SALT

SOLUTION BRIEF • COMPETITIVE ANALYSIS

Salt Security vs. Fiddler AI

Fiddler AI measures how your AI models perform. Salt Security protects the APIs and MCP servers those models act on. AI observability and agentic security solve different problems for different buyers — and only one of them is a security platform.

CATEGORY	PLATFORMS COMPARED	YEAR
Agentic Security	Salt Security & Fiddler AI	2026

OVERVIEW

AI observability and API security are not the same discipline

Founded in 2018 as an ML model observability company, Fiddler AI has expanded into AI agent monitoring through its Trust Service — providing guardrails for LLM inputs and outputs, monitoring 100+ metrics including hallucination, toxicity, PII detection, and drift. Its primary buyers are data science and ML engineering teams that need to operationalize AI reliably at scale. That is a legitimate and important discipline. It is not API security.

Salt Security was built from the ground up as a security platform for the infrastructure that AI agents act on: the APIs, MCP servers, east-west traffic, and downstream services where agents execute real-world actions. Eight years of production API security research means Salt's behavioral models detect what Fiddler's observability metrics cannot: adversarial attack campaigns composed of individually valid, authorized API calls.

8

Years of dedicated API security research. Fiddler: MLOps observability platform with security features added.

0

Model instrumentation required. Salt covers the full API fabric without any LLM telemetry dependency.

0

Hallucination metrics needed to detect an API attack. Salt focuses on the attack surface, not model quality.

PLATFORM APPROACHES

MLOps observability vs. API fabric security — different tools for different problems.

SALT SECURITY

Purpose-built API and agentic security

Salt Security was built as a security platform, not an observability tool. The Agentic Security Graph maps and correlates every LLM connection, MCP server, API endpoint, identity, and data flow across code, cloud, and runtime — detecting adversarial attacks, business logic abuse, and data exfiltration in the infrastructure Fiddler's observability layer never monitors.

Salt Surface discovers external API and MCP exposure. Salt Collect monitors live runtime traffic across 70+ infrastructure technologies. Salt Code governs pre-production. Eight years of API behavioral research means Salt's anomaly detection understands what normal agent API behavior looks like and surfaces deviations with precision that no observability dashboard can replicate.

FIDDLER AI

ML model observability and LLM guardrails

Fiddler AI deploys its Trust Service to provide guardrails for LLM inputs and outputs, monitoring 100+ metrics including hallucination rate, toxicity, PII exposure, bias, and model drift. Its primary use case is helping ML engineering teams understand whether their AI models are performing reliably, fairly, and safely in production.

The constraint: Fiddler was designed to answer reliability questions, not security questions. Whether your model is hallucinating, drifting from baseline, or producing toxic outputs are operational quality metrics. Whether an attacker is using your agents to exfiltrate data through your API infrastructure is a security event — and no observability metric surfaces it.

THE COVERAGE GAP

Fiddler tells you if your AI model is performing correctly. Salt tells you if your API infrastructure is being attacked. Those are different questions, different platforms, and different buyers. Attackers exploiting AI agents do not cause hallucinations — they send valid, authorized API calls that look clean in every Fiddler metric while executing data exfiltration, privilege escalation, or unauthorized workflow execution in your API fabric.

CAPABILITY COMPARISON

Salt Security vs. Fiddler AI: side by side

CAPABILITY	SALT SECURITY	COMPETITOR
Unified Agentic Discovery	**STRONG** Discovers APIs, MCP servers, and AI-driven assets across external exposure, cloud, code repositories, and runtime as a unified environment.	**NOT IN SCOPE** Fiddler discovers AI models and agent applications within instrumented deployments. API, MCP, and infrastructure asset discovery across the full fabric is outside scope.
Agentic Security Graph	**STRONG** Correlates LLMs, MCP servers, APIs, identities, and sensitive data in one action-layer context — the only platform with this cross-fabric view.	**NOT IN SCOPE** Fiddler has no Agentic Security Graph. Cross-fabric correlation of LLM, MCP, API, and identity in a unified security model is not available.
Salt Code Governance	**STRONG** Governs API and MCP creation in repositories and developer workflows before risky logic ships to production.	**NOT IN SCOPE** Fiddler does not govern API and MCP creation in developer repositories.
Runtime-to-Code Remediation	**STRONG** Feeds runtime findings back into DevOps workflows and AI coding assistants to fix root causes — closing the loop between detection and fix.	**NOT IN SCOPE** Fiddler does not feed runtime findings back into DevOps workflows or AI coding assistants.
Identity-Aware Sequence Correlation	**STRONG** Tracks unique agentic identities and multi-step intent across sessions, tools, and services to detect campaigns, not just individual events.	**NOT IN SCOPE** Fiddler monitors model performance metrics. Identity-aware sequence correlation across API sessions and services is not available.
Behavioral Action-Layer Protection	**STRONG** Detects machine-speed business logic abuse beyond signatures, schemas, or known patterns.	**NOT IN SCOPE** Fiddler monitors model output quality. Behavioral detection of business logic abuse in downstream APIs is outside scope.
Internal & East-West Coverage	**STRONG** Protects internal APIs and downstream service interactions that edge-only and model-only tools miss entirely.	**NOT IN SCOPE** Fiddler monitors model-layer telemetry. East-west internal API traffic between infrastructure services is not monitored.
Action-Layer Data Security	**STRONG** Maps sensitive data in motion across APIs, MCP servers, and agent actions across the full fabric.	**PARTIAL** Fiddler detects PII and sensitive data in LLM inputs and outputs. Sensitive data in motion across the downstream API fabric is not mapped.

