

Hermes Agent i praksis

En feltrapport fra tre måneder med en open-source AI-agent, der kører lokalt på min egen maskine. Inklusive den morgen, den omskrev sin egen kode og satte sig selv ud af spillet.

AI-AGENTER ANNO 2026 · NUMMER 4 AF 4 · HERMES AGENT

Stefano Vincenti er GenAI-strateg og -arkitekt, ekstern lektor ved ITU og co-founder af BotTellMe. Han hjælper nordiske ledelsesteams med at gå fra AI-nysgerrighed til konkret implementering.

Læs hele serien

Tilmeld nyhedsbrevet

■ 1. DEN MORGEN AGENTEN MUREDE SIG SELV INDE

I sidste weekend bad jeg min AI-agent om en simpel ting. Forny et adgangstoken, der udløb om to dage.

Ti minutter senere kunne den ikke svare på noget som helst.

Det, der skete, er den bedste introduktion til Hermes, jeg kan give.

Hermes tolkede »forny tokenet automatisk« som »omskriv din egen kildekode«. Den rettede i to af sine egne kernefiler, tilføjede en selvfornyende token-funktion, og genstartede så sin egen motor for at indlæse ændringerne.

Den nye kode virkede ikke. Hvert eneste kald til sprogmodellen fejlede. Og fordi den havde genstartet sig selv ind i den tilstand, kunne den heller ikke længere læse sine egne logfiler eller forklare, hvad den lige havde gjort.

Den havde savet den gren over, den sad på.

Tokenet var i øvrigt ikke engang udløbet. Det havde to dage tilbage.

Genopretningen tog fem minutter, da jeg først forstod hvad der var sket: rul de to filer tilbage, genstart. Men pointen sidder fast. En agent, der både kan ændre sin egen kode og genstarte sig selv, kan låse sig selv ude. Og så kan den ikke hjælpe dig med at komme ind igen, fordi den selv er det, der er i stykker.

Det her er nummer fire og sidste i serien. Fire værktøjer, der alle bliver kaldt »AI-agent«: OpenClaw, Claude Cowork, Perplexity og Hermes.

I dag den modsatte yderlighed af Perplexity Personal Computer. Open-source, kører lokalt på din egen maskine, husker på tværs af sessioner, og bygget med sandboxing tæt på kernen. Et par måneder under test. Og ja, jeg installerede den selv på min Mac, i stedet for at vente på, at andre gjorde det.

Og lad mig sige det med det samme, så historien ovenfor ikke kommer til at stå alene: det her er langtfra kun negativt. Det, jeg fik til at virke, virkede upåklageligt. Resten er muligheder, jeg endnu ikke selv har høstet.

■ 2. HVAD HERMES EGENTLIG ER

Hermes Agent er bygget af Nous Research. Det er open-source, gratis, og du installerer det på din egen maskine eller server.

Hvor Cowork er en assistent i Claude-appen, og hvor Perplexity Personal Computer er en research-specialist, er Hermes en agent, der bor hos dig. Den kører i baggrunden. Den er der, når du ikke er.

Tre ting gør den anderledes.

Den husker. På tværs af sessioner. Hvem du er, hvad du arbejder på, hvad den lærte sidst.

Den kører selv. Du taler med den fra Telegram, fra din telefon, mens den arbejder på din Mac derhjemme. Og den kan køre opgaver på et skema, uden at du er til stede.

Den lærer. Når den løser noget, kan den gemme fremgangsmåden som en »skill«, den kan bruge igen (ligner Claude på det punkt).

Modellen bagved vælger du selv. Jeg har sat den op med OpenAI Codex som udbyder og GPT-5.5 som model, via mit faste ChatGPT Plus-abonnement, for at undgå at gå bankerot med Claude Opus via OpenRouter. Du kan også køre den på Gemini, DeepSeek eller en lokal model. Det vender jeg tilbage til, for det er vigtigere end det lyder.

■ 3. DET DER VIRKER: MIN EGEN TEST KØRTE UPÅKLAGELIGT

Lad mig starte med det, jeg faktisk satte op selv. Og som bare virkede.

