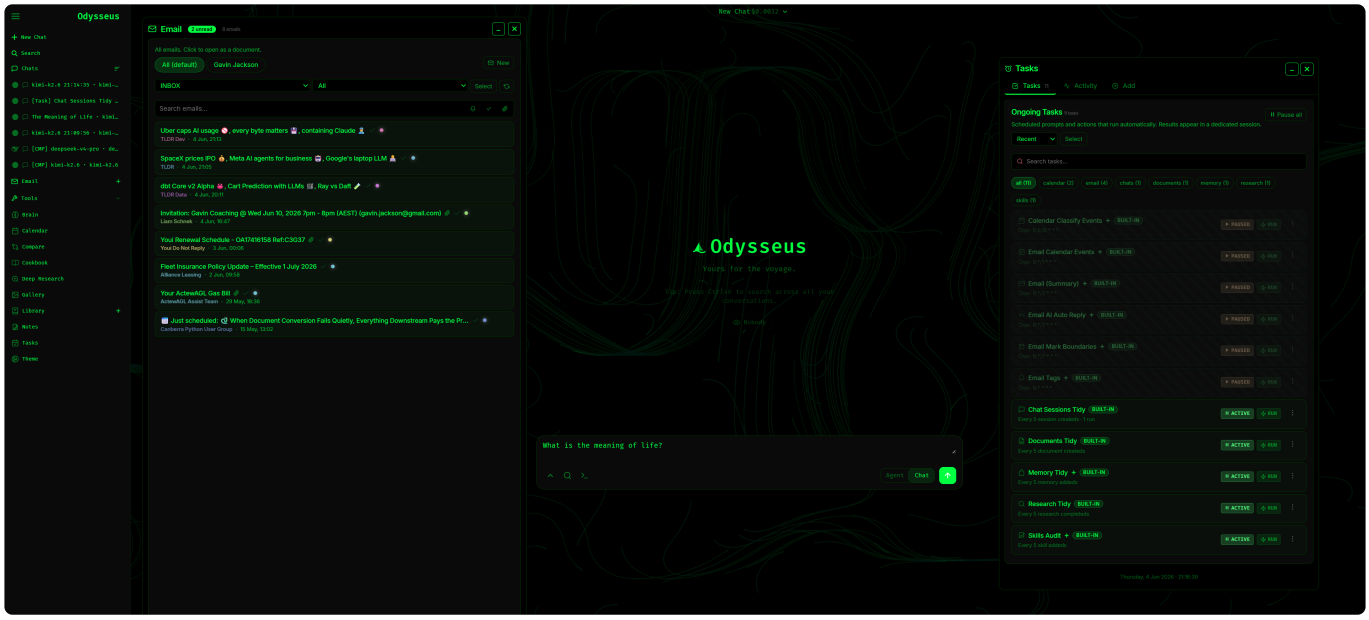
Odysseus Is Beautiful, But I Still Trust Hermes More

June 4, 2026 / Gavin Jackson

odysseus hermes ai agent self-hosted local-llm gmail open-source



I have spent the last few days using Odysseus, the self-hosted AI workspace that has very quickly become one of the more interesting projects in the open source agent ecosystem. I went in wanting to like it, partly because local-first AI needs projects with ambition, and partly because I am still looking for the right shape of personal agent for my own setup.

My reaction ended up mixed. There are parts of Odysseus I genuinely love, especially the web interface and the way it lowers the friction around local models. There are also parts that frustrated me quickly, particularly the email, calendar, and contacts experience. After trying Hermes the week before, Odysseus felt less like an assistant I could build a relationship with and more like a very polished AI control panel.

For some users, that trade-off will be perfectly fine. If your goal is to run models, compare outputs, manage local inference, and keep your AI work inside a self-hosted workspace, Odysseus has a lot going for it. If your goal is to build something that feels like a persistent personal assistant, Hermes still feels closer to the mark for me.

The Web Interface Is Excellent

The first thing Odysseus gets right is the browser experience. It is probably the best web interface I have seen in the AI agent space so far, and I do not mean "good for an open source project." It feels like someone actually thought about the experience of using local AI every day, rather than building a collection of disconnected tools with a chat box attached.

The UI has real coherence. Chat, agents, research, documents, notes, tasks, image tools, memory, email, calendar, and model management all sit inside one workspace. The design has personality without becoming obnoxious, and it avoids the common self-hosted trap where the user is expected to tolerate a rough interface because the underlying idea is interesting.

The contrast with Hermes is pretty clear here. Hermes is much more terminal-and-channel shaped, which suits my own assistant workflow because it mostly lives in Telegram and on a VPS. Odysseus gives local AI a proper workspace: panels, previews, documents, model workflows, comparison views, and settings that feel like they belong together.

If someone asked me which project makes the stronger visual first impression, I would say Odysseus. It looks and feels more like a complete application.

