Subscription or managed key

Every governed request runs in one of two **modes**, and the difference is who pays.

Subscription

The developer's own AI subscription pays for the request. Claude Code with a personal or company Claude plan is the common case. Sentilai governs and audits the request but does not touch billing.

Use this when your developers already have subscriptions you are happy with, and you want governance rather than centralized spend.

Managed API key

The request goes out on **your** provider key, from **Providers**, and is billed to your organization. You get one bill, per-model control through the key's enabled-models list, and the ability to cut off access by removing the key.

Use this when you want the spend centralized or when the tool cannot carry a subscription login.

Setting it

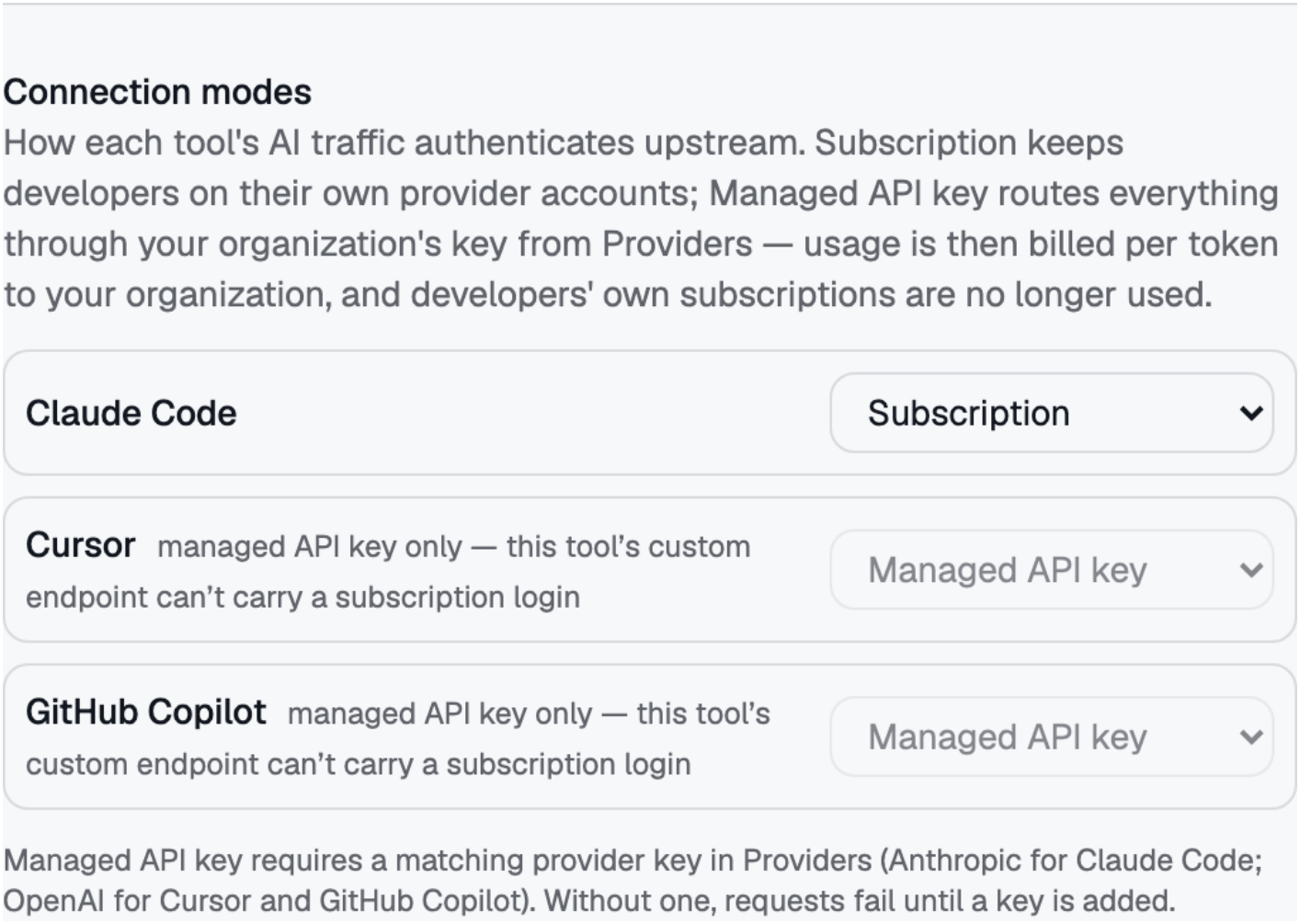
Policy ? Connection modes, per tool. Claude Code, Cursor, GitHub Copilot, Gemini CLI and Codex CLI each get their own setting.

Cursor is locked to managed API key — its custom-endpoint mechanism cannot carry a subscription login. If you want Cursor governed, you need an OpenAI provider key.

What goes wrong

- Managed mode with no matching provider key: requests fail. The tool shows a provider error and Activity shows the request with an upstream error. Add the key.
- Managed mode with two keys of the same type whose model lists overlap: Sentilai cannot tell which key you meant and refuses the request. Make each key's **enabled models** list distinct.

Both modes are audited identically. In **Activity**, the **Mode** column tells you which one a given request used.



Policy ? Connection modes is where the subscription-or-managed-key choice is made, per tool.

Revision #12
Created 2026-08-02 10:19:50 UTC by Sentilai Docs
Updated 2026-09-29 14:39:27 UTC by Sentilai Docs