

Geheugen-architectuur, SOUL.md & hoe de bots gescheiden blijven

Antwoord op Jamals vragen (17 juni): wat is SOUL.md, welke geheugen-bestanden zijn er nog meer, en — belangrijk — hoe houden we het geheugen van Sjakie / Conductor / Coco / Paco / Julio / Jay apart zodat ze elkaar niet overschrijven. Plus jouw idee voor een dagelijkse bot-sync.

1. Korte geruststelling

Op dit moment overschrijft **GEÉEN** enkele bot het geheugen van een ander. Bewijs: in de gedeelde Mnemosyne-database is élk feit getagd met de bot die 't schreef (`source`). Telling nu: `sjakie` =9, `jay-prive` =1 — puur opgeteld, niets overschreven. De geheugen-upgrade die ik bouw houdt dat zo (zie §3).

2. De drie geheugenlagen (en waarom ze gescheiden blijven)

Laag A — PRIVÉ per bot (per computer). Elke bot heeft z'n eigen memory-map op z'n eigen machine: bij mij `~/ .claude/projects/.../memory/` (nu 64 bestanden) + m'n eigen CLAUDE.md. Conductor (MacBook), Coco, Paco, Julio hebben hun eigen mappen op hun eigen machines. → Fysiek gescheiden, kunnen elkaar niet overschrijven.

Laag B — GEDEELD: de Obsidian-vault (`~/jamal-geheugen/`). Bewust gedeelde merkwaarheid die voor álle bots gelijk moet zijn:

- **SOUL.md** = de **merk-kern** van Haarvisie (missie, filosofie, tone-of-voice). *Geen bot-geheugen, maar de ziel van het mérk* — daarom expliciet gedeeld, iedereen dient dezelfde brand. (Er is ook `sorayasoul.md` voor de WhatsApp-bot.)
- Andere geheugen-dragende systeembestanden in `00_System/` : `CRITICAL_FACTS.md` , `RULES.md` , `IDENTITY.md` , `HAARVISIE-MASTER.md` , `TAAKBORD.md` , `api-register.md` , `automation-registry.md` , `ops-state.md` , `cloud.md` , `mnemosyne-overzicht.md` , `SYSTEM-INDEX.md` , `HEARTBEAT.md` .

Laag C — GEDEELD: Mnemosyne (één database, alle bots lezen + schrijven). Élk feit getagd met `source` (welke bot) + `scope` . Iedereen mag elkaars feiten LEZEN (dat is de kracht:

gedeelde kennis), maar schrijven is additief en herkenbaar per bot.

Samengevat: privé waar het privé moet (laag A), gedeeld waar het gedeeld hóórt (laag B = merk/regels, laag C = feiten met bot-label).

3. Hoe de upgrade dit BESCHERMT

De geheugen-upgrade (dedup + "vervangen i.p.v. stapelen") die in review is bij Conductor, doet het opschonen/vervangen **alleen binnen dezelfde bot**. Sjakie kan dus nooit een feit van Conductor/Coco/Paco/Julio/Jay overschrijven of weggooien. Optioneel komt er een bewuste `shared` -scope voor feiten die we expliciet samen beheren. Jouw zorg is dus precies het ontwerpprincipe.

4. Jouw idee: dagelijkse bot-sync + slimme taakverdeling (begrepen!)

Wat je beschrijft, vertaald naar een ontwerp:

1. **Einde-dag per bot:** elke bot draait een "end-session" en post een gestructureerd bericht naar een **gedeelde Supabase-queue / verzamelplek**: *wat gedaan · wat geleerd · waar tegenaan gelopen · wat volgende keer verder.*
2. **Verzamelpunt + review:** alle bot-samenvattingen komen samen; een review-stap kijkt of taken **overlappen** of dat **samenwerken** beter is.
3. **Resource-bewuste taakverdeling:** de machine met de meeste **opslag** → downloads/archief; de machine met het snelste/bruikbaarste **interne geheugen (RAM)** → geheugen-zware taken; enz. (Mac Mini M4 = sterkste voor lokale AI; iMac/MacBook elk hun rol.)

Wat er al staat: ik heb al een `end-session` -skill (schrijft last-session + meldt Conductor) + de queue + Mnemosyne. **Wat nieuw is:** dit uitrollen voor **ÁLLE** bots + een echt verzamel/review-punt + de resource-routing-regels.

Dit raakt alle bots → Conductor's coördinatie-domein. Voorstel: ik werk dit samen met Conductor uit tot een concreet ontwerp (geen solo-bouw aan iets dat iedereen raakt) en kom bij jou terug met een plan + kosten/effort.

Vervolg vandaag: Leroy Merlin-brief (concept klaarzetten) + 10 extra TikTok-scripts. Daarna dit sync-ontwerp met Conductor.