akto + LiteLLM

Datasheet

Secure AI Agents Routed through LiteLLM

Catch the AI attacks and risks that LiteLLM ‘s guardrail’s miss



Akto runs inside the LiteLLM proxy as a chained guardrail. LiteLLM's built-in checks handle the fast, deterministic rules. Akto adds model-based detection for prompt injection, semantic data leaks, and intent-level policy violations.

Why LiteLLM's built-in guardrails aren't enough

The built-in content filter runs regex and keyword matching, and LiteLLM's docs say it's for cases that don't need an ML model. Prompt injection and data leaks usually do.

Patterns miss intent

LiteLLM's content filter is for cases that don't need an ML model. A probe spread over ten harmless-looking turns needs one.

Agents act mid-task

Agents call tools and MCP servers between model calls. Each of those calls needs its own policy check and log entry.

Rules cover known attacks

Guardrails catch the threats you already know about. Akto also red teams your agents to find the rest before launch.

RISK	GUARDRAILS BUILT INTO LITELLM	AKTO AI GUARDRAILS
Prompt injection	Fuzzy match on known phrases; optional LLM based checks	Blocks injection attempts by their intent, however they're worded
Sensitive data	Regex, keyword lists, and Presidio entity detection you host yourself	Model-based detection of semantic PII leaks, judged in context
Secrets and keys	Secret detection and redaction on the Enterprise tier	Blocks exposed keys and credentials with a Secrets policy, on any LiteLLM tier
Custom policies	Limited to banned words and custom regex, which can't judge intent	Enforces custom prompt based intent-level policies you manage centrally in the Akto dashboard

```
guardrails:
  - guardrail_name: "akto-validate"
    litellm_params:
      guardrail: akto
      mode: pre_call
      akto_base_url: os.environ/AKTO_GUARDRAIL_API_BASE
      akto_api_key: os.environ/AKTO_API_KEY
      default_on: true
      unreachable_fallback: fail_closed   # optional: fail_open | fail_closed (default: fail_closed)
      guardrail_timeout: 5              # optional, default: 5
      akto_account_id: "1000000"        # optional, env fallback: AKTO_ACCOUNT_ID
      akto_vxlan_id: "0"                # optional, env fallback: AKTO_VXLAN_ID
```

How it works

1 **Apps and agents call LiteLLM**

No code changes. LiteLLM's own rules still run.

2 **Akto checks before the model**

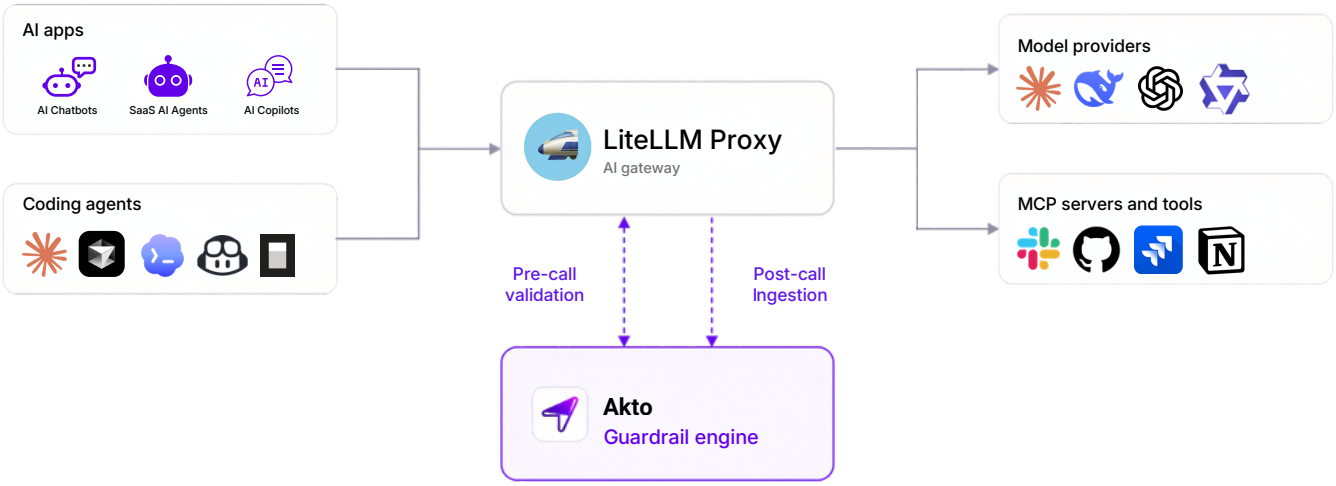
Requests that break a policy get a 403.

3 **LiteLLM routes clean traffic**

On to 100+ model providers or MCP servers.

4 **Akto records every exchange**

Prompt, response, and verdict, for alerts and audit.



Red team the agents behind LiteLLM

Guardrails stop known attacks at runtime. Red teaming finds the weaknesses you don't know about yet, before an attacker does.

Probes built for agents

4000+ probes cover prompt injection, tool misuse, data leakage, policy evasion, and multi-turn exploits across the whole agent workflow.

Full OWASP Top 10 Coverage

Stress test your AI agents with full coverage of the OWASP Top 10 for LLMs, Agentic Applications, and MCPs.

Check if your guardrails hold

Run attacks against your agents and confirm your Akto guardrail policies block them before you ship.

Setup Guide

https://ai-security-docs.akto.io/akto-argus-agentic-ai-security-for-homegrown-ai/connectors/ai-agent-security/litellm



Akto Named a Pioneer in Gartner®'s Emerging Market Quadrant for AI Security