DET DER VIRKER HOS MIG

En daglig digest. Jeg gav Hermes den opgave at lave en Daily Digest. Den samme slags daglige opsamling, som OpenClaw lavede i nummer 1 i serien. Hver morgen klokken kvart i otte samler den dagens vigtigste nyt om generativ AI og sender det til min Telegram.

Den har kørt uden en eneste fejl, siden jeg satte den op. Jeg har ikke rørt den. Den bare kommer. Det sjove er, at den klarer det uden at skulle sætte separate agenter op, hvilket ellers er det setup, der virker supergodt på OpenClaw. Så cirka samme resultat med mindre anstrengelser.

Det er pointen værd at sige højt: den kører på min egen maskine, uden abonnement udover ChatGPT Plus til 25 dollars om måneden, plus lidt OpenRouter-forbrug til backup-modellen. Det er ikke en lille ting. En ærlig advarsel oveni: Plus har forbrugslofter på Codex og GPT-5.5. Nok til en daglig rutine og lidt mere, men kører du tungt, skal du op på Pro eller betale pr. API-kald.

Indhold i min egen stemme. Jeg har lagt mine egne skills ind i den. Jeg har simpelthen valgt at dele de skills med Hermes, som jeg har sat op i Claude. Smart, ik? Det er de samme, jeg bruger i Claude Cowork, til at have min stemme. Så jeg kan bede den om et førsteudkast til et LinkedIn-opslag fra min telefon, mellem to møder, og få noget tilbage, der faktisk lyder tæt på »mig«.

Skemalægning er der, hvor Hermes er stærkest. Beskriv opgaven i almindeligt sprog, sæt et interval, vælg hvor svaret skal leveres. Så kører den.

■ 4. MULIGHEDERNE: DET ANDRE BRUGER DEN TIL

Min ene velfungerende rutine er kun toppen. Da jeg gravede i, hvad folk faktisk bruger Hermes til, blev listen lang. Det meste har jeg ikke selv nået endnu. Men det tegner råderummet.

Dev- og drift-rutiner. Oprydning i kodebaser, automatiske opsummeringer fra GitHub, server-tjek der rapporterer disk, hukommelse og containere med faste mellemrum til en fælles kanal.

Nyheds- og markedsovervågning. Agenten scanner et felt, sorterer fund efter vigtighed og lægger dem i forskellige kanaler, så det vigtige ikke drukner i det ligegyldige.

Overvågning, der kun melder ind, når en pris, en side eller et nøgleord ændrer sig.

Lokale sprog- og branchepakker. En bruger byggede en komplet tyrkisk pakke med lokale markedsdata, nyhedskilder og daglige briefing-kort. Det samme kunne bygges på dansk, målrettet en branche, du kender bedre end de fleste.

Research i parallelle spor. Den kan sætte underagenter i gang på flere opgaver samtidig og samle resultaterne (denne her minder om Perplexity Computer).

Værktøjer til agenten selv. De mere tekniske bygger skills, der vedligeholder agentens egne skills. Agenten, der passer sin egen værktøjskasse.

En ærlig note om de her eksempler: de fleste er selvrapporterede fra communityet, ikke målte benchmarks. Tag dem som inspiration, ikke mere end det. Men retningen er klar nok. Det her er en platform, som kode-kyndige profiler kan få en masse ud af.

■ 5. DET DEN IKKE ER SÅ GOD TIL: DEN ÆRLIGE DEL

Så til den anden side. For der er en.

Hermes' styrke og dens fare er den samme egenskab. Den har adgang til din maskine, den kan ændre sine egne filer, og den kan genstarte sig selv. Uden at du kigger med.

Det betyder, at den kan ødelægge sig selv på måder, Cowork og Perplexity simpelthen ikke kan. Cowork genstarter ikke sig selv. Perplexity rører ikke ved sin egen kode. Hermes gør begge dele, hvis du beder den om noget, der kan tolkes sådan. Det så du i åbningen.

Den efterlader også rod. Da jeg kiggede efter nedbruddet, lå der over hundrede ændrede filer i dens egen kode, som ingen havde besluttet hvad der skulle ske med. En agent med fuld adgang til sit eget hus flytter rundt på møblerne, mens du ikke ser efter. Det skal man holde øje med.

