

Kennissysteem-masterclasses, grondig

Je vond de eerste analyse te oppervlakkig. Terecht.

Die keek alleen naar de 3 "kennissysteem"-delen (fundering, skills, chief of staff) en gaf een top 4.

Deze keer heb ik **alle masterclasses regel voor regel** doorgelezen, plus het echte demo-materiaal (de skills, de hooks en de prorail-oefenmap). Ik heb er ELKE techniek uitgehaald, ook de kleine.

Wat de eerste analyse miste (kort)

De eerste analyse liet **4 hele masterclasses links liggen**:

- **Codex** (masterclass 2): goedkeuring-instellingen, plan-modus, threads als parallelle taken, skills met \$ en plugins met @, computer-use-beperkingen.
- **Claude Cowork** (masterclass 5): de "chat wordt een project"-signalen, agent een eigen computer geven, connector eerst los testen, scheduled taak op **1 over het hele uur** zetten, Haiku voor achtergrondtaken, known-good-dataset als controle, de swipe/tinder-verificatie, overdrachtsbrief tussen gesprekken, Beeper-MCP, caveman-modus om tokens te sparen.
- **Lokale AI / privacy** (masterclass 6): LM Studio, Gemma, grounding, en de allerbeste truc: **een cloud-AI bouwt een tool die daarna 100% lokaal draait** (AVG-proof). Plus MLX voor Mac, gefingeerde namen.
- **Microsoft Copilot** (masterclass 7): de model-council (twee modellen die elkaar controleren), skill uit je eigen schrijfstijl, interview-mij-één-vraag-tegelijk, human-in-the-loop, gevoeligheid-label dat meegroeit.

Veel hiervan heb je al. Maar het waren **echte gaten in de eerste analyse**, want ze werden niet eens genoemd.

Deel A - De kennissysteem-fundering (de 3 kern-masterclasses)

Bron: de research-samenvatting + het echte demo-materiaal (skills/, hooks/, prorail-map).

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
A1	Map met losse markdown-bestanden, AI vist er just-in-time uit	Ja. MEMORY.md + Obsidian-vault + gBrain	Al goed
A2	Alles naar Markdown zetten (zuinig, modellen zijn erop getraind)	Ja. Hele vault is markdown	Al goed
A3	Obsidian als leesbare lens op de bestanden	Ja	Al goed
A4	Werkdocument als spin in het web (status boven, links eronder)	Ja. MEMORY.md is exact dit + HEARTBEAT.md live-status	Al goed
A5	Regel: mag "weet ik niet" zeggen	Half. Staat in oefenproject, niet vloot-breed	Ja, tekst klaar (regel 1)
A6	Regel: eerst letterlijk citeren uit je bestanden	Half. Zelfde	Ja, tekst klaar (regel 2)
A7	Skill = markdown met instructie, behandel als software (vertrouwde bronnen)	Ja, ruim. 100+ skills onder git	Al goed
A8	Skill-frontmatter: name + description bepalen wanneer hij triggert	Ja. Al je skills hebben dit	Al goed
A9	Kebab-case voor de skill-naam (anders triggert hij niet)	Ja	Al goed

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
A10	Progressive disclosure: bij start alleen descriptions, rest pas bij gebruik	Ja. Zo werkt Claude Code standaard	Al goed
A11	references/ -map: 2-3 goedgekeurde voorbeelden als toon-ijkpunt	Nee (was het grootste gat)	Al gebouwd, zie onder
A12	3 kern-skills: brief (gespreksprep), research, rapport	Deels. research + rapport ja, per-gesprek-brief nee	brief: later
A13	naar-markdown-helper (Word/PDF/URL omzetten)	Ja. firecrawl, pandoc, scrape-skill, curl	Al goed
A14	3 automatiseringslagen: skill (jij roept) / routine (op tijd) / hook (op gebeurtenis)	Ja, zwaarder. skills + LaunchAgents/cron + volle settings.json	Al goed
A15	Geheugen: sessie.md + hook die opslaat op Stop	Ja. auto-obsidian-stop + stop-checkpoint	Al goed
A16	Geheugen: hook die inlaadt op SessionStart	Ja. session-start-context	Al goed
A17	Em-dashes weghalen als AI-tell	Ja. humanizer-skill + de regel	Al goed
A18	Privacy-kader: niet alles mag naar een cloud-AI	Ja, deels. Zie deel D	Aanscherpen

