

[• AI MODELS](#)

Which AI models does SumizAI work with?

SumizAI does not sell you tokens. You connect an API key you already have, to a provider you already chose, and the application talks to it on your behalf. **Seven providers are supported**, including one that runs entirely on your own machine.

Updated: 2026-08-14 8 questions 4 min read

Q. Which providers can I connect?

Anthropic, OpenAI, Gemini, Groq, OpenRouter, a local Ollama, and a custom server of your own.

Under the hood there are two clients rather than seven. Anthropic has its own, and the other six speak the OpenAI protocol, so one implementation covers them. That is why adding a self-hosted endpoint works at all — if it speaks that protocol, it is already supported.

You can connect more than one and choose which to use for a given conversation, with the cost tier shown next to the model in the header.

Q. Why do I have to bring my own key?

Because the alternative is a markup. If the application supplied the model, the price would have to include a margin on every token you generate, and a dollar a month would be impossible.

It also means the commercial relationship for the expensive part is between you and your provider. You see your own usage, on your own dashboard, at your own rate — including any credits or committed-use discount you already have.

And it is the reason a local Ollama is a first-class option. If your key is the boundary, so is your choice to have no external provider at all.

Q. Where does the list of models come from?

From the provider, asked with your key, cached for fifteen minutes.

There used to be a catalogue in the code and it was quietly wrong: it would offer a model the provider had already retired, and you would find out when a note failed to generate. Now an unconnected provider shows an empty list, and a connected one shows what it actually has.

Pasting a key is enough to get started. The application asks the provider with that key and takes the strongest model it offers, and a bad key fails immediately with the provider's own reason rather than being saved and breaking later.

Q. How is the list ordered?

Free models above paid ones, and within each group strongest first — our quality assessment, then prompt price, then context window, then alphabetically. The interface shows the two groups and does not re-sort them.

This exists because of OpenRouter, where the list runs to hundreds of models. At that scale the useful first question is “what do I have for free”, and only then “what is the best thing worth paying for”.

If you only ever connect one provider you will barely notice it. If you connect OpenRouter it is the difference between a usable dropdown and an unusable one.

Q. Does everything use the same model?

No, and that is a cost decision. There is a main model and an auxiliary one.

The main model answers you and, when the table of contents is rebuilt, lays out the chapters. The auxiliary one does the smaller structured jobs: drafting the title and summary, and scoring whether a new note duplicates an existing one.

Duplicate scoring in particular runs on every note against several candidates, so putting it on a cheaper model is the difference between a feature you leave on and one you turn off.

Q. Can I tell which provider is actually fastest for me?

Yes — there is a speed test in settings that measures the real response time of each connected provider using its own model.

Published benchmarks are averages over other people's network conditions. Your latency to a given provider from where you actually sit is a different number, and it is the one that determines whether the app feels quick.

It is worth re-running occasionally. Providers change, and so does your connection.

Q. What happens when the provider fails?

You get a specific reason and a retry in place, rather than a generic error.

The application distinguishes an invalid key, a rate limit and a provider outage, and says which one happened. Your question was written down before the call was made, so it is still there — the failed turn is marked and you retry it without retyping.

There is a detail specific to the desktop application: calls go out from your machine, so the provider sees an origin header. Anthropic rejects those without an explicit opt-in header, and the rejection *looks* like an authentication failure. The application sends the right header; it is worth knowing because if you build something similar yourself, that error message will send you hunting for a key problem you do not have.

Q. Is my material used to train anything?

Not by SumizAI. There is no training pipeline here; your notes are files on your disk.

What your provider does with the API traffic is governed by your agreement with that provider, which is one more argument for the key being yours: you can read those terms, choose accordingly, and change provider without changing note-taking application.

If you want the question closed entirely, connect a local Ollama. Nothing leaves the machine.