AI CODING AGENTS AND MCP GUARDRAILS · DATASHEET

Safeguard

Let coding agents work, and decide what they may touch.

Guardrails for the traffic to your models, controls over which MCP tools an assistant can call, and permission modes for the Safeguard CLI agent.

THE PROBLEM

Coding agents read untrusted content, call tools and run commands on developer machines and in CI. One poisoned document or one over-broad tool grant is enough for an agent to leak data or change something nobody approved, and most teams cannot say which agent holds which access.

WHAT IT DOES

AI Gateway guardrails

An LLM firewall for prompt injection, jailbreak and PII egress, in monitor mode first and then enforce, with fail-open safety for your app.

MCP connectors, gated

Install Safeguard from the Claude connector directory. Administrators choose which tools an assistant can see and call, and a new workspace starts read-only.

CLI permission modes

The Safeguard CLI agent asks before it changes anything by default (supervised), with separate modes for edits, plan, guarded, strict and autonomous.

Agent identity

Track which agent holds which access, at what level, so an audit can tell a person from an agent.

HOW IT WORKS

1. **Agent works** — Prompts, retrieved content, tool calls
2. **Inspect** — AI Gateway checks prompts and responses
3. **Gate tools** — MCP tools enabled per tenant
4. **Mode** — CLI permission mode decides what asks
5. **Record** — Guardrail events, redacted metadata

MODES ARE SEPARATE, NOT A LADDER

Supervised, edits, plan, guarded, strict and autonomous each fit a different job. Pick one per task. Autonomous is refused outside a sandbox runner unless you pass an explicit override.

CAPABILITIES IN DETAIL

AI Gateway	Inline proxy or SDK shim. Checks direct and indirect prompt injection, jailbreak and PII egress. Monitor is the default, enforce is per tenant, and an error fails open with an alert.
MCP server	mcp.safeguard.sh, listed in the Claude connector directory and the official MCP Registry. OAuth 2.0 with PKCE.
Tool gating	Per-tenant feature flags govern both discovery and execution. A tool that is off is not advertised and cannot be called.
Supervised (default)	Reads run. Every change and command asks first.
Edits and plan	Edits: workspace edits and allow-listed commands run, other commands ask. Plan: read-only, changes refused.
Guarded and strict	Guarded: read-only and policy-allowed commands run, the rest ask. Strict: pre-approved only, everything else denied, which suits CI.
Autonomous	Nothing asks. Refused outside a sandbox runner unless you pass an explicit override.

INPUTS

- Prompts, retrieved content and tool output
- Your tenant policy and enabled tool set
- A policy file with the CLI mode
- Your model provider

OUTPUTS

- Allow, redact or block verdicts
- Guardrail events with redacted metadata
- Tool calls limited to what your tenant enabled
- AI security findings in the unified queue

INTEGRATIONS AND SURFACES

Claude · ChatGPT · Cursor · Other MCP clients · Claude Code · MCP Guardrail · AI Gateway · Agent identity

DEPLOYMENT AND DATA HANDLING

Privacy

Guardrail events carry redacted metadata only. Raw prompt and response text is never persisted.

Local runner

The CLI scans on your machine and sends only encrypted results. No open inbound connections; it polls.

Regions and hosting

SaaS in US, EU (Frankfurt) and Gov regions. Private VPC, on-prem and fully air-gapped deployments are available.

Model choice

Zero AI mode runs the platform deterministically with no LLM. Or bring your own model (Claude, OpenAI and others), or run Safeguard models inside your environment.

See MCP and agent security

safeguard.sh/use-cases/mcp-server-security

Scan the code or type the address. Questions: hi@safeguard.sh

Safeguard

SOC 2 Type I and Type II audited; reports available on request (safeguard.sh/trust). Designed for FedRAMP High.