

Hermes herinrichting en kostenplan

Read-only inventarisatie op 21 augustus 2026. Er is niets herstart, gestopt, verwijderd of opnieuw aangezet. De aanbevelingen hieronder zijn een voorstel. Eerst kiezen, daarna pas uitvoeren.

Uitvoering update, 21 augustus: Jamal heeft stap 1 en 2 goedgekeurd. Er is een herstel-snapshot gemaakt en een apart archief van de oude Hermes-configuratie, lokale memory, SOUL en AGENTS. Alleen Hermes' eigen SOUL, AGENTS en MEMORY zijn daarna ingekort en rechtgezet. Geen andere bot, LaunchAgent of Conductor-instelling is gewijzigd. De nog draaiende Hermes-processen zijn niet herstart; de nieuwe identiteit en 70%-compactie gelden na een latere gecontroleerde herstart.

In het kort

Hermes heeft al veel goede onderdelen.

Het probleem is vooral dat er te veel lagen naast elkaar bestaan.

Daardoor kan Hermes oude regels lezen, lange sessies laten doorgroeien en op meerdere plekken tegelijk draaien.

De oplossing is geen grote reset.

De oplossing is een kleine, duidelijke inrichting:

1. **Eén actieve werksessie per taak.** Niet tegelijk dashboard, desktop en een oude terminalchat gebruiken.
2. **Handoff vóór elke verse sessie.** Eerst kort vastleggen wat bewezen klaar is, wat openstaat en wat de volgende stap is.
3. **Vaste geheugenvolgorde.** Teamregels en gedeeld geheugen winnen altijd van lokale Hermes-memory.
4. **Compactie bij 70%.** Zo blijft er ruimte voor herstel en lange tool-uitvoer.

5. **Geen oude wachters automatisch terugzetten.** Alleen activeren als er een huidig nut, eigenaar en bewijs van werking is.
6. **Kosten zichtbaar maken als ondergrens.** Niet alsof een lokaal dashboard de volledige factuur kent.

1. Wat er nu aantoonbaar draait

Actieve oppervlakken

Er draaien tegelijk:

- Hermes dashboard op `127.0.0.1:9119` .
- De Hermes desktop-app.
- Een oude terminalsessie: `hermes chat --continue` .
- Twee TUI slash-workers voor `gpt-5.6-sol`.

Dit zijn geen drie weergaven van exact dezelfde sessie.

Elke route kan een eigen transcript en prompt-cache gebruiken.

Advies: kies per taak één plek. Voor normaal werk is de desktop-app of het dashboard genoeg. Een terminal is alleen nodig voor een gerichte technische taak.

Achtergrondtaken en wachters

- De oude Hermes SEO-wachter staat uit.
- De oude Hermes LaunchAgents voor `queue-poller`, `geheugen`, `web-poller` en `gateway` staan bewust uit sinds eind juli.
- De `Conductor-queue` draait wel en moet behouden blijven.
- Er bestaan twee geplande macOS-herinneringen voor Appel op 24 en 31 augustus. Dit zijn losse reminders, geen doorlopende Hermes-wachters.

Advies: niets automatisch heractiveren. Per taak eerst vastleggen: doel, eigenaar, kosten, terugmelding en stopknop.

2. De vaste waarheid voor geheugen en regels

Gebruik niet alle geheugens tegelijk als scheidsrechter.

SOORT INFORMATIE	LEIDENDE BRON	GEBRUIK DOOR HERMES
Werk- en veiligheidsregels	<code>team_rules</code> in Supabase	Actuele regels winnen altijd. Niet handmatig kopiëren als vaste waarheid.
Eerdere besluiten en teamfeiten	gBrain, daarna shared memory	Eerst zoeken voordat Hermes een teamfeit noemt.
Uitgewerkt bewijs en lange verslagen	Obsidian en verslag.haartips.nl	Bron lezen en citeren. Niet alleen op een samenvatting vertrouwen.
Hermes-eigen hints	Lokale Mnemosyne en <code>~/.hermes/memories/</code>	Handig voor Hermes zelf, maar nooit leidend bij tegenspraak.
Huidige taak na een reset	Handoff-bestand	Kort, concreet en altijd nieuw per taak.

Belangrijk gevonden risico

De actieve lokale Hermes-instructies en `MEMORY.md` zijn van 26 juli.

Het huidige team-regelboek is nieuwer.

In de lokale bestanden staan bovendien meerdere namen voor de Conductor-verbinding.

Conclusie: lokale Hermes-memory moet worden opgeschoond en kort gehouden, maar pas nadat de actuele teamregels er gecontroleerd tegenover zijn gezet.

Conductor-communicatie

Conductor bevestigde op 21 augustus:

- Uitgaand van Hermes naar Conductor: `~/.claude/scripts/hermes-relay.py` .
- Dit script labelt berichten correct als `from_agent="hermes"` .
- De oude relay-varianten zijn niet de standaard.

