ENTERPRISE PILOT & TRUST PACK

SigmaShake

Control what AI coding agents can do before they do it.

Runtime policy, human approvals, and audit trails for teams adopting Claude Code, Codex, Cursor, Copilot, Gemini, and MCP-based tools.

This pack is written for CISOs, VPs of Engineering, platform leaders, and security reviewers evaluating a 30-day SigmaShake pilot.

Contact
sales@sigmashake.com
security@sigmashake.com

Web
sigmashake.com
docs.sigmashake.com

Prepared
July 2026
Version 1.1

Call to action
Schedule a 30-minute scoping call.

Executive Summary

What is SigmaShake?

SigmaShake is a runtime control layer for AI coding agents. It blocks or approval-gates agent actions that match configured policy, and records every decision in an audit trail your team can review.

SSG is the engine that powers SigmaShake's local enforcement layer. It evaluates an agent's tool call before execution and applies your organization's policy in real time.

WHO IT IS FOR

Security, engineering, and platform teams that already use, or are about to approve, AI coding agents in real development workflows.

WHAT IT SOLVES

AI agents can run commands, edit files, install packages, access secrets, and interact with infrastructure. Most teams do not have a consistent pre-execution control layer for those actions.

WHY NOW

AI agents are moving from autocomplete to autonomous execution. SigmaShake helps teams adopt AI agents safely.

> **Plain-English value:** SigmaShake lets engineering teams use AI agents without giving those agents unchecked access to shells, files, secrets, package managers, infrastructure, and production systems.

The problem

WHAT SECURITY TEAMS CANNOT SEE TODAY

- Which AI agent attempted a risky command.
- Which files, secrets, tools, or infrastructure it tried to touch.
- Whether the action was blocked, approved, allowed, or redirected.
- Which rule or policy made that decision.
- Which human approved an exception.

WHAT THEY CANNOT CONSISTENTLY CONTROL

- Destructive shell commands.
- Credential and environment-file access.
- Persistence mechanisms such as startup jobs and scheduled tasks.
- Supply-chain risk from package installs and plugin changes.
- Cross-tool policy drift across Claude Code, Cursor, Codex, Copilot, Gemini, and MCP hosts.

Solution and Architecture

How SigmaShake helps

SigmaShake places a deterministic policy gate at the tool-call boundary. Before an AI coding agent takes an action, SigmaShake evaluates the request against local policy and returns a decision.

BLOCK

Stop actions that violate policy, such as destructive commands or credential access.

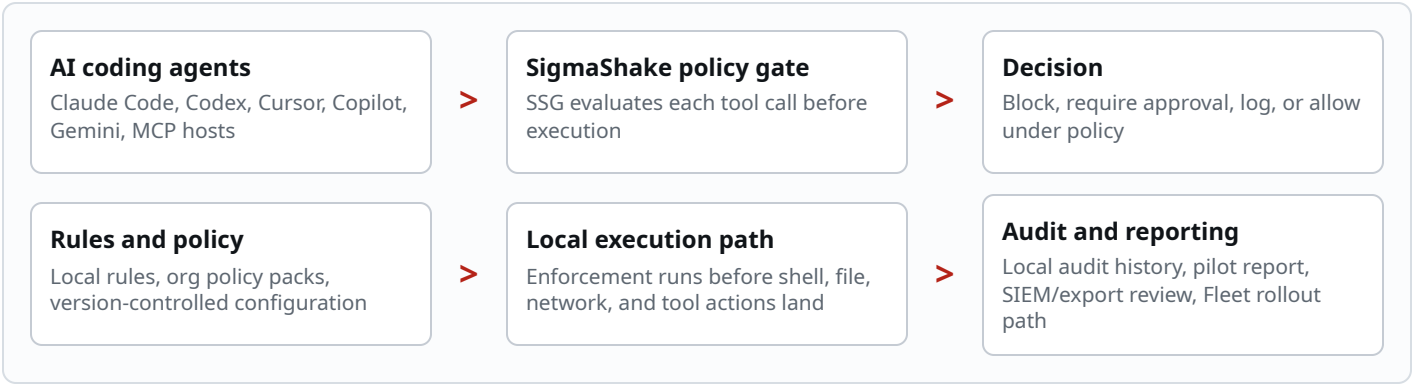
APPROVE

Hold policy-matched actions for human review when policy requires judgment.