De chieft of staff (deel 3)

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
A19	/chief : dagelijkse A4 met terugblik, taken per urgentie, open eindjes, te nemen besluiten	Ja, ruim. /start + daily-research-brief + conductor-briefing (2x/dag, zelfs als podcast)	Al goed

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
A20	Chief berekent "achterstallig / vandaag / wacht op anderen" uit velden, niet uit secties	Half. Onze taken zijn losser	Later
A21	taken.md met buckets (bezig/deze week/backlog/gereed) + per taak [due] [owner] [prio]	Half. Task Board + Supabase-taken, minder strak	Later
A22	Render-script maakt het A4 elke dag identiek (HTML)	Ja, andere vorm. Cockpit + verslag-site	Al goed
A23	Chief eindigt met een voorstel voor de volgende stap, voert niet zelf uit	Ja. "voorstellen, jij beslist" is al je werkregel	Al goed
A24	/chief transcript : gesprek automatisch in het dossier verwerken	Ja. Plaud-pijplijn + Granola-sync	Al goed
A25	vragenronde: 5+ keuzes als flashcards, wegtikken met cijfertoetsen (1-9, a =vrij, backtick=skip)	Nee. Jij kiest via Telegram, 1 voor 1. Script ligt klaar in de kennisbasis	Later (past minder bij afstand-werk)
A26	Breadcrumbs: agent schrijft na een inzicht 1 logboekregel per dag in het werkdoc	Half. milestone-save schrijft mijlpalen, geen inline dossier-logboek	Overwegen
A27	Kleine schoonmaak: hook kort Logboek automatisch in tot 10 regels (stil, goedkoop)	Nee	Al gebouwd, zie onder
A28	Grote schoonmaak: hook stelt maandelijks opschon voor en vraagt eerst (via vragenronde)	Nee. Je schoont met de hand op	Al gebouwd, zie onder
A29	blinde-vlekken: skill die zelf 1-3 nieuwe skills voorstelt op basis	Half. workflow-insights kijkt hoe je werkt, stelt geen	Klein gat, overwegen

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
	van herhaald handwerk	nieuwe skills voor	
A30	Principe: alles stelt voor, jij beslist; echte deadlines blijven heilig	Ja. Kernwerkregel	Al goed
A31	Take-home: maak 1 of 2 documenten heel belangrijk, verwijst daar steeds naar	Ja. MEMORY.md + CLAUDE.md doen dit	Al goed

Deel B - Skills en sub-agents (masterclass 4)

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
B1	Skill-map met SKILL.md (all caps, anders leest hij hem niet) + optioneel scripts/ , references/ , assets/	Ja	Al goed
B2	Skill maken met AI (skill-creator), stelt vragen met multiple-choice	Ja. skill-creator plugin actief	Al goed
B3	Skill "trainen" op smaak: toon 3 voorbeelden i.p.v. beschrijven	Deels. Nu via references-map voor verslagen	Al beter
B4	Skill als antipatroon: eerst de default-smaak (AI-slop, paars/roze) neutraliseren	Nee, niet expliciet	Overwegen bij design-skills
B5	Aan het eind van een goede run: "zullen we dit omzetten naar een skill?"	Half. Doe je soms handmatig	Overwegen als vaste reflex
B6	Skills kunnen skills aanroepen (skill-teams)	Ja. Skills verwijzen naar elkaar	Al goed

