Opruimronde: van 68 wachters naar 54

Jamal, 3 augustus: *"door alle regels en hooks en watchers die wij bouwen creëren we alleen maar meer manieren om dingen om te laten vallen — ons systeem is veel te fragiel."*

Hij heeft gelijk. Dit verslag laat zien hoe erg het was, wat eruit ging, en welke regel we vanaf nu aanhouden zodat het niet terugkomt.

Hoe erg het was

Op de MacBook draaiden **68 wachters**, 24 hooks en 293 scripts. Op één dag gingen er drie van onze eigen bewakers stuk:

- de session-limit-wachter sloeg elk uur vals alarm (hij las zijn eigen berichten terug)
- de vastloper-wacht bewaakte alleen klussen, niet de hoofdsessie — terwijl we dachten van wel
- het wachtrij-hulpje at een opdracht van Jamal op en meldde hem als "verwerkt"

Elk van die drie was ooit gebouwd om iets te beschermen. Alle drie werden ze zelf het probleem.

Wat eruit ging (12 stuks, allemaal terug te zetten)

Klaar met hun werk

WACHTER	WAAROM WEG
herinnering-noodstop	De noodstop is vandaag gebouwd en getest. De herinnering heeft zijn taak gedaan.
herinnering-hotjar	Hotjar draait sinds 26 juli.

Overlap: iets anders doet het beter

WACHTER	WORDT GEDEKT DOOR
telegram-poller-watchdog (draaide elke minuut, log 23 dagen leeg)	telegram-lezer-wacht, die verbindingen meet in plaats van processen
sjakie-session-warn	Sjakies eigen session-limit-watcher, vandaag gerepareerd
transcript-watchdog	bot-watchdog + de telegram-lezer-wacht

Weken niets gedaan, wel elke paar minuten gedraaid

WACHTER	INTERVAL	LOG STIL SINDS
dns-queue	5 min	24 dagen
heartbeat-sync	10 min	25 dagen
plaud-digest	20 min	30 dagen
appel-verslag-poller	5 min	18 dagen
usage-hourly-watch	1 uur	26 dagen
aireport-analyse	30 min	16 dagen
actiepunten-3daags-check	3 dagen	23 dagen

Bij elkaar draaiden deze twaalf ruim **50.000 keer per maand** zonder aantoonbaar resultaat.

Terugzetten kan altijd: alle plists staan in `~/.claude/launchagents-uitgezet-20260803/`.
Terug is één commando: `cp <plist> ~/Library/LaunchAgents/ && launchctl bootstrap gui/501 <plist>`.

Gecontroleerd na afloop: de wachtrij-worker, telegram-lezer-wacht, noodstop-wachter, geheugen-deler, hoofdsessie-wacht, besluitenlijst en Kimi-bot draaien allemaal nog.

Ronde 2: de aanname klopte niet — en dat is goed nieuws

Ik had gemeld dat er "zeven bewakers zijn die allemaal hetzelfde doen" en stelde voor ze samen te voegen tot één. Bij het lezen van de code bleek die claim **te grof**. Ze doen grotendeels verschillende dingen:

WACHTER	WAT HIJ ECHT DOET	OORDEEL
bot-watchdog	bots met een dode hartslag herstarten	blijft — dit is de kern
fleet-heartbeat	schrijft de hartslagen weg	blijft — dit is de databron, geen bewaker
stille-storing	vindt automatiseringen die stil kapot zijn	blijft — unieke functie
uptime-monitor	websites down	blijft — unieke functie
dubbelwerk-wachter	taken die elke nacht hetzelfde werk overdoen	blijft — unieke functie
vastloper	hangende klussen	**UIT** — logboek 4 dagen leeg, bewaakte alleen `claude -p` -klussen; bot-watchdog en de nieuwe hoofdsessie-wacht dekken dit
mcp-monitor	MCP-servers elk uur	**UIT** — draaide 24x per dag, meldde in zijn hele bestaan **0** problemen en deed **0** reparaties; de SessionStart-hook toont hetzelfde bij elke sessiestart

Belangrijker dan de twee die eruit gingen: er is niets nieuws gebouwd. Samenvoegen zou een nieuw draaiend onderdeel hebben opgeleverd dat zelf kan breken. Wegnemen wat al gedekt
is, is altijd beter dan samenvoegen tot iets nieuws.

De rapportage-cluster (8 stuks die Jamal losse berichtjes sturen) staat nog open. Dat is wél een echte bundel-klus, maar die vraagt bouwen — dus alleen met Jamals expliciete ja.

De regel die dit voortaan voorkomt

Een nieuwe wachter mag er alleen komen als hij drie dingen heeft:

1. Een **zelftest** die bewijst dat hij afgaat wanneer het moet én zwijgt wanneer het niet moet. Ook de vraag: *kan mijn eigen melding, of gepraat over mij, mij triggeren?* (die fout maakte de session-limit-wachter).
2. Een **vervaldatum of doel**: wanneer is deze wachter klaar met zijn werk? Herinneringen ruimen zichzelf op.
3. Het antwoord op: **welke bestaande wachter kan dit erbij doen?** Liever een check toevoegen
   aan iets dat al draait dan een nieuw draaiend onderdeel.

En het belangrijkste onderscheid, want niet alle bouwsels zijn gelijk:

> Een **wachter** kijkt of iets misgaat — dat is een extra onderdeel dat zelf kan breken.
> Een **oorzaak wegnemen** maakt het systeem kleiner.

Voorbeeld van vandaag: Sjakies zeven overtollige Telegram-lezers zijn niet bewaakt, ze zijn weggehaald. Dat onderdeel kan nu niet meer stuk. Zo hoort het.