent communication, making the extension of the AI Trace Auditor an urgent and highly viable strategic imperative.

The core challenge lies in the nature of multi-agent observability. Unlike a traditional REST API that returns a deterministic response, an agentic system reasons, plans, retrieves data, and spawns sub-agents, creating a Directed Acyclic Graph (DAG) of execution paths.¹⁰ Existing compliance tools—ranging from static policy registries to basic LLM observability platforms—fail to capture the granular, hierarchical telemetry required to attribute liability when one agent's non-compliant behavior is triggered by another agent's instruction. Furthermore, the enforcement of the EU AI Act, which enters its most stringent phases leading into 2026, imposes severe penalties (up to €35 million or 7% of global turnover) for failures in record-keeping (Article 12) and supply chain accountability (Article 25).⁸

The definitive build/don't-build decision is: **Build**. The market for AI agent observability and evaluation is projected to grow from \$0.55B in 2025 to \$2.05B by 2030, driven explicitly by enterprise compliance requirements.¹³ To capture this value, the AI Trace Auditor must be

fundamentally re-architected. The data model must evolve from a linear sequence of spans into a hierarchical, graph-based engine capable of ingesting OpenTelemetry (OTel) standard conventions for agents, computing bottom-up penalty propagation for compliance scoring, and isolating liability at the node level. This report exhaustively details the trace standards, regulatory interpretations, competitive gaps, and specific data model extensions required to build an enterprise-grade multi-agent compliance auditor.

2. Trace Standards Landscape

To audit multi-agent systems, the AI Trace Auditor must ingest and normalize telemetry that accurately reflects complex delegation, tool use, and memory access. The observability ecosystem is currently undergoing a massive standardization effort to accommodate these requirements, moving away from proprietary logging toward unified OpenTelemetry (OTel) conventions.

2.1. OpenTelemetry (OTel) Semantic Conventions for Agentic Systems

The OpenTelemetry community is actively developing semantic conventions specific to generative AI agentic systems to prevent ecosystem fragmentation. Key proposals, notably Pull Requests and Issues #2664, #2665, #2883, and #2112, define the structural metadata required for multi-agent observability.¹⁴ The standard divides multi-agent execution into distinct conceptual namespaces:

- **Tasks (`gen_ai.task.*`):** Represents the minimal trackable unit of work or objective. Attributes define high-level goals, task hierarchies (parent/subtask relationships), and lifecycle states (created, started, ended). This namespace is critical for compliance, as it captures the *requester context*—identifying whether a human, a system role, or an external request initiated the task.¹⁵
- **Actions (`gen_ai.action.*`):** Captures the non-deterministic execution paths, including LLM completions, external API requests, and vector database queries.¹⁴
- **Agents (`gen_ai.agent.*`):** Defines the autonomous entities orchestrating the actions. Crucial attributes for identity attribution include `gen_ai.agent.id` (unique identifier), `gen_ai.agent.name` (human-readable role, e.g., "Compliance Reviewer"), and `gen_ai.agent.description`.¹⁸
- **Teams (`gen_ai.team.*`):** Represents dynamic groups of agents collaborating on shared goals, structuring the distribution of roles and inter-agent communication.¹⁴
- **Session Correlation (`session.id`):** Proposed in Issue #2883, this attribute replaces the limited `gen_ai.conversation.id` to correlate multi-step, multi-agent workflows across distributed systems, providing a stable boundary for end-to-end trace grouping.¹⁶

Furthermore, OTel distinguishes between span kinds for agents: `CLIENT` is utilized for remote agent services (e.g., AWS Bedrock Agents accessed via network), while `INTERNAL` is utilized for in-process agent execution frameworks (e.g., LangChain, CrewAI).¹⁸

2.2. Langfuse: Hierarchies and Identity Attribution

Langfuse addresses multi-agent architectures by organizing data into nested Trace and Span hierarchies.¹⁹ In frameworks like LangGraph, nodes represent units of work (e.g., Python functions). Langfuse utilizes a `CallbackHandler` passed into the graph's execution configuration to automatically capture these internal state transitions as discrete sub-spans.²⁰

For agent identity attribution, Langfuse relies on attribute propagation. Because an execution trace might cross multiple process boundaries (e.g., from a Python orchestrator to a Go-based tool executor), Langfuse uses the `propagate_attributes()` context manager.²¹ This function ensures that metadata tags—such as the agent's name, the crew it belongs to, and its specific environment—are inherited by all child observations (spans and generations).²¹ Without this explicit propagation, trace-level attributes are overwritten by downstream agents, destroying the audit trail required to prove which agent initiated a specific tool call.²² Langfuse also leverages deterministic Trace IDs generated via a seed (e.g., `langfuse.create_trace_id(seed=external_request_id)`) to reliably link external compliance incidents to internal multi-agent execution graphs.²³

2.3. Arize Phoenix: Agent Graph and Path Visualization

Arize Phoenix addresses multi-agent observability by abstracting granular span data into Directed Acyclic Graphs (DAGs), visually representing the logical flow and handoffs between agents.²⁴ This is powered by the OpenInference standard, which mandates specific metadata attributes on spans to define nodes and edges.

To construct the agent graph, Arize requires the `graph.node.id` attribute (a unique identifier for the component) and recommends the `graph.node.parent_id` attribute to explicitly define hierarchical delegation (e.g., an orchestrator delegating to a researcher).²⁴ Frameworks with built-in support emit these natively: AutoGen handles transitions via `_handoffs` tracking; LangGraph uses `metadata.langgraph_node` and `metadata.langgraph_step`; and CrewAI injects agent roles directly into the trace.²⁴ When auditing, the AI Trace Auditor must parse these specific OpenInference graph attributes to reconstruct the DAG and identify circular delegation loops or unauthorized handoffs.

2.4. Google Agent Development Kit (ADK) Telemetry

Google's ADK is a purpose-built framework for multi-agent orchestration that utilizes a strongly typed schema approach.²⁵ ADK agents are defined by specific execution patterns—`SequentialAgent`, `ParallelAgent`, and `LoopAgent`—and communicate via a standardized Agent-to-Agent (A2A) gRPC protocol.²⁶

ADK inherently relies on OpenTelemetry, sending trace data via the OTLP API. Because ADK enforces strict JSON-schema compliance for all inputs, outputs, and memory states, the resulting telemetry is highly structured.²⁵ This structured state management is a significant advantage for compliance auditing, as the auditor can programmatically verify if an agent

accessed a shared global context memory or restricted local memory without relying on fragile string parsing.²⁵

2.5. Model Context Protocol (MCP) Trace Limitations

The Model Context Protocol (MCP) by Anthropic standardizes tool usage, allowing agents to dynamically connect to external data sources and APIs via JSON-RPC 2.0 over stdio or Server-Sent Events (SSE).²⁹ While MCP facilitates rapid capability composition, it introduces severe tracing and auditing vulnerabilities.