Local Models Are a Strength

The other part I really liked was how naturally Odysseus facilitates local-first models. The project is built around the idea that you can run models on your own hardware or point the app at whatever endpoints you already use. Its README lists support for local and API-backed options including Ollama, llama.cpp, vLLM, OpenRouter, and OpenAI-compatible endpoints.

In practice, this maps neatly onto how I use AI now. I do not have one model that solves everything. I have local models for privacy-sensitive or low-stakes work, cloud models for harder reasoning, and a constant curiosity about whether the latest Gemma, Qwen, DeepSeek, or Kimi model is good enough for a particular job.

The Cookbook feature is the kind of glue local AI needs. Running a model locally is not just a matter of choosing a name from a list. You need to know what fits in VRAM, which runtime makes sense, what quantization to choose, and whether the result will be fast enough to use without constantly sighing at the screen. Odysseus tries to make that process less opaque by scanning hardware, recommending models, and helping with download and serving workflows.

I tested with local-first models such as Gemma, and this was where Odysseus felt most aligned with what I want from open source AI tooling. Local models are not always the smartest or fastest option, but they change the privacy and control posture. For email, notes, documents, and memory, that trade-off is worth taking seriously.

Hermes can also use local or custom model endpoints, so this is not a case of Hermes being cloud-only. The difference is emphasis. Hermes feels like an always-on agent runtime that happens to support flexible models. Odysseus feels more like a local model workbench with agent features around it. When I am actively comparing models, managing local inference, or experimenting with what my hardware can handle, I would rather be in Odysseus.

Model Comparison Is Actually Useful

The Compare feature is one of the more practical ideas in Odysseus. You send the same prompt to multiple models and compare the outputs side by side. Odysseus also supports blind comparison, which is useful because it reduces the temptation to forgive a model just because you already like its name.

I would not treat this as a novelty feature. Model selection is now a real part of using AI well. A local Gemma variant might be fine for summarising notes but weak at tool use. DeepSeek might reason well in one situation and become brittle in another. Kimi might be excellent at long context but feel less natural as an assistant. A smaller model might be enough for classification, while a frontier model is still worth paying for when the task involves planning, coding, or judgement.

You only really learn this by testing models against your own prompts. Benchmarks are helpful, but they rarely tell you whether a model can handle your writing style, your inbox, your codebase, or your tolerance for waffle.

Odysseus gives that testing process a proper interface. Hermes has multi-model routing and can use different providers, but its strength is not side-by-side evaluation. Hermes is more operational. Odysseus is more experimental. If you are tuning a local AI setup and trying to decide which model deserves a place in your daily stack, Odysseus has the better shape for that job.

The Integrations Let It Down

The part that lost me was integration.

On paper, email and calendar support are exactly what I want. Odysseus advertises IMAP/SMTP email triage and CalDAV-aware calendar features, which is a sensible local-first approach. It avoids building the whole product around one vendor API, and it fits the self-hosted philosophy.

In practice, as a long-term Gmail user, I found it too primitive to be useful. The Gmail experience was minimal at best and broken at worst. It did not feel like a mature integration with the way I actually use email.

Gmail is not just an inbox. It is labels, search, threads, identity, contacts, calendar context, priority signals, spam handling, filters, and years of accumulated behaviour. A thin IMAP/SMTP layer can technically connect to it, but technical connectivity is not the same as a useful assistant.

The same problem showed up around calendar and contacts. The pieces were present in a basic sense, but I never felt like Odysseus understood my actual communication environment. It could touch the surface, but it did not feel like it could live inside the workflow.

I know that is a hard bar. Email and calendar integrations are notoriously difficult, especially if you are trying to stay open, local-first, and provider-neutral. Still, personal agents live or die on this kind of context. An agent that handles email poorly is worse than no agent at all, because it adds friction to one of the most sensitive parts of the day.

I need confidence that an email assistant understands threads, recipients, dates, context, and intent. I need drafts that feel like they belong in my inbox. I need it to know the difference between "summarise this" and "help me respond to this person I have known for a decade." Odysseus did not give me that confidence in its current form.

I am not writing it off forever. This is early software, and integrations like this take time. But based on my testing, the email, calendar, and contacts side made the product feel much less useful than the feature list suggested.

It Did Not Feel Personable

The more subtle problem was personality.

With Hermes, I have been able to shape an assistant that feels like an assistant. It has persistent memory, it lives in my Telegram workflow, and it gradually accumulates a model of who I am and how I like work to be done. I gave it a name. I talk to it from my phone. It nudges me in the channel where I already live.

It can sound sentimental, but in practice it changes how I use the system. If an AI agent is going to become part of my day, I need to trust it as a continuing presence rather than treat it as a model selector with a prompt box. I need some continuity between yesterday's instruction, today's task, and tomorrow's follow-up. I also need it to present itself as my agent, not as whichever model happens to be selected.

