

Gemini CLI and Codex CLI

Both are handled by the standard adapter path: detected automatically, connected by **Govern all detected tools** or individually from their card, and disconnected the same way.

Connecting

Tools, find the card, **Connect**. Then **restart your terminal** — like every other command-line tool here, they read their environment at startup.

Connection mode

Both appear in **Policy ? Connection modes** in the admin console, where an admin chooses whether they run on the developer's own subscription or on the organization's managed provider key.

Codex CLI and posture

Codex CLI has settings that materially change how much autonomy the agent has, and the Endpoint Suite reports on them:

- `approval_policy: never` — the agent never asks before acting.
- `sandbox_mode: danger-full-access` — the sandbox is effectively off.
- `sandbox_workspace_write.network_access: true` — the agent can reach the network from inside its workspace sandbox.

These are reported, not blocked. They appear as **posture findings** on the device in the admin console's Diagnostics screen, so an admin can have a conversation about them. See [Posture findings](#).

🛡️ Govern all detected tools

3 tools detected · all routed through Sentilai.

🛡️ All detected tools governed

- ✔️ Claude Code: already governed
- ✔️ Cursor: already governed
- ✔️ GitHub Copilot (CLI): already governed
- ✔️ Gemini CLI: already governed
- ✔️ Local MCP servers: already governed

Routes Claude Code, Cursor and the Copilot CLI in one step. You can still connect or disconnect each individually below — and each still needs a restart to pick up the change. Copilot for VS Code is set up separately (guided) below.

Gemini CLI and Codex CLI are governed through “Govern all detected tools”, which names them as it goes. Unlike Claude Code, Cursor and Copilot, neither has a card of its own on the Tools tab.

Revision #16

Created 2026-08-02 10:19:40 UTC by Sentilai Docs

Updated 2026-08-14 12:39:52 UTC by Sentilai Docs