Currently, MCP lacks a native mechanism for user context propagation.³⁰ When an agent invokes an MCP tool, the protocol does not pass the initiating user's identity, permissions, or tenant scope via standardized request headers to the MCP server.³⁰ This creates "Accountability and Audit Trail Issues" because the downstream resource server cannot distinguish between different agents or users initiating the request.³¹ To mitigate this, emerging standards suggest embedding OpenTelemetry W3C Trace Context headers within the MCP `_meta` property to propagate trace IDs across the client-server boundary.³² For the AI Trace Auditor, an MCP tool execution span lacking explicit W3C Trace Context propagation must be flagged as a critical violation of compliance traceability.

Framework / Standard	Telemetry Format	Key Attributes for Multi-Agent Tracing	Identity Attribution Mechanism
OpenTelemetry (Draft)	OTLP	gen_ai.agent.id, gen_ai.task.*, session.id	Native span attributes per agent execution. ¹⁸
Langfuse	OTLP / Custom API	parentObservationId, trace_id	propagate_attributes() context manager. ²¹
Arize Phoenix	OpenInference (OTLP)	graph.node.id, graph.node.parent_id	Native metadata tagging and _handoffs tracking. ²⁴
Google ADK	OTLP (gRPC)	Typed JSON schemas, A2A protocol traces	Built-in OTel TraceProvider per agent. ²⁷

MCP	JSON-RPC 2.0	_meta (for trace context)	Gap: Lacks native identity propagation; relies on W3C headers. ³⁰
------------	--------------	---------------------------	---

3. Regulatory Analysis: EU AI Act & NIST RMF

The EU AI Act and the NIST AI RMF were designed primarily around monolithic AI deployments. The advent of multi-agent systems—where capabilities are composed dynamically at runtime, and agents autonomously delegate reasoning tasks—creates profound legal ambiguities regarding liability, record-keeping, and human oversight.

3.1. Article 12: Record-Keeping and Autonomous Traceability

Article 12 of the EU AI Act mandates that high-risk AI systems must technically allow for the "automatic recording of events (logs) over the duration of the lifetime of the system" to ensure traceability appropriate to the system's intended purpose.³⁴ In the context of multi-agent systems, standard application logging is legally indefensible.

The regulation demands the complete reconstructability of algorithmic decisions to facilitate post-market monitoring (Article 72) and risk identification (Article 79).³⁵ For a multi-agent orchestration, the audit trail must capture:

- The exact period of use (timestamped context for every agent decision).
- The reference database or context window utilized (e.g., which vector database an agent queried before making a decision).
- The exact input data that led to a match or tool invocation.
- The identification of any natural person involved in verifying the results (Human-in-the-Loop oversight under Article 14).³⁶

Because multi-agent systems operate across distributed environments, retroactively stitching logs is explicitly insufficient and will trigger regulatory penalties.³⁷ Emerging engineering standards, such as the Internet Engineering Task Force (IETF) draft for the "Agent Audit Trail" (AAT), propose cryptographic hash-chaining of records.³⁸ In this model, an ordered sequence of JSON audit records is linked via SHA-256 digests, rendering unauthorized modifications or "tombstoning" tamper-evident.³⁸ The AI Trace Auditor must evaluate traces to ensure that distributed agent handoffs maintain an unbroken, immutable chain of custody.

3.2. Article 25: The Provider vs. Deployer Liability Mesh

Article 25 ("Responsibilities along the AI value chain") represents the most significant legal hazard for enterprises utilizing multi-agent systems. The Act distinguishes between a "Provider" (who develops and places the system on the market, bearing heavy compliance burdens like CE marking and conformity assessments) and a "Deployer" (who simply uses the system under

their authority).³⁹

However, Article 25(1) establishes mechanisms where a deployer legally transforms into a provider. This occurs if the deployer puts their trademark on a high-risk system, modifies the intended purpose of a system to become high-risk, or makes a "**substantial modification**" to an existing high-risk AI system.¹²

In a multi-agent context, the barrier between deployer and provider is perilously thin. If an enterprise licenses a General Purpose AI (GPAI) model (acting as a deployer) but wraps it in a LangGraph orchestration, fine-tunes a router agent, and grants it autonomous access to internal databases via MCP tools, they have engineered a novel "compound AI system." Legal analyses suggest that chaining agents together or integrating dynamic decision-making workflows constitutes a substantial modification, immediately shifting the enterprise into the Provider liability tier.⁴²

Furthermore, systems that dynamically compose capabilities at runtime present a severe regulatory challenge. Article 3(23) defines substantial modification as changes "not foreseen or planned in the initial conformity assessment".⁴⁵ If an agent dynamically selects a tool from an updating registry at runtime, the provider cannot foresee the risks introduced, potentially triggering Article 25 and dispersing liability across model providers, system providers, and tool developers.⁴⁵ The AI Trace Auditor must flag dynamic, unvetted tool retrieval as a critical compliance violation.

3.3. Liability Attribution in Multi-Agent Chains

Academic and legal analyses highlight the "many hands problem" in multi-agent governance, where accountability is obscured across a multi-stakeholder value chain.⁴⁶

- **Board Oversight:** Legal doctrines such as *Caremark* (which establishes corporate director liability for systematic failure of oversight) are being applied to AI deployments. Deployers face liability for automated decisions made by their agents (e.g., an agentic system erroneously disabling user accounts or committing housing discrimination).⁴⁷
- **Causation:** In multi-agent environments (e.g., an orchestrator agent instructing a research agent, which hallucinates based on a database retrieval agent's error), establishing legal causation is exceptionally difficult.⁴⁹ If Agent A's prompt to Agent B lacks necessary constraints, and Agent B commits a violation, liability must be traced through the DAG.
- **GDPR Article 28:** The AI Act operates alongside the GDPR. If an agent delegates data processing to another agent hosted by a different vendor, the data controller is liable for ensuring Data Processing Agreements (DPAs) exist with every sub-processor in the chain.⁵⁰ Autonomous cross-border delegation by an agent breaks this compliance model entirely.

3.4. NIST AI RMF Implications

The NIST AI Risk Management Framework (RMF) emphasizes the continuous mapping and measurement of AI risks. For compound systems, the "Map" function requires organizations to define the boundaries of the AI system, including interdependent models and external APIs.⁵¹ Because multi-agent systems increase the attack surface (e.g., prompt injection leading to unauthorized tool use), the "Measure" and "Manage" functions require real-time monitoring of agent state and the implementation of strict policy enforcement gateways at the tool boundary (e.g., verifying agent identity before allowing an MCP tool execution).⁵¹

4. Competitive Landscape

The enterprise AI governance market is evolving to address agentic AI, but a significant gap remains between static policy documentation platforms and dynamic, code-level trace auditing tools. Legal and compliance teams require deterministic mapping of runtime behavior to regulatory articles, which most existing tools cannot natively provide for multi-agent DAGs.