LOG

Record decisions so security and engineering can reconstruct what happened.

Simple architecture



A pilot can start local-first with one engineering team. No org-wide rollout is required before the buyer has evidence that the controls fit the workflow.

What changes for security and engineering

BEFORE SIGMASHAKE	WITH SIGMASHAKE
Agent actions are governed differently across tools, or not governed at all.	One policy layer applies across supported AI coding agents.
Security reviews agent activity after the fact, if logs exist.	Actions that match configured policy are blocked or approval-gated before they execute.
Approvals happen ad hoc in chat, PR comments, or not at all.	Approval decisions are tied to the action, rule, timestamp, and reviewer.
Procurement asks for proof and the team has anecdotes.	The pilot produces audit samples, blocked-risk examples, policy gaps, false positives, and a rollout plan.

30-Day Enterprise Pilot

Pilot objective

The pilot proves whether SigmaShake can govern real AI-agent activity in your environment without blocking legitimate engineering work. It is designed to start with one team, one set of tools, and a baseline policy that can be tuned while engineers keep working. Access is scoped and provisioned under a pilot agreement after scoping, so governed use remains authenticated and controlled.

What is included

DEPLOYMENT AND POLICY

- Fast-start SigmaShake deployment for one production engineering team.
- Runtime controls for supported AI coding agents.
- Baseline policies for destructive commands, credential access, persistence, supply-chain risk, and defense evasion.
- Policy tuning for the team's actual workflow.

EVIDENCE AND REPORTING

- Human approval workflow for policy-matched actions that require review.
- Local audit trail and decision history.
- SIEM/export review for the buyer's security stack.
- Buyer-run test cases and blocked-risk examples reviewed against policy.
- Weekly review and final executive report.

Success criteria

- A production engineering team uses AI coding agents under SigmaShake controls throughout the pilot.
- Actions that match configured policy are blocked or approval-gated before execution.
- Developers can keep using AI agents with acceptable friction.
- Security can answer who did what, what was blocked, which rule applied, and who approved exceptions.
- The pilot produces audit logs, blocked-risk examples, false-positive findings, policy gaps, and a clear rollout plan.

Fast-start 30-day timeline

PHASE	WORK	OUTPUT
Day 0-1	Confirm pilot team, AI tools, repositories, risk boundaries, and approval owner.	Scope locked and baseline policy selected.
Days 1-3	Install SigmaShake for the pilot team and apply baseline policy.	Governed agent usage begins in the first week.
Days 4-20	Run in normal engineering workflow, review approvals, collect audit evidence, identify false positives.	Weekly findings and policy changes.
Days 21-27	Review buyer-run test cases, blocked-risk examples, and SIEM/export needs.	Evidence package and rollout gaps.
Days 28-30	Executive readout and expansion planning.	Final report and rollout recommendation.

Security Architecture and Compliance Status

Security architecture

LOCAL-FIRST ENFORCEMENT

Policy evaluation happens on the developer machine before the action executes. No LLM is required in the enforcement path.

VERSIONED POLICY

Rules are text-based and can be reviewed, versioned, and rolled out like other engineering policy.

APPROVAL WORKFLOW

Risky actions can be paused for human approval instead of silently proceeding or being blocked without context.

AUDIT RECORD

Decisions are recorded so the team can reconstruct agent actions during a review, postmortem, or compliance process.