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
B7	Geen secrets in skills, nooit (het zijn platte tekstbestanden, exfiltreerbaar)	Ja. Secrets-regel + Keychain	Al goed
B8	Customer-facing skills = beschouw als openbaar (prompts zijn te achterhalen)	Ja, deels. Bots-binnen-het-huis regel	Al goed
B9	Sub-agent = eigen contextvenster, eigen prompt- en toolrechten	Ja. Je gebruikt subagents volop	Al goed
B10	Sub-agents tegen "context window infighting": deel losse onderwerpen op	Ja	Al goed
B11	Orchestrator-patroon: hoofdagente spant research/data/writer-agents met dependencies	Ja. Conductor + agent-teams	Al goed
B12	Model per sub-agent kiezen: middenwerk = Haiku (kosten + snelheid)	Ja. Model-routeringsregel (bots = Sonnet)	Al goed
B13	agents zijn rollen, skills zijn competenties (mentaal model)	Ja, impliciet	Al goed
B14	CLAUDE.md in de root, wordt CONSTANT gelezen, per project een eigen	Ja. Je hebt project- + globale CLAUDE.md	Al goed
B15	/init laat Claude zelf de CLAUDE.md schrijven en de mapstructuur documenteren	Ja. /init beschikbaar	Al goed
B16	AGENTS.md erbij zodat andere modellen het ook oppakken (model-agnostisch)	Nee	Klein, overwegen
B17	MCP = connector; tokens horen in de connector, niet in de skill	Ja. MCP-stack	Al goed

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
B18	Plugin = CLAUDE.md + MCP + sub-agents + skills gebundeld in 1 zip	Ja. Je draait al plugins	Al goed
B19	BoltAI / OpenCode: 1 abstractielaag hoger om van model te wisselen	Deels. Cursor, MoA, live model-switch	Al goed

Deel C - Claude Cowork (masterclass 5)

Dit is de masterclass die de eerste analyse volledig oversloeg (het bleek "de kennissysteem-5" maar is de Cowork-masterclass).

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
C1	Model-harnas: een model (Opus) + een pakje eromheen dat je computer in kan	Ja, concept. Claude Code zelf	Al goed
C2	Signalen dat een chat een project moet worden: draad kwijt, 3+ bestanden, veel connectors, chat hergebruiken	Deels. Jij werkt altijd in Claude Code (al "project")	Inzicht
C3	Geef de agent een eigen computer (tweedehands Mac mini / Intel NUC in de meterkast)	Ja. Sjakie op Mac Mini, bots op eigen machines	Al goed
C4	Nieuwe connector eerst LOS testen in de chat voordat je hem in een pijplijn zet	Half. Doe je soms	Ja, goede reflex
C5	Denk niet vóór de AI: zeg gewoon "zet mijn bonnen erin", laat de structuur los	Ja. "doe alles zelf", natuurlijke opdrachten	Al goed
C6	Scheduled taak op 1 over het hele uur zetten (spreiding, anders lopen alle taken tegelijk vast)	Nee, niet bewust	Ja, simpele tweak voor je cron/LaunchAgents

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
C7	"Run now" om een scheduled taak meteen te testen	Ja, doe je	Al goed
C8	Preferred model op Haiku voor achtergrondtaken (bijna alles kan, sneller + goedkoper)	Ja. Ollama/Haiku voor cron	Al goed
C9	Known-good dataset ernaast leggen om een pijplijn te controleren (99,9% goed = ik durf het aan)	Half. verify-skill, nooit-opleveren-zonder-bewijs	Ja, aanscherpen
C10	Niet-deterministische modellen zijn slecht met getallen: laat externe OCR (Mistral) het cijferwerk doen	Deels. Gemini-vision	Overwegen bij bon/cijferwerk
C11	Swipe/tinder-verificatie-UI: links de output, rechts de bron, wegtikken wat niet klopt	Nee	Leuk idee (koppelt aan vragenronde)
C12	Overdrachtsbrief: laat de chat een prompt schrijven die je aan de volgende agent geeft (distillation)	Ja. end-session handoff + ga-verder-in-interactive	Al goed
C13	Hele gesprek naar 1 markdown-bestand als eerste projectbestand	Ja. Handoff-doc	Al goed
C14	Beeper-MCP tussen je messaging (iMessage/WhatsApp/Telegram) en de AI	Ja. Beeper-MCP actief	Al goed
C15	Scheduler die dagelijks je berichten scant: "heb ik ergens een vraag gemist?"	Deels. WhatsApp-monitor, inbox-automatisering	Overwegen
C16	Caveman-modus: vraag super korte antwoorden om output-tokens te sparen	Nee	Kleine tip