Tool / Platform	Multi-Agent Support	Pricing Model	The Capability Gap for Multi-Agent Auditing
Credo AI	Partial (GAIA Assistant)	Custom Enterprise SaaS	Excellent for mapping AI projects to frameworks (EU AI Act, NIST) via documentation and workflows. However, it operates primarily as an intelligence layer and registry; it lacks deep, code-level execution trace ingestion, DAG parsing, and automated span-by-span behavioral validation. ⁵³

Holistic AI	Low	Custom Enterprise SaaS	Operates as a high-level risk management and compliance auditing function. Focuses on system inventory, bias audits, and documentation generation, but does not natively parse OTLP multi-agent telemetry or monitor dynamic agent handoffs at runtime. ⁵⁵
AgentOps	High	Freemium / SaaS	Designed specifically for agent observability. Excellent for session replays, agent execution tracking, and debugging. ⁵⁷ The gap lies in compliance mapping: it does not automatically generate EU AI Act conformity scores or Article 12 compliance artifacts based on trace data.
Braintrust / Weave	Medium	SaaS	Highly focused on LLM-as-a-judge scoring, prompt engineering workflows, and

			<p>single-model evaluations.¹³ While they support trace visualization, they lack the specialized regulatory compliance matrices required by legal teams to prove value-chain accountability.</p>
Patronus AI	Medium	Custom Enterprise SaaS	<p>Exceptional at adversarial testing and safety compliance (e.g., detecting PII leaks or hallucinations).⁵⁷ However, its architecture primarily targets LLM outputs rather than auditing the structural compliance of multi-agent DAGs and inter-agent delegation vulnerabilities.</p>
Openlayer	Medium	SaaS	<p>Combines compliance automation with 100+ behavioral tests and runtime security controls.⁵³ It maps to the EU AI Act and NIST, but is still maturing its ability to parse complex multi-agent</p>

			telemetry and attribute specific failures to individual nodes within an agent swarm. ⁵³
--	--	--	--

4.1. Open-Source Tooling

There are currently no open-source CLIs designed specifically to ingest OTLP traces from multi-agent systems and output deterministic EU AI Act compliance scores.

- **Petri** (by Anthropic) is an open-source tool that automates AI safety research via multi-turn probing and adversarial conversations, but it is an evaluation generator, not a post-execution compliance auditor.⁵⁸
- **mcp-sec-audit** provides static application security testing (SAST) and dynamic sandboxing specifically for MCP servers, detecting risky capabilities like command execution.⁵⁹ While highly relevant for securing tools, it does not audit the orchestration traces of the agents calling those tools.
- **Prowler** automates cloud security posture management (CSPM) against frameworks like CIS and GDPR, but operates at the cloud infrastructure level, not the AI agent trace level.⁶⁰

4.2. Cloud Provider Audit Logs

Major cloud providers expose audit logs, but these are optimized for infrastructure security, not algorithmic transparency.

- **AWS Bedrock Agents:** Logs actions via CloudTrail data events (e.g., InvokeAgent). While Bedrock AgentCore emits OTLP-compatible telemetry to CloudWatch (capturing session count, latency, and memory resources), the traces are raw telemetry.⁶¹ Translating a CloudTrail JSON log into an Article 12 compliance artifact requires extensive custom engineering.
- **Azure AI Agent Service:** Utilizes Application Insights (an OpenTelemetry feature within Azure Monitor) to track end-to-end visibility.⁶³ Traces are stored in the apptraces table with specific fields (SDKVersion, SessionId, SeverityLevel), and operations are represented as hierarchical spans.⁶³ Azure provides a dedicated "Agent details view," but converting these spans into EU AI Act conformity assessments relies on integration with Purview Compliance Manager.⁶³
- **Google Vertex AI Agent Builder / ADK:** Emits structured trace data via the OTLP API to Cloud Trace and logs administrative actions to Cloud Audit Logs (Admin Activity and Data Access).⁶⁶ While highly observable, it remains the responsibility of the developer to map these traces to regulatory frameworks.

5. Recommended Data Model Extensions

To transition the AI Trace Auditor from a single-model tool to a multi-agent compliance engine, the core Pydantic schemas (NormalizedTrace and NormalizedSpan) must be fundamentally refactored. A linear list of spans cannot accurately represent parallel agent execution, tool-use chains, or recursive reasoning loops. The data model must embrace a Directed Acyclic Graph (DAG) architecture.

5.1. Refactoring NormalizedSpan

The NormalizedSpan model must be extended to capture agent identity, semantic actions, and exact cryptographic provenance to satisfy Article 12.

Agent Identity & Role Context:

- `agent_id` (String): The unique identifier of the agent executing the span (mapping to OTel `gen_ai.agent.id`).
- `agent_name` (String): The functional role of the agent (e.g., "Data Sanitizer") (mapping to OTel `gen_ai.agent.name`).
- `agent_framework` (String): The orchestration engine utilized (e.g., langgraph, crewai, adk), crucial for determining how state is managed.

Hierarchical and DAG Context:

- `parent_span_id` (String): The immediate calling span, required for reconstructing the execution tree.
- `orchestrator_id` (String): The identifier of the root agent or router managing the overall session.
- `delegation_path` (List): An ordered array of `agent_ids` representing the exact chain of command that led to the current span. This is critical for auditing the "Confused Deputy" problem, where a malicious input traverses multiple agents before executing a payload.

Tool Execution and Provenance:

- `span_kind` (Enum): Must distinguish between LLM_GENERATION, TOOL_CALL, AGENT_HANDOFF, MEMORY_READ, and MEMORY_WRITE.
- `tool_name` (String): The specific capability invoked.
- `mcp_server_uri` (String): If an MCP tool is used, the endpoint of the server.
- `data_provenance_hash` (String): A SHA-256 cryptographic digest of the input, output, and the previous span's hash. This directly addresses the tamper-evident logging requirements of the Agent Audit Trail (AAT) proposed to satisfy Article 12.³⁸

5.2. Refactoring NormalizedTrace

The NormalizedTrace model must represent the session holistically, grouping multi-agent

workflows and evaluating system-level compliance.

- **session_id (String):** Correlates all distributed spans belonging to a single multi-turn interaction (mapping to OTel session.id).
- **dag_adjacency_list (Dict):** A mathematical representation of the graph edges (span-to-span relationships). This allows the auditing engine to detect infinite loops (circular delegation) or unauthorized agent-to-agent communication bypassing the orchestrator.
- **compliance_scores (Object):**
 - **system_score (Float):** The aggregate compliance score for the entire DAG.
 - **agent_scores (Dict):** Individual compliance scores attributed to specific agent_ids. This isolation is mandatory for root-cause analysis when an upstream agent causes a downstream compliance failure.

