SOLUTION BRIEF • COMPETITIVE ANALYSIS

Salt Security vs. Zenity

Zenity monitors how AI agents behave within SaaS and endpoint environments. Salt Security secures the APIs, MCP servers, and downstream infrastructure where those agents execute — the action layer Zenity's platform-dependent model does not cover.

CATEGORY	PLATFORMS COMPARED	YEAR
Agentic Security	Salt Security & Zenity	2026

OVERVIEW

Agent governance within SaaS platforms is not the same as agentic security

Founded by Microsoft cloud security veterans and backed by Microsoft's M12, Zenity delivers strong coverage of agent activity within the Microsoft ecosystem — M365 Copilot, Azure AI, Teams agents, and supported enterprise SaaS platforms. Its step-level agent behavior analysis and incident correlation are genuine capabilities within that environment. What Zenity does not cover is the action layer: the APIs, MCP servers, and downstream services where agent behavior translates into enterprise risk outside SaaS platforms.

Salt Security covers the full API fabric regardless of platform, framework, or agent origin. The Agentic Security Graph maps and correlates LLM connections, MCP servers, APIs, identities, and data flows without dependency on which agent platform generated the interaction — covering custom LangChain workflows, Databricks agents, and every infrastructure connection that Zenity's platform-specific model never reaches.

0

Device agents or platform connectors required. Salt covers the full API fabric with no endpoint deployment.

100%

Platform-agnostic coverage — LangChain, Databricks, custom agents, and beyond Zenity's supported list.

8

Years of production API security research. Zenity: SaaS-platform governance depth.

PLATFORM APPROACHES

Platform-dependent SaaS monitoring vs. platform-agnostic API fabric security.

SALT SECURITY

Full API fabric coverage across every platform and framework

Salt Security covers the action layer below and beyond what Zenity monitors: every API, MCP server, and downstream service that agents interact with — regardless of whether the agent runs in Microsoft, a custom LangChain workflow, Databricks, or any other framework. The Agentic Security Graph is platform-agnostic by design.

No device agent required. No platform connector to configure. Salt Surface discovers external exposure from the internet out. Salt Collect monitors live runtime traffic across 70+ technologies. Coverage extends to every custom agent framework and every downstream API that Zenity's supported platform list does not include.

ZENITY

AI agent governance across SaaS and endpoint environments

Zenity monitors agent activity at the step level within the platforms it natively supports: Microsoft 365, Azure AI Foundry, Salesforce, ServiceNow, ChatGPT Enterprise, and similar enterprise SaaS platforms. Its Correlation Agent connects incident signals across identity and posture. A device agent extends coverage to browser-based agentic activity.

The constraint: Zenity's coverage is explicitly scoped to supported platforms. Custom agent frameworks including LangChain, CrewAI, and Databricks are outside its coverage. Downstream enterprise APIs and east-west infrastructure — the fabric that SaaS agent actions call into — are not monitored.

THE COVERAGE GAP

Zenity tells you what your SaaS-platform agents are doing within supported platforms. Salt tells you what those agents — and every other agent in your environment — are doing to your enterprise API fabric. The downstream APIs that agents call, the MCP servers created outside SaaS governance processes, and the east-west infrastructure connections that never surface at the SaaS layer are all visible to Salt and invisible to Zenity.

CAPABILITY COMPARISON

Salt Security vs. Zenity: 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.	**PARTIAL** Zenity discovers agents within its supported SaaS platforms. Custom agent frameworks, external APIs, and infrastructure connections outside supported platforms are not in 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 AVAILABLE** Zenity has no Agentic Security Graph. Cross-fabric correlation of LLM, MCP, API, and identity in a unified 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 AVAILABLE** Zenity does not govern API and MCP creation in developer repositories pre-production.
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 AVAILABLE** Zenity does not feed runtime findings back into DevOps workflows or 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.	**PARTIAL** Zenity correlates incident signals across its supported platform environment. Identity-aware sequence correlation across downstream API calls outside SaaS platforms is not available.
Behavioral Action-Layer Protection	**STRONG** Detects machine-speed business logic abuse beyond signatures, schemas, or known patterns.	**PARTIAL** Zenity detects behavioral anomalies within supported SaaS platform interactions. Business logic abuse in downstream enterprise APIs is outside current scope.
Internal & East-West Coverage	**STRONG** Protects internal APIs and downstream service interactions that edge-only and model-only tools miss entirely.	**NOT AVAILABLE** Zenity monitors agent behavior within supported SaaS platforms. East-west internal API traffic and downstream service interactions are not covered.
Action-Layer Data Security	**STRONG** Maps sensitive data in motion across APIs, MCP servers, and agent actions across the full fabric.	**PARTIAL** Zenity monitors data access within supported platform interactions. Sensitive data in motion across the