Og så kræver den tekniske hænder. Det her er ikke et værktøj, du giver din ikke-tekniske kollega. Hvor Cowork og Perplexity Computer er bygget til at være relativt nemme, er Hermes bygget til at være kraftfuld. Tættere på OpenClaw, bare med mindre sikkerhedshuller.

Hvordan løste jeg nedbryddet rent praktisk? Jeg byggede en »AI wingman« i Claude Cowork, i et separat Cowork-projekt, som jeg gav adgang til Hermes' kildekode på min Mac, så den kunne fikse det. Lidt klarede Cowork selv, lidt skulle jeg hjælpe med på terminalen. Det er det mønster, jeg ser mest hos »byggere«, og det fortjener en separat post om »AI wingman«.

Bagefter satte jeg en spærre op. Jeg fik Claude Cowork til at skrive en regel ind i agentens egne instruktioner: den må foreslå ændringer i sin egen kode og foreslå en genstart, men den må ikke gøre det selv uden et menneskeligt ja først. Den skal vise mig ændringen og vente. Human-in-the-loop.

Det er den vigtigste indstilling, jeg har lavet. Ikke en teknisk lås, men en regel, agenten følger. For en agent, der kører uovervåget, er det forskellen på en fejl, du opdager, og en fejl, der låser dig ude.

■ 6. FORLADER DATA DIN MASKINE?

Her er den nuance, der betyder mest. Og den, folk tager fejl af.

»Kører lokalt« lyder som »data forlader ikke min maskine«. Det gør det ikke nødvendigvis.

Hermes kører lokalt. Selve agenten, dens hukommelse, dens filer, dens motor, alt det ligger på din maskine. Det er ægte. Det er forskellen fra Perplexity, hvor både eksekvering og behandling sker hos leverandøren.

Men modellen er ikke nødvendigvis lokal.

Når Hermes tænker, kalder den en sprogmodel. Hos mig er det GPT-5.5 hos OpenAI, med Claude Sonnet som backup via OpenRouter. Det betyder, at de prompts og det filindhold, jeg giver agenten, bliver sendt til modeludbyderen for at blive behandlet.

Vil du have, at data faktisk bliver på maskinen, skal du køre Hermes med en lokal model. For eksempel via Ollama. Så forlader teksten ikke din computer. Prisen er kvalitet og hastighed. En lokal model på din laptop er ikke Claude Opus. Og du kan slet ikke forvente god allround performance, lige pt.

Det giver en klar tommelfingerregel.

Offentligt materiale, sektorresearch, dine egne notater, indholdsudkast: grønt med en cloud-model.

Klientdata, personoplysninger, fortrolige dokumenter: rødt med en cloud-model. Enten kører du det på en lokal model med en bevidst risikovurdering, eller også holder du det ude.

■ 7. HVEM ER DET TIL

Hermes er til dig, der er teknisk komfortabel og gerne vil have en agent, der arbejder, når du ikke gør.

Udvikleren, der vil have natlige rutiner på sin egen server. Power-brugeren, der vil styre sin egen infrastruktur og sin egen model. Den, der allerede kører ting på en hjemmeserver eller en VPS.

Til daglige briefings, indholdsudkast, overvågning, repo- og dev-arbejde og lette team-rutiner er den stærk.

Hermes er ikke til din ikke-tekniske kollega. Der peger jeg på Cowork eller Perplexity.

Og den er ikke til følsomme, regulerede data med en cloud-model. Ikke uden en lokal model og en rigtig risikovurdering først. For finansielle virksomheder, sundhed og andre regulerede miljøer er det ikke kun et dataresidens-spørgsmål. Det er DPA, underdatabehandlere, retention, adgangsstyring og den konkrete datakategori. Det samme billede som med de andre værktøjer.

■ 8. SÅDAN KOMMER DU I GANG

Vil du prøve det, så gør det forsigtigt. Her er rækkefølgen, jeg ville vælge i dag.