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
C17	Index/router over je markdown-bestanden zodat hij niet elke keer alles opnieuw leest	Ja. obsidian-index-sync + memory-recall	Al goed
C18	Dispatch: vanaf je telefoon de agent op je computer aansturen (walkie-talkie)	Ja. Telegram-bridge + remote control	Al goed
C19	Computer use als LAATSTE redmiddel, pas als connectors falen	Ja. 2-pogingen-regel, dan API/CLI	Al goed
C20	Let op de sandbox: Finder-toegang = veel meer dan je denkt; per Cloud-onderdeel andere rechten	Ja, bewust	Al goed
C21	Geen bestellingen/betalingen out-of-the-box (bewuste rem tegen misbruik)	N.v.t.	Weten

Deel D - Lokale AI zonder dat je data je computer verlaat (masterclass 6)

Dit is waar je een **echt strategisch gat** hebt voor AVG-gevoelig werk (klantdata Coco/Soraya, Dani's schoolwerk).

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
D1	Lokale runtime-apps: LM Studio (hoofdtool), Ollama, Jan, Osaurus (Mac), Locally AI (iOS)	Deels. Ollama draait al (cron, backfill). LM Studio nog niet	Overwegen
D2	Lokale modellen: Gemma (schrijven), GPT-OSS (tool-calling), Qwen, Mistral (EU)	Deels via Ollama	Overwegen

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
D3	Quant-keuze: 4-bit is "het mooie gemiddelde"	Weten	Naslag
D4	Het gaat om VRAM/RAM , niet CPU: check hoeveel geheugen een machine heeft	Weten	Naslag
D5	Grounding: geef het kleine model kaders (web via DuckDuckGo-MCP, of je eigen database), anders bluft het	Ja, concept. Web search + gBrain	Al goed
D6	De kern-truc: gebruik een cloud-AI (Claude Code/Codex) om een tool te bouwen die daarna 100% lokaal draait via het LM Studio-endpoint	Nee	Ja, sterkste idee voor AVG-gevoelig werk
D7	Voorbeeld gebouwde lokale tools: tekst-redigeer-menulet, OCR-app voor scans, dagboek-app, allemaal offline	Nee	Idee voor klantdata/Dani
D8	Op Mac zoek je MLX -modellen (veel sneller op Apple-hardware); elders GGUF	Nee	Ja, relevant (je zit op Mac)
D9	Modellen laten samenwerken: GPT-OSS zoekt de naald, Gemma schrijft de analyse	Ja, concept. MoA, subagents	Al goed
D10	Gefingeerde/gepseudonimiseerde namen bij gevoelige data naar een cloud-AI	Half. Privacy-regels, secrets	Ja, als expliciete regel
D11	7 redenen voor lokaal: privacy, data-honger cloud, kosten, energie, weerbaarheid, compliance, autonomie	Weten	Naslag
D12	Speech-to-text is lokaal opgelost (faster-whisper); samenvatten grotendeels ook	Ja. faster-whisper draait al	Al goed

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
D13	Wat NIET goed lokaal kan: feitenvragen (hallucineert), naald-in-hooiberg over lange docs, zelf software schrijven	Weten	Naslag
D14	Compliance-opties org: contract met provider, regionaal datacenter, on-premise, dark fiber	N.v.t. (te groot)	Naslag
D15	Hardware voor serieus lokaal: richting 128GB VRAM, 3.500-6.000 euro; niet de 10-15k upsell	Weten	Naslag

Deel E - Microsoft Copilot (masterclass 7)

Copilot-specifiek, maar met een paar sterke overdraagbare ideeën.