CAPABILITY	SALT SECURITY	COMPETITOR
Full API Fabric Security	**STRONG** Salt secures every API — internal, external, shadow, and third-party — regardless of which model or agent framework generated the call, with no model instrumentation required.	**NOT IN SCOPE** Fiddler monitors LLM and model application performance. API fabric security across the infrastructure agents interact with is outside the observability platform scope.
MCP Server Discovery and Governance	**STRONG** Salt discovers and governs MCP servers across code, cloud, and runtime — including rogue and shadow MCP servers created outside any monitored model application.	**NOT IN SCOPE** Fiddler monitors model inputs and outputs. MCP server discovery and governance across the infrastructure layer is not a product capability.
Agentic Attack Detection	**STRONG** Salt detects active adversarial attacks against your API fabric — distinct from model quality degradation, drift, or performance issues that observability platforms are designed to surface.	**NOT IN SCOPE** Fiddler is an observability platform designed to surface model quality and reliability issues. Detecting adversarial API attacks is a security discipline outside its product scope.

Questions to ask in your evaluation

The most productive evaluation question is not which platform has more features, but which platform covers the layer where your risk is highest. Use these scenarios to drive the conversation.

If you need to detect attacks, not measure performance

Fiddler's 100+ metrics — hallucination rate, toxicity, drift, bias — are quality and reliability measurements. An attacker using your agents to exfiltrate data will not cause any of those metrics to change. They will send authorized, expected API calls that look perfectly normal in every Fiddler dashboard. Salt was built to detect that category of attack. Fiddler was not.

If API and infrastructure security is in scope

Ask your Fiddler team what visibility they provide into the downstream API calls that your agents generate. If that visibility ends at the model output layer, you have identified the gap. Salt Surface can show you the full external exposure of your agentic API infrastructure today — with zero deployment — demonstrating coverage that Fiddler is not designed to provide.

If you are buying both Fiddler and Salt

Fiddler and Salt address genuinely different problems and have different buyers within the same organization. Fiddler is purchased by ML engineering teams focused on model reliability and responsible AI. Salt is purchased by security teams focused on API attack surface and agentic threat detection. These are complementary investments — and organizations that need both typically find that the security gap Salt fills is more urgent than additional observability depth.

If compliance obligations are driving the evaluation

AI governance frameworks and EU AI Act obligations that apply to model quality, bias, and responsible AI are squarely in Fiddler's domain. The security obligations under EU AI Act Articles 12, 14, 15, 20, 26, 72, and 73 — which apply to the action layer where agents call APIs and access data — are squarely in Salt's domain. Understanding which obligations are driving your compliance posture will clarify which platform applies.

Key differentiators

SALT SECURITY IS THE STRONGER CHOICE WHEN…

- API security and attack detection is the requirement — not model quality monitoring
- Coverage of the full API fabric that agents interact with is needed
- Agentic Security Graph correlation across LLM, MCP, and API layers is required
- No model instrumentation or LLM telemetry dependency is acceptable
- East-west and internal API infrastructure coverage is a requirement
- EU AI Act compliance at the action layer (Arts. 12, 14, 15, 20, 26, 72, 73) is in scope
- Eight years of dedicated API security research behind behavioral detection matters

FIDDLER AI MAY BE CONSIDERED WHEN…

Fiddler AI is appropriate for a specific and non-security use case:

- ML model performance monitoring, hallucination detection, bias analysis, and responsible AI observability are the primary requirements
- The buyer is an ML engineering or data science team focused on model reliability, not a security team focused on API attacks

For any organization evaluating agentic security for their API and infrastructure attack surface, Fiddler AI is not a substitute — it is a different product for a different buyer with a different problem.

THE COMBINED PICTURE

AI observability and API security are complementary disciplines that belong to different teams with different mandates. Fiddler AI helps ML teams know whether their models are performing reliably. Salt Security helps security teams know whether their API infrastructure is being attacked. Organizations that need both can buy both — but they should not mistake one for the other. The security gap that Salt fills is not addressed by any amount of observability investment.

SEE THE ACTION LAYER FOR YOURSELF

Salt Surface can show you the full external exposure of your agentic API infrastructure today — the APIs, MCP servers, and agent connections visible to an attacker — with zero deployment and no model instrumentation required. **Request a demo at salt.security**