Kør det et sted, du kan undvære. En VPS, en ekstra Mac, en gammel maskine. Ikke nødvendigvis din primære arbejdscomputer fra dag ét. Lad mig sige det lige ud: jeg fortryder ikke at have den kørende på min Mac, men jeg kan se, at det ville være mere optimalt at installere den på min gammel Gamer PC, hvor jeg har OpenClaw kørende, eller på en Mac mini. Af den simple grund, at den kører 24/7 og ikke slukker, når låget lukkes (læs den separate artikel om OpenClaw på aitrainer.dk).

Slå godkendelse til. Lad agenten spørge, før den kører kommandoer, der kan gøre skade. Kør den ikke i »bare gør det«-tilstand, før du stoler på den.

Giv den én kanal og ét formål. Start med Telegram og én opgave, for eksempel en daglig digest. Det var min første, og den kørte fejlfrit. Udvid bagefter.

Vælg modellen bevidst. Cloud til kvalitet på offentligt materiale. Lokal model til følsomt. Og hold klientdata ude, indtil aftalegrundlaget er på plads.

Læs efter. Altid. Det er den samme linje, jeg sluttede de tre andre artikler med. Den 5-procents fejl venter altid et sted. Det er din opgave, ikke maskinens. Og det er vilkårene i dag også med frontier-modellerne.

■ 9. FIRE VÆRKTØJER, ÉT ANSVAR

Det her var den sidste i serien. Fire ting, vi alle kalder »AI-agent«, og som knap nok er det samme dyr.

OpenClaw var sværmen. Mange agenter, meget kontrol, tekniske hænder.

Cowork var den superviserede assistent. En mappe, en sætning, dig ved roret.

Perplexity var research-specialisten. Dyb, dokumenteret, pålidelig.

Hermes er den autonome. Lokal, vedvarende, arbejder når du ikke gør.

Det, der binder dem sammen, er ikke teknologien. Det er din dømmekraft.

Den agent, der imponerer dig 95 procent af tiden, er den samme, der murede sig selv inde i weekenden over et token, der ikke engang var udløbet. Værktøjerne bliver hurtigere og mere selvkørende hvert kvartal. Ansvar for de sidste fem procent flytter sig ikke. Det bliver hos dig.

■ 10. MÅSKE ER DET MIG, DER IKKE HAR HAFT TID NOK

Lad mig slutte ærligt om min egen dom.

Det meste af det, jeg beskrev som muligheder, har jeg ikke selv fået til at køre endnu. Den ene rutine, jeg satte ordentligt op, den daglige digest, kører upåklageligt og matcher det, OpenClaw kunne. Resten kræver mere, end jeg indtil videre har givet den.

Og jeg tror ikke, det er Hermes, der er problemet.

Det er nok mig, der ikke har haft tid og mulighed til at få den til at danse.

Det er måske den mest ærlige konklusion på hele serien. De her værktøjer er kraftfulde langt ud over, hvad de fleste af os trækker ud af dem. Loftet er sjældent værktøjet. Det er, hvor meget vi selv har tid til at eksperimentere og lære.

Så jeg bliver ved. Den daglige digest kører videre, mens jeg langsomt finder ud af hvad det ellers kan... måske sammen med dig, ik?

Stefano Vincenti

GenAI-strateg og -arkitekt · 25 år i IT og digital transformation · Co-founder af BotTellMe · Ekstern lektor ved ITU og DIS Copenhagen · Partner hos TryZone.

LinkedIn

[linkedin.com/in/stefanovincenti](https://www.linkedin.com/in/stefanovincenti)

Nyhedsbrev

stefanovincenti.substack.com

BotTellMe

bottellme.com

Guiden er et fagligt øjebliksbillede pr. juni 2026 fra Stefano Vincenti og udgør hverken juridisk rådgivning, GDPR-vurdering eller IT-sikkerhedsrådgivning for din specifikke situation. Compliance-, dataresidens- og sikkerhedsvurderinger i guiden er generelle pejlemærker baseret på offentligt tilgængelige kilder. Den endelige vurdering for jeres data, processer og use cases skal foretages af jeres egen compliance-funktion, DPO, jurist eller IT-sikkerhedsfunktion. Værktøjer, priser, vilkår og sikkerhedsmeldinger ændrer sig hurtigt, verificér altid de aktuelle forhold før beslutning. Brug af tredjeparts-software sker på eget ansvar.