Odysseus never quite crossed that line for me. I tested it with DeepSeek and Kimi K2.5, and the experience often felt like talking directly to the underlying model. The models presented themselves as the model, not the agent. At times, they did not seem to understand what Odysseus was.

The distinction is subtle, but it shows up quickly when a response starts with something like:

> *I am Kimi, a language model...*

and a response that behaves like:

> *I am your Odysseus agent, and I can help with the workspace, tools, models, and tasks available here.*

The first breaks the wrapper. The second gives you a coherent assistant experience. This may be fixable with better system prompts, stronger identity handling, memory injection, or model-specific instruction tuning, but out of the box I found myself getting frustrated. The app around the model looked polished, while the agent inside the app felt under-defined.

For a local model workspace, that might be acceptable. For a personal agent, it gets in the way.

Channels Matter More Than People Think

Hermes still feels ahead here because it lives where I already communicate: Telegram, Discord, Slack, WhatsApp, Signal, email, and other gateway options. I can talk to my agent from my phone, send it a message while I am away from the desk, and receive scheduled briefings in the same place I receive normal messages.

A good web UI is useful, but channel presence changes the relationship. If I have to open a specific web interface to talk to an agent, I think of it as software I use when I remember to use it. If it can message me in Telegram or WhatsApp, it starts to feel like part of the day.

I do not mean that in a gimmicky way. An agent in a messaging channel can interrupt, remind, summarise, ask for approval, and receive quick instructions in the gaps between other work. A browser workspace is better for deliberate sessions: research, model testing, document editing, and configuration. It is less natural for an ambient assistant that should be available while I am doing other things.

Hermes clicked for me so quickly partly because it did not just have tools; it had reach. Odysseus has breadth inside the app, while Hermes has stronger presence outside the app. For a personal assistant, I care more about that presence than I expected to.

What Odysseus Does Better Than Hermes

None of this is meant as a dismissal. Odysseus does several things better than Hermes today.

The web interface is clearly better. Hermes has a strong CLI and gateway story, but Odysseus feels like a first-class browser workspace. The local model workflow is also more visible. Cookbook, model serving, hardware-aware recommendations, and the local inference focus make Odysseus feel built for people actively experimenting with models on their own machines.

The model comparison feature is another standout. Side-by-side and blind comparison are exactly what I want when evaluating outputs across DeepSeek, Kimi, Gemma, Qwen, and whatever else is currently interesting. Hermes can route between providers, but Odysseus makes evaluation feel like a normal part of the workflow.

Odysseus also offers a broader in-app bundle. Documents, notes, tasks, image tools, deep research, chat, agents, and model management all live in the same application. Hermes can do a lot, but it feels more like an agent runtime. Odysseus feels more like a local AI suite.

For someone who wants to build a local AI lab on their desktop, compare models, run experiments, and keep documents inside a self-hosted workspace, I would keep Odysseus high on the list. It is a serious project, not a toy.

It just is not quite what I am trying to build. I want an assistant.

My Honest Take

Odysseus is a beautiful local-first AI workspace with a lot of promise, but it is not yet the personal agent I want to live with.

The web interface is excellent. The support for local models is genuinely valuable. The model comparison workflow is one of its best ideas. I like the ambition, and I think the open source agent ecosystem is better for having projects like this appear.

My frustration is that the parts I care about most in a daily assistant still feel undercooked. The email, calendar, and contacts side was too primitive for my real Gmail-heavy life. The assistant identity was weak. The models too often behaved like standalone models rather than one coherent agent. Without mature messaging channels like WhatsApp or Telegram at the centre of the experience, Odysseus never gave me the same sense of continuity that Hermes did.

Hermes still feels like the stronger bet for my own setup because it behaves more like an assistant that can live with me over time. It has persistent memory, channel presence, scheduled tasks, and a clearer path toward becoming "Bob" rather than just another interface for models.

Odysseus feels like a very promising workbench, maybe even the best-looking local AI workbench I have tried. I can imagine using it alongside Hermes rather than instead of it: Odysseus for model management, comparison, and deliberate local AI sessions; Hermes for the always-on assistant layer.

The open source agent ecosystem is only just lifting off, and we are still working out whether the winning shape is a local workspace, a messaging-native assistant, an IDE agent, a cloud worker, a phone companion, or some combination of all of those. My guess is that the future is hybrid: Odysseus-like model management and comparison, Hermes-like memory and channel presence, local models for private context, cloud models for hard reasoning, and an agent identity that survives across all of it.

In that split, Odysseus is strongest as a local AI workspace. Hermes, for me, still feels stronger as the assistant layer.

References

- Odysseus official website
- Odysseus on GitHub
- Hermes Agent documentation
- Hermes Agent integrations
- My earlier post: Hermes Agent Made My AI Assistant Useful Again

Downloaded from https://www.gavinj.net/post/odysseus-review-local-first-ai-workspace

Generated November 14, 2026. Copyright Gavin Jackson. All rights reserved.