5.3. Computing Compliance: Bottom-Up Penalty Propagation

In a multi-agent system, compliance cannot be averaged. A critical failure in a single leaf node (e.g., an agent executing an MCP tool that leaks Personally Identifiable Information) fundamentally compromises the entire upstream chain and violates Article 25 provider obligations.

The auditing logic must utilize a **Bottom-Up Penalty Propagation** algorithm:

1. **Leaf Node Evaluation:** The auditor first evaluates terminal spans (tool executions, final LLM outputs, database queries) against strict compliance rules (e.g., PII redaction, prompt injection boundaries).
2. **Propagation of Liability:** If a leaf node fails a compliance check (e.g., Article 15 Cybersecurity), a weighted penalty propagates up the `delegation_path`. The parent agent that delegated the task receives a governance penalty for failing to implement output validation or human-in-the-loop guardrails.
3. **Responsibility Attribution:** If Agent A instructs Agent B to perform a non-compliant action, the auditor analyzes the `AGENT_HANDOFF` span. If Agent A's instruction prompt lacked safety constraints or contained malicious intent, Agent A's `agent_score` is severely penalized. If Agent A provided secure instructions but Agent B hallucinated and violated those instructions, the penalty is isolated to Agent B. This logic directly maps to the legal requirement of identifying causation across the AI value chain.⁴⁹

6. Implementation Priorities and Testing

Transitioning the AI Trace Auditor to support multi-agent systems requires a phased engineering approach, prioritizing data ingestion and graph reconstruction before implementing advanced compliance heuristics.

6.1. Phased Implementation Roadmap

Phase 1: DAG Data Model & OTLP Ingestion (Immediate)

- Refactor the Pydantic schemas to support the NormalizedSpan and NormalizedTrace extensions detailed above.
- Implement parsing logic for the emerging OTel gen_ai.* semantic conventions, specifically extracting gen_ai.agent.id and session.id.
- Build custom adapters for Langfuse (parsing parentObservationId and propagate_attributes()) and Arize (parsing graph.node.id and graph.node.parent_id) to ensure accurate DAG reconstruction from external trace formats.

Phase 2: Graph-Based Compliance Scoring (Short-Term)

- Develop the Bottom-Up Penalty Propagation algorithm to compute agent_scores and system_score.
- Implement cryptographic hash-chain validation to verify the integrity of the Agent Audit Trail (AAT), explicitly fulfilling the tamper-evidence requirements of EU AI Act Article 12.
- Map specific DAG anomalies (e.g., an agent writing to a database without a preceding human-approval span) to Article 14 (Human Oversight) violations.

Phase 3: MCP and Boundary Auditing (Medium-Term)

- Develop a specific auditing module for the Model Context Protocol. The auditor must flag any MCP tool execution span that lacks W3C Trace Context propagation in its _meta field, as this breaks cross-boundary accountability.³⁰
- Implement checks to detect dynamic, unvetted tool retrieval at runtime, flagging it as an unauthorized "substantial modification" under Article 25.⁴⁵

6.2. Testing Multi-Agent Compliance

Validating a graph-based compliance auditor requires complex, non-deterministic test fixtures. Standard single-prompt unit tests are inadequate.

Test Fixtures and Scenarios:

- **Linear Delegation (CrewAI pattern):** Agent A (Researcher) retrieves data and hands it off to Agent B (Writer). Validate that the trace correctly links the parent-child data flow, maintains session correlation, and propagates metadata attributes successfully.
- **Conditional Routing (LangGraph pattern):** A Router Agent dynamically chooses between a safe RAG Agent and a high-risk Code Execution Agent based on user input. Validate that the auditor maps the unexecuted branch as "skipped" and correctly audits the executed path for safety constraints.
- **Concurrent Debate (AutoGen pattern):** Multiple agents converse simultaneously in a shared context. Validate that the auditor can parse concurrent INTERNAL spans, distinguish agent identities, and attribute compliance violations to the specific agent that introduced a hallucination or toxic output.

Generating Synthetic Traces:

Collecting real-world multi-agent traces with specific, severe compliance violations (e.g., PII leaks via MCP) is difficult and poses privacy risks. The development pipeline must utilize synthetic trace generation.

- Frameworks like **TraceLLM** can be adapted to act as a world model, generating realistic synthetic multi-step behavioral traces.⁶⁸
- By prompting an LLM to output specifically malformed OpenTelemetry JSON payloads (e.g., simulating a trace where an agent accesses a local filesystem via an MCP tool without authorization), developers can build a robust CI/CD pipeline for the auditing logic itself, ensuring the CLI accurately catches complex value-chain vulnerabilities.

Edge Cases for Validation:

- **Circular Delegation (Infinite Loops):** Agents endlessly handing off tasks to one another. The DAG parser must detect graph cycles and flag them as an Article 15 (Robustness) failure.
- **Agent Self-Modification:** An agent dynamically rewriting its own system prompt or spawning an undocumented sub-agent at runtime. This triggers severe Article 25 provider liability and must be flagged as a critical violation.
- **MCP Server Chains:** An agent calling an MCP server that internally proxies the request to another unmonitored API. The auditor must recognize the boundary of the trace, identify the loss of provenance, and flag the black-box execution as an Article 12 transparency risk.

By executing this roadmap, the AI Trace Auditor will evolve from a basic observability utility into an indispensable enterprise governance platform, directly addressing the complex regulatory realities of the agentic AI era.

7. Sources

- ⁶⁹ EU AI Act US Enterprise Compliance Guide: <https://ajithp.com/2026/03/09/eu-ai-act-us-enterprise-compliance-guide/>
- ²⁰ Langfuse LangGraph Agents Example: https://langfuse.com/guides/cookbook/example_langgraph_agents
- ²³ Langfuse Trace IDs and Distributed Tracing: <https://langfuse.com/docs/observability/features/trace-ids-and-distributed-tracing>
- ¹⁴ OpenTelemetry Semantic Conventions (gen_ai issue): <https://github.com/open-telemetry/semantic-conventions/issues/2664>
- ²⁴ Arize Phoenix Agent Graph Visualization: <https://arize.com/docs/ax/observe/tracing/agents>
- ¹⁰ Arize Phoenix Tracing Tutorial: <https://arize.com/docs/phoenix/tracing/tutorial>
- ⁵ Credo AI Webinar (Forrester):

