

Two keys of the same type

You can have several keys of the same provider type — a cheap one for general use, an expensive one for a specific team. Sentilai picks between them using the **enabled models** list.

The rule

Sentilai looks at the model the request asks for and finds the key whose enabled-models list contains it.

The failure

If a model appears in **more than one** key's list, the request is ambiguous and is refused with a conflict error rather than a guess. Guessing which of your keys to bill would be the wrong behaviour for something that costs money.

Fixing it

Make the lists disjoint. Each model should appear in exactly one key's enabled-models list.

If you genuinely want two teams on the same model with separate billing, that is not something the enabled-models mechanism can express today — tell us, because it is a reasonable thing to want.

Keys are write-only

After saving, a key is shown only as `....` and its last four characters. Editing a key and leaving the field blank keeps the stored value; you never need to retrieve the original just to change the model list.

Removing a key stops every tool that depends on it, and the confirmation says so.

Revision #5

Created 2026-08-02 10:20:07 UTC by Sentilai Docs

Updated 2026-08-02 16:29:25 UTC by Sentilai Docs