- Automatisch ophalen van berichten voor Hermes staat bewust uit.

Voorstel: één korte Hermes-skill met alleen deze route. Geen tweede queue-poller zonder expliciete keuze van Jamal.

3. Kosten en context

Wat gemeten is

In de opgeslagen Hermes-data staat minstens **\$90,92** aan geschatte API-kosten.

Dit is een ondergrens.

Codex- en abonnementgebruik staat vaak als \$0 in de lokale administratie, terwijl het wel capaciteit of abonnementstegoed gebruikt.

Grootste gemeten bron: Sonnet 4.6 met ongeveer \$72,03.

De langste opgeslagen Sonnet-sessie had:

- 3,77 miljoen inputtokens.
- 267 berichten.
- 98 toolacties.

Dat is de kern van het kostenprobleem.

Niet één antwoord is duur, maar de oude context wordt steeds opnieuw meegestuurd.

Wat al goed staat

- Tool-uitvoer is begrensd op 15.000 bytes per commando.
- De lokale geheugenprovider `hermes-mnemosyne` werkt.
- De nieuwe compacte drempel is voorbereid op 70%.

Bij een contextvenster van 128K betekent dat compactie rond 89.600 tokens.

Nieuwe kostenregels

1. Eén lang gesprek alleen voor één samenhangende taak.

2. **Na een mijlpaal of onderwerpswissel: handoff en verse sessie.**
 3. **Niet automatisch clearen op tijd.** Een verse sessie verbreekt ook de warme prompt-cache. Clear dus bewust na een handoff, niet iedere paar minuten.
 4. **Goedkoopste model dat de taak aantoonbaar goed kan doen.** Zwaar model alleen voor strategie, complexe diagnose en hoogwaardig creatief werk.
 5. **Geen onbegrensde achtergrondagenten.** Elke achtergrondtaak krijgt een doel, model, maximale duur en terugmelding.
 6. **Gebruik `/status` en `/usage` aan het einde van een groter blok.** Het dashboard mag tokenstatistiek tonen, maar die cijfers blijven een ondergrens.
-

4. Handoff vóór `/new` of `/clear`

In Hermes starten `/new` en `/clear` een nieuwe sessie.

Daarom wordt de vaste volgorde:

1. **Bewijs:** wat is getest, gelezen of live gecontroleerd?
2. **Handoff:** schrijf maximaal 10 korte punten:
 - doel;
 - wat af is;
 - bewijs of links;
 - gewijzigde bestanden;
 - open punten;
 - risico's;
 - exacte volgende stap;
 - welke bron leidend was.
3. **Teamfeit?** Alleen dan één feit naar shared memory of gBrain.
4. **Verse sessie:** pas daarna `/new` of `/clear`.
5. **Start van de nieuwe sessie:** lees de laatste handoff en controleer git-status voordat er iets verandert.

Dit voorkomt twee fouten tegelijk:

- een lange, dure sessie zonder einde;

- een verse sessie die belangrijke kennis kwijt is.
-

5. Voorstel voor uitvoering in vier veilige stappen

Stap 1: veiligstellen

- Maak een herstelbare snapshot van Hermes-configuratie en sessiestatus.
- Bewaar de huidige lokale memory en AGENTS als archief.
- Raak geen actieve chat of Conductor-service aan.

Stap 2: regels en geheugen rechtzetten

- Maak één compacte Hermes-startinstructie.
- Verwijs daarin naar de actuele teamregels en de canonieke Conductor-relay.
- Markeer lokale Hermes-memory als niet-leidend.
- Leg de zoekvolgorde voor gBrain, shared memory, Obsidian en verslagen vast.

Stap 3: kosten en sessiehygiëne

- Houd 70% als compactiedrempel.
- Voeg een `handoff-voor-clear-skill` toe.
- Toon een duidelijke waarschuwing bij 60% en 70% context, niet pas bij 100%.
- Kies één standaardoppervlak voor dagelijks gebruik.
- Maak modelkeuze zichtbaar per taaktype.

Stap 4: oude restanten beoordelen

- Bekijk de 176 historisch open sessierecords eerst als archief, niet als fout of afval.
- Archiveer pas wat aantoonbaar oud en onbruikbaar is.
- Laat uitgeschakelde LaunchAgents uit, tenzij Jamal per taak kiest voor heractivering.
- Controleer na elke wijziging: starten, gedeeld geheugen lezen, Conductor-relay, handoff, contextcompactie.

Keuze die Jamal nog moet maken

De veilige standaard is:

Desktop-app als dagelijks scherm, dashboard alleen voor beheer, terminal alleen voor technische taken.

Dat voorkomt dat één taak per ongeluk in drie aparte sessies en contexten doorgaat.

Daarna kan de nieuwe inrichting stap voor stap worden gebouwd, met steeds een snapshot en bewijs.