- <https://www.credo.ai/webinar/the-ai-governance-control-plane-in-2026-with-forrester>
- ⁷⁰ Holistic AI Audits: <https://www.holisticai.com/ai-audits>
- ⁵⁷ Top 5 Agent Simulation Platforms: <https://dev.to/debmckinney/top-5-agent-simulation-platforms-in-2026-333j>
- ¹ Braintrust Agent Observability: <https://www.braintrust.dev/articles/best-ai-agent-observability-tools-2026>
- ¹³ Openlayer Agent Evaluation: <https://www.openlayer.com/blog/post/best-ai-agent-evaluation-platforms>
- ² McKinsey - State of AI 2025: <https://www.mckinsey.com/capabilities/business-building/our-insights/how-to-build-businesses-faster-and-better-with-ai>
- ³ State of AI Report (Sequoia/Air Street): <https://www.stateof.ai/>
- ⁶ The State of AI 2025 (WAIU): <https://caio.waiu.org/p/the-state-of-ai-2025>
- ⁴ Leonis Cap - State of AI 2025: <https://www.leoniscap.com/research/the-state-of-ai-in-2025>
- ⁶⁶ Google Vertex AI Audit Logging: <https://docs.cloud.google.com/vertex-ai/docs/general/audit-logging>
- ⁵³ Credo AI Reviews and Alternatives: <https://www.openlayer.com/blog/post/credo-ai-reviews-pricing-alternatives>
- ⁵⁵ AI Governance Platforms Comparison: <https://www.superblocks.com/blog/ai-governance-platform>
- ⁷¹ 10 Best AI Governance Tools: <https://www.reco.ai/compare/ai-governance-tools>
- ¹¹ EU AI Act Regulatory Analysis: <https://georgiactsa.org/research/regulatory-knowledge-support/blog-eu1.html>
- ⁴⁷ Liability Considerations for Agentic AI: <https://www.lathropgpm.com/insights/liability-considerations-for-developers-and-users-of-agentic-ai-systems/>
- ⁷ Zycus Agentic AI RFPs: <https://cporising.com/2026/01/14/best-of-2025-zycus-goes-all-in-on-agentic-ai-at-horizon-2025/>
- ⁸ Aisera Agentic AI Compliance: <https://aisera.com/blog/agentic-ai-compliance/>
- ⁹ KPMG AI Governance for Agentic Era: <https://kpmg.com/us/en/articles/2025/ai-governance-for-the-agentic-ai-era.html>
- ¹⁴ OpenTelemetry gen_ai Semantic Conventions: <https://github.com/open-telemetry/semantic-conventions/issues/2664>
- ¹⁵ OpenTelemetry gen_ai Task Conventions: <https://github.com/open-telemetry/semantic-conventions/issues/2665>
- ⁷² TraceLLM Synthetic Trace Generation: <https://arxiv.org/html/2502.17439v3>
- ⁵⁹ MCP Security Audit Tool: <https://arxiv.org/html/2603.21641v1>
- ⁶³ Azure AI Agent Service Observability: <https://bharathkumarr2498.medium.com/building-cloud-ai-agents-microsoft-azure-secu>

- [rity-management-made-simple-de27c447e2eb](#)
- ⁴² EU AI Act Deployers vs Providers:
<https://www.aoshearman.com/en/insights/ao-shearman-on-tech/zooming-in-on-ai-4-what-is-the-interplay-between-deployers-and-providers-in-the-eu-ai-act>
- ¹² EU AI Act Provider Obligations: <https://arxiv.org/html/2510.13591v1>
- ⁵⁰ Agent Data Handling Policy Discussions:
<https://github.com/StevenJohnson998/agent-data-handling-policy/discussions/4>
- ⁵⁸ Anthropic Petri Safety Auditing: <https://alignment.anthropic.com/2025/petri/>
- ³⁹ EU AI Act Best Practices:
https://www.appliedai.de/uploads/files/Best-Practices-for-Implementing-the-EU-AI-Act_2025-07-02-092027_vwvf.pdf
- ⁴⁸ AI Agent Deployer Liability: <https://astraia.law/insights/ai-agent-deployer-liability>
- ⁷³ MSH-GAN Synthetic Traffic Generation:
https://www.researchgate.net/publication/399748733_Synthetic_Packet_Traffic_Generative_Adversarial_Networks_in_Multi_Agents_With_Peer-to-Peer_and_Global_Priority_Queue_Generation
- ⁶⁰ Prowler Open Cloud Security: <https://github.com/prowler-cloud/prowler>
- ³⁰ MCP Trace Context Issues: <https://arxiv.org/html/2603.13417>
- ²⁹ Edge Delta MCP Overview: <https://docs.edgedelta.com/edge-delta-mcp-overview/>
- ⁴⁶ Governing AI Agents under EU AI Act:
<https://thefuturesociety.org/wp-content/uploads/2023/04/Report-Ahead-of-the-Curve-Governing-AI-Agents-Under-the-EU-AI-Act-4-June-2025.pdf>
- ⁶⁸ Divide-and-Conquer Multi-Context Reasoning:
<https://aclanthology.org/2025.emnlp-main.4.pdf>
- ¹⁶ OpenTelemetry session.id Proposal:
<https://github.com/open-telemetry/semantic-conventions/issues/2883>
- ¹⁷ OpenTelemetry gen_ai.agent attributes:
<https://github.com/open-telemetry/semantic-conventions/issues/2112>
- ¹⁸ OpenTelemetry gen_ai.agent Span Attributes:
<https://opentelemetry.io/docs/specs/semconv/gen-ai/gen-ai-agent-spans/>
- ⁷⁴ Decoding Key Roles in the AI Act:
<https://haerting.de/en/insights/provider-or-deployer-decoding-the-key-roles-in-the-ai-act/>
- ⁴⁰ European Union AI Act Guide:
<https://www.twobirds.com/-/media/new-website-content/pdfs/capabilities/artificial-intelligence/european-union-artificial-intelligence-act-guide.pdf>
- ³⁸ Agent Audit Trail (IETF Draft):
<https://datatracker.ietf.org/doc/draft-sharif-agent-audit-trail/>
- ²⁵ Google ADK Multi-Agent Applications:
<https://www.aalpha.net/blog/google-agent-development-kit-adk-for-multi-agent-applications/>