#	TECHNIEK (OVERDRAAGBAAR)	HEBBEN WE DIT AL?	DE MOEITE WAARD?
E1	Model Council: 2 modellen (bijv. GPT + Claude) doen apart onderzoek en bekritisieren elkaars rapport in een synthese	Ja. MoA-tool + deep-research adversariële verificatie	Al goed
E2	Kennis-graaf/index van al je mail, agenda, documenten (Work IQ)	Ja. gBrain + Mnemosyne + Obsidian	Al goed
E3	Skill uit je eigen schrijfstijl: "maak een skill uit de toon van mails die IK schreef"	Deels. references-map doet dit voor verslagen; voor e-mail nog niet	Overwegen (mail-toon-skill)
E4	Interview-mij, stel de vragen 1 voor 1 (tegen het lege-document-probleem)	Ja. grill-me-skill	Al goed
E5	Voor-ingevulde prompts / snelknoppen voor terugkerende taken	Ja. Skills + slash-commands	Al goed

#	TECHNIEK (OVERDRAAGBAAR)	HEBBEN WE DIT AL?	DE MOEITE WAARD?
E6	Model bewust per taak kiezen (automatisch is soms dom)	Ja. Model-routeringsregel	Al goed
E7	Achtergrond-agent-paradigma i.p.v. reactieve chat (grote vraag, kom later terug)	Ja. Background agents, scheduled tasks	Al goed
E8	Scheduled flows met triggers (op tijd of op binnenkomend bericht)	Ja. LaunchAgents, pollers, hooks	Al goed
E9	Human-in-the-loop: eerst een ja/nee-mailtje voor de flow verder gaat	Ja. Goedkeuring-regels, deploy-lock	Al goed
E10	Sensitivity-label overerving: afgeleid document krijgt het hoogste geheim-niveau van zijn bronnen	Nee, niet expliciet	Overwegen voor secrets/data
E11	Model-agnostisch werken + moat-denken (niet afhankelijk van 1 model/framework)	Ja. harness-scan, moat-strategie	Al goed
E12	Agent genereert on-the-fly een mini-UI (formuliertje) dat je invult	Deels. Dashboards, playground	Weten
E13	Agent-output als input voor de volgende agent (research → notebook → podcast)	Ja. Podcast-pijplijn, agent-teams	Al goed

Deel F - Codex (masterclass 2)

Codex is OpenAI's tegenhanger van Claude Code. Je hebt Claude Code als hoofd, plus de codex-CLI. De meeste concepten heb je dus al, maar goed om te weten.

#	TECHNIEK	HEBBEN WE DIT AL?	DE MOEITE WAARD?
F1	Codex = geen model maar een harnas (GPT) dat in een map op je computer draait	Ja, concept. reference_codex_cli + codex-skill	Al goed
F2	Threads = parallelle taken	Ja. Background agents/subagents	Al goed
F3	Goedkeuring veilig instellen: begin met alleen leesrecht, schaal op bij vertrouwen	Deels. Jij draait bypassPermissions; per-tool-hooks vangen het gevaarlijke af	Bewust houden
F4	Plan-modus: laat eerst een plan maken voor hij uitvoert	Ja. plan-skill + ExitPlanMode	Al goed
F5	Skills roep je aan met \$, plugins met @	Ja, concept (andere syntax in Claude)	Al goed
F6	Computer use is VS-only, computer moet aanstaan, non-deterministisch (geen garanties)	Weten	Naslag
F7	Korte prompts werken het best	Ja. Actie-eerst, minder uitleg	Al goed
F8	Gooi geen gevoelige data in Codex	Ja. Secrets-regels	Al goed
F9	Doorwerken via mobiel	Ja. Telegram	Al goed

Wat ik nu heb GEBOUWD (bouwen mocht)

1. De references/-stijlmap (het grootste gat, nu dicht)

- Nieuwe skill: ~/.claude/skills/rapport/ met een references/ -map.
- Daarin 3 goedgekeurde voorbeelden als toon-ijkpunt: het homepage-redesign-advies, het SEO-diepteraapport en een korte gap-analyse.