CAPABILITY	SALT SECURITY	COMPETITOR
		downstream API fabric and east-west infrastructure is not mapped.
Full API Fabric Coverage	**STRONG** Salt secures every API — internal, external, third-party, and shadow — regardless of which platform or agent framework generated the interaction.	**NOT AVAILABLE** Zenity's coverage is scoped to supported SaaS platforms. Enterprise APIs outside those platforms, east-west microservice traffic, and downstream infrastructure are not covered.
Platform-Agnostic Agentic Coverage	**STRONG** Salt covers every agent framework — LangChain, CrewAI, Databricks, custom-built agents — with no dependency on which SaaS platforms the agents are deployed within.	**LIMITED PLATFORM SUPPORT** Zenity explicitly supports a defined list of SaaS platforms. Custom agent frameworks including LangChain and Databricks are outside current coverage scope.
No Device Agent Required	**STRONG** Salt delivers full agentic security coverage without deploying a device agent to any endpoint — fully out-of-band with zero endpoint instrumentation dependency.	**DEVICE AGENT REQUIRED** Zenity requires a device agent deployed to endpoints to provide agentic browser coverage, adding a deployment and management dependency.

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 your agents run on custom frameworks or beyond Microsoft

Zenity's coverage is deepest for Microsoft-ecosystem agents. If any of your agent deployments use LangChain, CrewAI, Databricks, or custom-built frameworks, ask Zenity to demonstrate what they see for those agents specifically. For Salt, platform and framework make no difference — coverage extends to every API connection regardless of source.

If downstream API visibility is needed

Ask Zenity to show you the enterprise APIs that your SaaS agents are calling — not the agent steps, but the downstream API traffic those steps generate. If that visibility is outside Zenity's scope, you have identified the gap where Salt adds coverage. The downstream API layer is where business logic abuse at enterprise scale occurs.

If device agent deployment is a concern

Zenity requires a device agent for endpoint-based agentic browser coverage. In environments with strict endpoint management policies, large device fleets, or complex change management requirements, that deployment dependency adds friction. Salt has no device agent requirement — coverage is fully out-of-band.

If you have agents outside supported SaaS platforms

Enterprise AI deployments increasingly span SaaS platforms, cloud-hosted agents, and custom-built frameworks. Zenity's platform-specific coverage model requires each environment to be on its supported list. Salt's coverage has no such dependency — it monitors the API fabric regardless of where the agent originated.

SUMMARY

Key differentiators

SALT SECURITY IS THE STRONGER CHOICE WHEN…

- Full API fabric coverage across all platforms and agent frameworks is required
- Coverage of custom agent frameworks beyond Zenity's supported SaaS platform list is needed
- No device agent deployment to endpoints is acceptable
- East-west and internal API infrastructure coverage is required
- Downstream API business logic protection is a priority
- Platform-agnostic agentic security with no vendor lock-in to Microsoft ecosystem is needed
- Eight years of production API security research behind behavioral detection matters

ZENITY MAY BE CONSIDERED WHEN…

Zenity's platform-specific coverage may be appropriate when:

- All agent deployments are exclusively within Zenity's supported Microsoft and enterprise SaaS platform list with no custom frameworks or external API dependencies
- Step-level agent behavior monitoring within SaaS platforms is the specific requirement with no downstream API security needs

For any organization with agents running on custom frameworks, legacy infrastructure, or downstream API dependencies outside SaaS platforms, Zenity's coverage will not extend to the primary risk surface.

THE COMBINED PICTURE

SaaS agent governance and API fabric security protect different surfaces. Zenity delivers strong step-level monitoring within its supported Microsoft and SaaS platform environment. Salt secures the full API fabric — every downstream connection, every custom framework agent, every east-west infrastructure interaction — regardless of which platform generated it. In organizations with diverse agent deployments, the coverage that Zenity cannot provide is the coverage that matters most.

SEE THE ACTION LAYER FOR YOURSELF

Salt Surface can scan your full agentic environment today and show you every API your agents are connecting to — including the ones running on frameworks outside Zenity's supported platform list — with zero deployment required. **Request a demo at salt.security**