- ²⁸ Inside Google ADK:
<https://medium.com/@vsankarayogi/inside-googles-agent-development-kit-adk-part-2-2b1a7b8ae885>
- ⁷⁵ Google ADK Logging: <https://google.github.io/adk-docs/observability/logging/>
- ²⁴ Arize Phoenix Custom Tracing: <https://arize.com/docs/ax/observe/tracing/agents>
- ⁶¹ AWS Bedrock CloudTrail Logging:
<https://docs.aws.amazon.com/bedrock/latest/userguide/logging-using-cloudtrail.html>
- ⁶² AWS Bedrock AgentCore Observability:
<https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/observability.html>
- ⁶⁵ Azure Monitor Agents View:
<https://learn.microsoft.com/en-us/azure/azure-monitor/app/agents-view>
- ⁶⁴ Azure AppTraces Table:
<https://learn.microsoft.com/en-us/azure/azure-monitor/reference/tables/appttraces>
- ⁷⁶ Copilot Studio AppInsights Telemetry:
<https://learn.microsoft.com/en-us/dynamics365/guidance/resources/copilot-studio-appinsights>
- ⁵⁶ Teramind AI Governance Tools: <https://www.teramind.co/blog/ai-governance-tools/>
- ²³ Langfuse Trace IDs:
<https://langfuse.com/docs/observability/features/trace-ids-and-distributed-tracing>
- ¹⁹ Langfuse OpenTelemetry Mapping:
<https://langfuse.com/docs/observability/sdk/overview>
- ²⁶ The Complete Guide to Google ADK:
<https://sidbharath.com/blog/the-complete-guide-to-googles-agent-development-kit-adk/>
- ⁶⁷ Instrumenting Google ADK:
<https://docs.cloud.google.com/stackdriver/docs/instrumentation/ai-agent-adk>
- ⁷⁷ Operator Model of EU AI Act:
https://medium.com/@ive_20203/the-eu-ai-acts-operator-model-what-every-company-needs-to-know-before-deploying-ai-9a58452aaf61
- ³⁵ AI Act Record-Keeping:
<https://truescreen.io/insights/ai-act-record-keeping-requirements/>
- ³¹ MCP Security Best Practices:
https://modelcontextprotocol.io/docs/tutorials/security/security_best_practices
- ¹⁸ OTEL gen_ai.agent.version Attributes:
<https://opentelemetry.io/docs/specs/semconv/gen-ai/gen-ai-agent-spans/>
- ⁴¹ EU AI Act Article 25 Text: <https://www.activemind.legal/legislation/ai-act/article-25/>
- ⁷⁸ EU AI Office Article 12 FAQ: <https://ai-act-service-desk.ec.europa.eu/en/ai-act/article-12>
- ³⁶ EU AI Act Article 12 Text: <https://www.activemind.legal/legislation/ai-act/article-12/>
- ²⁷ ADK Go 1.0 Release: <https://developers.googleblog.com/adk-go-10-arrives/>
- ¹⁸ OpenTelemetry CLIENT and INTERNAL spans:
<https://opentelemetry.io/docs/specs/semconv/gen-ai/gen-ai-agent-spans/>

- ⁴⁵ EU AI Act Structural Failures: <https://www.europeanlawblog.eu/pub/dq249o3c>
- ⁵¹ NAIC AI Compound Systems Oversight: <https://content.naic.org/sites/default/files/inline-files/AI%20Model%20Law%20Request%20for%20Information-Comments%20Received%207.3.25.pdf>
- ⁴³ Pitch Law Provider vs Deployer: <https://www.pitch.law/knowledge-base/provider-vs-deployer-ai-act>
- ³³ MCP Specification Draft: <https://modelcontextprotocol.io/specification/draft/basic>
- ³⁷ ISO 42001 and EU AI Act Logging: <https://www.isms.online/iso-42001/eu-ai-act/article-12/>
- ³⁴ AI Act Searchable Database: <https://aiact.algolia.com/article-12/>
- ⁵⁴ Credo AI Product Governance: <https://www.credo.ai/product>
- ²² Langfuse Multi-Agent Tags: <https://github.com/orgs/langfuse/discussions/10473>
- ²⁵ ADK Task and Memory Validation: <https://www.aalpha.net/blog/google-agent-development-kit-adk-for-multi-agent-applications/>
- ³⁶ ActiveMind Legal Article 12: <https://www.activemind.legal/legislation/ai-act/article-12/>
- ⁵² Privacy-Preserving Multi-Agent Governance: https://www.tdcommons.org/cgi/viewcontent.cgi?article=10818&context=dpubs_series
- ⁴⁴ EDPB Privacy Risks in LLMs: <https://www.edpb.europa.eu/system/files/2025-04/ai-privacy-risks-and-mitigations-in-llms.pdf>
- ⁴⁹ MHC AI Liability Causation: <https://www.mhc.ie/latest/insights/ai-liability>
- ⁷⁹ MCP Provenance Tracking: <https://arxiv.org/html/2511.20920v1>
- ⁵⁴ Credo AI The Complete AI Governance Stack: <https://www.credo.ai/product>
- ²¹ Langfuse Metadata Feature: <https://langfuse.com/docs/observability/features/metadata>

Works cited

1. 5 best AI agent observability tools for agent reliability in 2026 - Articles - Braintrust, accessed April 2, 2026, <https://www.braintrust.dev/articles/best-ai-agent-observability-tools-2026>
2. How to build businesses faster and better with AI - McKinsey, accessed April 2, 2026, <https://www.mckinsey.com/capabilities/business-building/our-insights/how-to-build-businesses-faster-and-better-with-ai>
3. Welcome to State of AI Report 2025, accessed April 2, 2026, <https://www.stateof.ai/>
4. The State of AI in 2025 - Leonis Capital, accessed April 2, 2026, <https://www.leoniscap.com/research/the-state-of-ai-in-2025>
5. The AI Governance Control Plane in 2026 with Forrester - Credo AI, accessed April 2, 2026, <https://www.credo.ai/webinar/the-ai-governance-control-plane-in-2026-with-forrester>

[rester](#)