- Plus een `README.md` dat je toon uitlegt: dyslexie-vriendelijk, korte zinnen, bulletjes, vet voor nadruk, geen em-dashes, oordeel eerst, "tarieven" niet "prijzen", eerlijk over onzekerheid.
- Vanaf nu schrijft de rapport-skill automatisch in jouw stijl, in plaats van dat je 'm elke keer opnieuw uitlegt.
- Append-only: keur je een nieuw verslag goed, dan groeit de map mee (golden master).

2. Zelf-opschonend geheugen (het gat met het hoogste nut)

- Nieuw script: `~/ .claude/scripts/memory-opschoon.py` .
- Twee snelheden, net als de masterclass:
- **Klein** (`--archiveer`): verplaatst oude, gedateerde sessie-notities naar `_archief/` . Nooit verwijderen, altijd omkeerbaar.
- **Groot** (`--voorstel`): als `MEMORY.md` over de 200 regels gaat, schrijft het een krimp-voorstel en **vraagt eerst**. Het past `MEMORY.md` nooit zelf aan.
- Getest (droog): `MEMORY.md` is nu 70 regels (slank), en het vond veilig 4 oude checkpoints om te archiveren.
- Belangrijk: de cron/LaunchAgent staat NIET aan. De activering laat ik aan Conductor, zoals afgesproken.

3. De 2 vaste regels (klaar voor Conductor)

- Kant-en-klare tekst in `~/ .claude/kennissysteem-2-regels-voor-conductor.md` .
- Regel 1: **mag "weet ik niet" zeggen** (nooit een feit/cijfer/link verzinnen).
- Regel 2: **eerst letterlijk citeren uit de bron** voordat je antwoordt.
- Ik heb de globale `CLAUDE.md` NIET met de hand aangepast (die wordt uit Supabase `team_rules` gesynct). Conductor voegt ze via de tabel toe, dan syncen ze naar alle bots.

Samenvatting in 1 blik

- **Al in huis (vaak zwaarder)**: de markdown-map, het werkdocument, skills-als-software, progressive disclosure, de 3 automatiseringslagen, geheugen-hooks op Stop en Start, de dagelijkse chieft-A4, transcript-verwerking, em-dash-opschoning, naar-markdown, sub-

agents met Haiku-routing, orchestrator-patroon, model-council, kennis-graaf, grill-me, Beeper, Dispatch, distillation-handoff.

- **Nu gebouwd:** de references/-stijlmap (1), het zelf-opschonende geheugen (2), en de tekst voor de 2 regels (3).
- **Nieuw gevonden en echt de moeite (nog niet gebouwd):**
- De cloud-bouwt-tool-lokaal-draait-truc (D6) voor AVG-gevoelig klant/Dani-werk. Sterkste nieuwe idee.
- Gefingeerde namen als expliciete regel bij gevoelige cloud-prompts (D10).
- Scheduled taken op 1-over-het-uur spreiden (C6). Simpele tweak.
- Known-good-dataset als vaste controle bij cijferwerk (C9).
- MLX-modellen op je Mac (D8).
- **Overwegen (klein):** blinde-vlekken-skill (A29), breadcrumbs-logboek (A26), "omzetten naar skill?"-reflex (B5), mail-toon-skill (E3), sensitivity-overerving (E10), AGENTS.md (B16).
- **Later / past minder:** vragenronde-flashcards (A25, jij werkt veel op afstand via Telegram), swipe-verificatie (C11).

Niets van "nieuw gevonden" is gebouwd. Dat is advies. Zeg **go** op wat je wilt, dan zet ik het klaar.