amaad.

My AI agent setup

How I run one AI agent that hands work to Claude and GPT, on the subscriptions I already pay for. By @byamaad.

The idea

I don't want five AI apps open. I want one agent I can talk to that already knows what I'm working on, decides which AI should do each job, and can't do anything risky without asking me.

That agent is Hermes Agent, free and open source from Nous Research.

- Code: github.com/NousResearch/hermes-agent
- Docs: hermes-agent.nousresearch.com/docs

The setup prompt on byamaad.com/agent does everything below for you. This guide explains what each part is and why it's there.

---

1. One brain, on your subscription

Hermes runs on Claude (Opus) through my Claude subscription, using the official Claude Subscription plugin, so there's no API key and no per-token bill. If you have ChatGPT instead, it signs in with your ChatGPT account the same way.

---

2. Worker lanes

When I ask for something, Hermes splits the job and hands each part to the right model:

LANE	WHAT IT DOES	MODEL
fast	reading files, summaries, lookups	Haiku (cheap and quick)
standard	research, writing, builds from a clear spec	Sonnet
code	writing and fixing code	GPT through Codex
deep	planning, reviewing, hard calls	Opus

The main agent keeps the thinking and the final call. Parts that don't depend on each other run at the same time.

---

3. The brief rule

Every worker starts knowing nothing. It can't see your chat. So Hermes writes each one a full brief: the goal, the exact files, what done looks like, and where to save the result. That's the part that makes the whole thing work. AI prompting AI, and it writes better prompts than I would.

---

4. A brain folder

One folder holds what the agent needs to know about you:

```
~/Brain/
  README.md      who you are, what you're working on, how you like answers
  decisions.md   every decision with the date and why
  projects/      one file per project
  inbox/         loose notes
```

Rule: the agent reads the README before it touches anything for you, so you never explain yourself twice.

---

5. Memory

Hermes remembers across sessions on its own. When you correct it, it saves the correction. When you do the same workflow twice, it saves it as a skill, a written procedure it follows next time. Optional: Honcho, a deeper memory service, if you want it to build a real picture of you over time.

6. The approval gate (the most important part)

A small script runs before every command and every connected-app action. If it looks like it could spend money, send something, publish, deploy or delete, Hermes has to stop and ask you first. If you say no, it isn't allowed to find another way around it. It reads everything; it changes nothing risky without you.

The setup prompt tests it: it asks Hermes to delete a junk folder, and the gate stops it.

Starter prompts

- "Read my brain folder and tell me what you know about me. Ask me what's missing."
- "Research the top 5 tools for [thing], have the fast lane read their docs, and give me a one-page pick."
- "This decision is expensive. Have two different models look at it separately and show me where they disagree."
- "Save what we just did as a skill so you do it the same way next time."

Follow @byamaad. I'm breaking down every part of this on my page.