6. The State of AI 2025 - The CAIO Hub, accessed April 2, 2026, <https://caio.waiu.org/p/the-state-of-ai-2025>
7. Best of 2025: Zycus Goes All-in on Agentic AI at Horizon 2025 - CPO Rising, accessed April 2, 2026, <https://cporising.com/2026/01/14/best-of-2025-zycus-goes-all-in-on-agentic-ai-at-horizon-2025/>
8. Agentic AI Compliance: A Technical Guide to Governing AI Agents - Aisera, accessed April 2, 2026, <https://aisera.com/blog/agentic-ai-compliance/>
9. AI governance for the agentic AI era - KPMG International, accessed April 2, 2026, <https://kpmg.com/us/en/articles/2025/ai-governance-for-the-agentic-ai-era.html>
10. Tracing Tutorial - Phoenix - Arize AI, accessed April 2, 2026, <https://arize.com/docs/phoenix/tracing/tutorial>
11. #4A Strengths and Vulnerabilities of the European Union's AI Act: Part I | Georgia Clinical and Translational Science Alliance | Atlanta GA, accessed April 2, 2026, <https://georgiactsa.org/research/regulatory-knowledge-support/blog-eu1.html>
12. Subject Roles in the EU AI Act: Mapping and Regulatory Implications - arXiv, accessed April 2, 2026, <https://arxiv.org/html/2510.13591v1>
13. Best AI Agent Evaluation Platforms | Openlayer, accessed April 2, 2026, <https://www.openlayer.com/blog/post/best-ai-agent-evaluation-platforms>
14. Semantic Conventions for Generative AI Agentic Systems (gen_ai.*) #2664 - GitHub, accessed April 2, 2026, <https://github.com/open-telemetry/semantic-conventions/issues/2664>
15. Semantic Conventions for Generative AI Tasks (gen_ai.task.*) · Issue #2665 - GitHub, accessed April 2, 2026, <https://github.com/open-telemetry/semantic-conventions/issues/2665>
16. Add session.id attribute to GenAI semantic conventions · Issue #2883 - GitHub, accessed April 2, 2026, <https://github.com/open-telemetry/semantic-conventions/issues/2883>
17. [gen-ai] gen_ai.agent semantic conventions need to be expanded further to include tools list #2112 - GitHub, accessed April 2, 2026, <https://github.com/open-telemetry/semantic-conventions/issues/2112>
18. Semantic Conventions for GenAI agent and framework spans - OpenTelemetry, accessed April 2, 2026, <https://opentelemetry.io/docs/specs/semconv/gen-ai/gen-ai-agent-spans/>
19. Langfuse SDKs, accessed April 2, 2026, <https://langfuse.com/docs/observability/sdk/overview>
20. Example - Trace and Evaluate LangGraph Agents - Langfuse, accessed April 2, 2026, https://langfuse.com/guides/cookbook/example_langgraph_agents
21. Metadata - Langfuse, accessed April 2, 2026, <https://langfuse.com/docs/observability/features/metadata>
22. Multi Process Ray Agent · langfuse · Discussion #10473 - GitHub, accessed April 2, 2026, <https://github.com/orgs/langfuse/discussions/10473>
23. Trace IDs & Distributed Tracing - Langfuse, accessed April 2, 2026, <https://langfuse.com/docs/observability/features/trace-ids-and-distributed-tracin>

- g
24. Agent Graph & Path - Arize AX Docs, accessed April 2, 2026, <https://arize.com/docs/ax/observe/tracing/agents>
 25. Guide to Google Agent Development Kit (ADK) - Aalpha, accessed April 2, 2026, <https://www.aalpha.net/blog/google-agent-development-kit-adk-for-multi-agent-applications/>
 26. The Complete Guide to Google's Agent Development Kit (ADK) - Sid Bharath, accessed April 2, 2026, <https://sidbharath.com/blog/the-complete-guide-to-googles-agent-development-kit-adk/>
 27. ADK Go 1.0 Arrives! - Google Developers Blog, accessed April 2, 2026, <https://developers.googleblog.com/adk-go-10-arrives/>
 28. Inside Google's Agent Development Kit (ADK) — Part 2 | by Vamsi Krishna Sankarayogi, accessed April 2, 2026, <https://medium.com/@vsankarayogi/inside-googles-agent-development-kit-adk-part-2-2b1a7b8ae885>
 29. Model Context Protocol (MCP) Guide | Connect AI Agents to Your Systems, accessed April 2, 2026, <https://docs.edgedelta.com/edge-delta-mcp-overview/>
 30. Design Patterns for Deploying AI Agents with Model Context Protocol - arXiv, accessed April 2, 2026, <https://arxiv.org/html/2603.13417>
 31. Security Best Practices - Model Context Protocol, accessed April 2, 2026, https://modelcontextprotocol.io/docs/tutorials/security/security_best_practices
 32. MCP Audit Logging: Tracing AI Agent Actions for Compliance - Tetrade, accessed April 2, 2026, <https://tetrade.io/learn/ai/mcp/mcp-audit-logging>
 33. Overview - Model Context Protocol, accessed April 2, 2026, <https://modelcontextprotocol.io/specification/draft/basic>
 34. Article 12: Record-Keeping | AI Act made searchable by Algolia. Chapters, articles and recitals easily readable, accessed April 2, 2026, <https://aiact.algolia.com/article-12/>
 35. EU AI Act Article 12: Record-Keeping for High-Risk AI Systems - TrueScreen, accessed April 2, 2026, <https://truescreen.io/insights/ai-act-record-keeping-requirements/>
 36. Article 12 - Record-keeping AI Act | activeMind.legal, accessed April 2, 2026, <https://www.activemind.legal/legislation/ai-act/article-12/>
 37. Demonstrating Compliance With EU AI Act Article 12 Record Keeping Using ISO 42001 Governance Controls - ISMS.online, accessed April 2, 2026, <https://www.isms.online/iso-42001/eu-ai-act/article-12/>
 38. Agent Audit Trail: A Standard Logging Format for Autonomous AI Systems - IETF Datatracker, accessed April 2, 2026, <https://datatracker.ietf.org/doc/draft-sharif-agent-audit-trail/>
 39. AI Act Governance: - Best Practices for Implementing the EU AI Act - appliedAI Initiative, accessed April 2, 2026, https://www.appliedai.de/uploads/files/Best-Practices-for-Implementing-the-EU-AI-Act_2025-07-02-092027_vwvf.pdf
 40. European Union Artificial Intelligence Act: a guide, accessed April 2, 2026,

- <https://www.twobirds.com/-/media/new-website-content/pdfs/capabilities/artificial-intelligence/european-union-artificial-intelligence-act-guide.pdf>
41. Article 25 - Responsibilities along the AI value chain AI Act | activeMind.legal, accessed April 2, 2026, <https://www.activemind.legal/legislation/ai-act/article-25/>
 42. Zooming in on AI – #4: What is the interplay between “Deployers” and “Providers” in the EU AI Act? - A&O Shearman, accessed April 2, 2026, <https://www.aoshearman.com/en/insights/ao-shearman-on-tech/zooming-in-on-ai-4-what-is-the-interplay-between-deployers-and-providers-in-the-eu-ai-act>
 43. Provider vs Deployer: Understanding Your Role Under the AI Act - Knowledge Base Pitch, accessed April 2, 2026, <https://www.pitch.law/knowledge-base/provider-vs-deployer-ai-act>
 44. AI Privacy Risks & Mitigations – Large Language Models (LLMs) - European Data Protection Board, accessed April 2, 2026, <https://www.edpb.europa.eu/system/files/2025-04/ai-privacy-risks-and-mitigations-in-llms.pdf>
 45. Agentic Tool Sovereignty - European Law Blog, accessed April 2, 2026, <https://www.europeanlawblog.eu/pub/dq249o3c>
 46. Ahead of the Curve: Governing AI Agents Under the EU AI Act | The Future Society, accessed April 2, 2026, <https://thefuturesociety.org/wp-content/uploads/2023/04/Report-Ahead-of-the-Curve-Governing-AI-Agents-Under-the-EU-AI-Act-4-June-2025.pdf>
 47. Liability Considerations for Developers and Users of Agentic AI Systems - Lathrop GPM, accessed April 2, 2026, <https://www.lathropgpm.com/insights/liability-considerations-for-developers-and-users-of-agentic-ai-systems/>
 48. Not an Agent. Not a Defense: Seven Doctrines That Already Hold AI Deployers Liable, accessed April 2, 2026, <https://astraea.law/insights/ai-agent-deployer-liability>
 49. AI Liability | Mason Hayes Curran, accessed April 2, 2026, <https://www.mhc.ie/latest/insights/ai-liability>
 50. EU AI Act & multi-agent chains. The autonomous agent calling compliance problem · StevenJohnson998 agent-data-handling-policy · Discussion #4 - GitHub, accessed April 2, 2026, <https://github.com/StevenJohnson998/agent-data-handling-policy/discussions/4>
 51. AI MODEL LAW REQUEST FOR INFORMATION – COMMENTS RECEIVED - NAIC, accessed April 2, 2026, <https://content.naic.org/sites/default/files/inline-files/AI%20Model%20Law%20Request%20for%20Information-Comments%20Received%207.3.25.pdf>
 52. Policy-Bound Governance for Privacy-Preserving Multi-Agent AI Systems Across Multiple Service Providers - Technical Disclosure Commons, accessed April 2, 2026, https://www.tdcommons.org/cgi/viewcontent.cgi?article=10818&context=dpubs_series
 53. Credo AI reviews, pricing, and alternatives (January 2026) - Openlayer, accessed April 2, 2026,