Current status and roadmap

AREA	STATUS	NOTES FOR REVIEWERS
Runtime policy gate	Available	SSG powers local pre-execution policy enforcement for supported AI coding agents.
Desktop shells	Available	Native Linux, macOS, and Windows shells supervise the local daemon and dashboard.
Fleet rollout	Available for pilot planning	Fleet provides the enterprise policy and device rollout path. Pilot scope determines what is enabled.
SSO and SIEM	Pilot-scoped	SSO and SIEM/export requirements are reviewed during scoping and included where needed for the pilot.
Security policy and disclosure	Available	Security contact and responsible disclosure process are available at security@sigmashake.com and security.sigmashake.com.
SOC 2	Readiness, not certified	SOC 2 Type II observation window is in progress. No final SOC 2 Type II report should be assumed until issued by an independent CPA firm.
Compliance evidence automation	Available internally	SigmaShake maintains evidence automation and audit-package tooling. Buyer-facing evidence is provided during vendor review.
Third-party assurance	Expanding	Additional third-party validation, procurement packet improvements, and continuity materials are roadmap items.

Transparency note: SigmaShake should be evaluated on what is deployed and evidenced today. We do not claim SOC 2 certification, FedRAMP authorization, EDR replacement, SIEM replacement, or that AI agents are safe by default.

Pilot and Deployment

Pricing

30-DAY GUIDED PILOT

No services fee

No pilot services fee for qualified teams after scoping. Runtime access is authenticated and provisioned under a pilot agreement; broader rollout is scoped after evidence.

AFTER THE PILOT

Scoped

If the pilot succeeds, SigmaShake will propose a rollout based on users, teams, support requirements, SSO/SIEM needs, and Fleet scope.

The pilot is intentionally low-friction: prove value with one production team before broader procurement or rollout planning. No anonymous product trial is implied; every governed runtime use remains access-controlled.

Deployment process

1. **Scoping call:** identify pilot team, AI tools, repositories, risk boundaries, and approval owner.
2. **Fast start:** install SigmaShake for the pilot team and apply baseline controls for destructive commands, credential access, persistence, package installs, and other sensitive actions.
3. **Confirm governance:** verify supported tools are governed and developers can keep working.
4. **Run in production workflow:** collect real decisions, approval events, blocked-risk examples, and false positives.
5. **Tune and measure:** update policy, review buyer-run test cases, and review SIEM/export needs.
6. **Executive readout:** deliver the final report and expansion recommendation.

FAQs

IS SIGMASHAKE AN EDR OR SIEM REPLACEMENT?

No. SigmaShake is a governance and audit layer for AI-agent actions before they execute. It complements EDR, SIEM, code scanning, secret scanning, and normal review workflows.

DOES THIS STOP ALL AI-AGENT RISK?

No. It reduces a specific class of risk: tool calls that violate policy before they execute. It does not make AI agents safe by default and is not a sandbox.

DOES ENFORCEMENT REQUIRE A CLOUD SERVICE?

No. The core enforcement path is local-first. Enterprise services can support licensing, policy distribution, Fleet rollout, and reporting, depending on the deployment.

WHICH AGENTS ARE IN SCOPE?

The pilot is scoped around the buyer's actual tools. SigmaShake is designed for AI coding agents such as Claude Code, Codex, Cursor, Copilot, Gemini, and MCP-based hosts.

WHAT DOES SECURITY GET AT THE END?

A concise report with governed usage, blocked-risk examples, approval events, audit samples, false positives, policy gaps, and a rollout recommendation.

WHAT DOES ENGINEERING GET AT THE END?

A practical answer to whether the team can keep using AI coding agents with acceptable friction under a consistent policy layer.

CAN WE START WITH ONE TEAM?

Yes. The pilot is designed to start with one production engineering team, then expand only if the evidence supports rollout.

CAN WE RUN OUR OWN TESTS?

Yes. The pilot can include buyer-run test cases against agreed policy boundaries so security can validate outcomes directly.

HOW DO WE DECIDE ROLLOUT READINESS?

Rollout is recommended only if the controls fit engineering workflow, govern the target tools, produce useful audit evidence, and meet the buyer's security requirements.

Schedule a 30-minute scoping call.

Email sales@sigmashake.com with your pilot team, AI coding tools in use, and the security owner who should join.