- <https://www.openlayer.com/blog/post/credo-ai-reviews-pricing-alternatives>
54. The Leader in Responsible AI - Product - Credo AI, accessed April 2, 2026, <https://www.credo.ai/product>
 55. 10 Best AI Governance Platforms for Enterprise Teams in 2026 - Superblocks, accessed April 2, 2026, <https://www.superblocks.com/blog/ai-governance-platform>
 56. The 7 Best AI Governance Tools in 2026 - Teramind, accessed April 2, 2026, <https://www.teramind.co/blog/ai-governance-tools/>
 57. Top 5 Agent Simulation Platforms in 2026 - DEV Community, accessed April 2, 2026, <https://dev.to/debmckinney/top-5-agent-simulation-platforms-in-2026-333j>
 58. Petri: An open-source auditing tool to accelerate AI safety research, accessed April 2, 2026, <https://alignment.anthropic.com/2025/petri/>
 59. Auditing MCP Servers for Over-Privileged Tool Capabilities - arXiv, accessed April 2, 2026, <https://arxiv.org/html/2603.21641v1>
 60. GitHub - prowler-cloud/prowler: Prowler is the world's most widely used open-source cloud security platform that automates security and compliance across any cloud environment., accessed April 2, 2026, <https://github.com/prowler-cloud/prowler>
 61. Monitor Amazon Bedrock API calls using CloudTrail, accessed April 2, 2026, <https://docs.aws.amazon.com/bedrock/latest/userguide/logging-using-cloudtrail.html>
 62. Observe your agent applications on Amazon Bedrock AgentCore Observability, accessed April 2, 2026, <https://docs.aws.amazon.com/bedrock-agentcore/latest/devguide/observability.html>
 63. Building Cloud AI Agents: Microsoft Azure Security Management Made Simple, accessed April 2, 2026, <https://bharathkumarr2498.medium.com/building-cloud-ai-agents-microsoft-azure-security-management-made-simple-de27c447e2eb>
 64. Azure Monitor Logs reference - AppTraces - Microsoft Learn, accessed April 2, 2026, <https://learn.microsoft.com/en-us/azure/azure-monitor/reference/tables/appraces>
 65. Monitor AI Agents with Application Insights - Azure - Microsoft Learn, accessed April 2, 2026, <https://learn.microsoft.com/en-us/azure/azure-monitor/app/agents-view>
 66. Vertex AI audit logging information - Google Cloud Documentation, accessed April 2, 2026, <https://docs.cloud.google.com/vertex-ai/docs/general/audit-logging>
 67. Instrument ADK applications with OpenTelemetry | Google Cloud Observability, accessed April 2, 2026, <https://docs.cloud.google.com/stackdriver/docs/instrumentation/ai-agent-adk>
 68. Large Language Models as Realistic Microservice Trace Generators - ACL Anthology, accessed April 2, 2026, <https://aclanthology.org/2025.emnlp-main.4.pdf>
 69. What the EU AI Act Means for US Enterprises with European Exposure, accessed

April 2, 2026,

<https://ajithp.com/2026/03/09/eu-ai-act-us-enterprise-compliance-guide/>

70. AI Audits - Mitigate & Monitor AI Risks - Holistic AI, accessed April 2, 2026, <https://www.holisticai.com/ai-audits>
71. Top 10 AI Governance Tools for Secure & Responsible AI Use [2025] - Reco, accessed April 2, 2026, <https://www.reco.ai/compare/ai-governance-tools>
72. Large Language Models as Realistic Microservice Trace Generators - arXiv.org, accessed April 2, 2026, <https://arxiv.org/html/2502.17439v3>
73. Synthetic Packet Traffic Generative Adversarial Networks in Multi Agents With Peer-to-Peer and Global Priority Queue Generation | Request PDF - ResearchGate, accessed April 2, 2026, https://www.researchgate.net/publication/399748733_Synthetic_Packet_Traffic_Generative_Adversarial_Networks_in_Multi_Agents_With_Peer-to-Peer_and_Global_Priority_Queue_Generation
74. Provider or Deployer? Decoding the Key Roles in the AI Act - HÄRTING Rechtsanwälte, accessed April 2, 2026, <https://haerting.de/en/insights/provider-or-deployer-decoding-the-key-roles-in-the-ai-act/>
75. Logging - Agent Development Kit (ADK) - Google, accessed April 2, 2026, <https://google.github.io/adk-docs/observability/logging/>
76. Application Insights telemetry with Microsoft Copilot Studio - Dynamics 365, accessed April 2, 2026, <https://learn.microsoft.com/en-us/dynamics365/guidance/resources/copilot-studio-appinsights>
77. The EU AI Act's operator model: what every company needs to know before deploying AI, accessed April 2, 2026, https://medium.com/@ive_20203/the-eu-ai-acts-operator-model-what-every-company-needs-to-know-before-deploying-ai-9a58452aaf61
78. AI Act Service Desk - Article 12: Record-keeping - European Union, accessed April 2, 2026, <https://ai-act-service-desk.ec.europa.eu/en/ai-act/article-12>
79. Securing the Model Context Protocol (MCP): Risks, Controls, and Governance - arXiv, accessed April 2, 2026, <https://arxiv.org/html/2511.20920v1>