

44 wachters over de vloot, en een meetfout van mijzelf

Jamal, 14-8-2026: "We zijn alleen maar bezig met het bouwen van wachters die moeten oppassen dat de wachter en de wachter moeten weer opletten op die wachter. Waarom niet een wachter die alles controleert? Waarom hebben we dat dan nog niet gedaan?"

Dit is het antwoord, met de meting erbij. Geteld op 14-8-2026 op vijf machines: de MacBook,

Sjakie, de iMac, Paco en de VPS. Coco, Julio en Lenovo konden niet gemeten worden.

In het kort

1. **Je hebt gelijk over de aantallen.** Er staan 44 wachters over de vloot, 20 daarvan op de MacBook. Dat is het echte probleem en dat staat als een huis.
2. **Maar mijn eerste alarm was vals.** Ik meldde dat zes wachters stilstonden. Na hermeting: vijf daarvan draaien gewoon. Ze schrijven alleen niets in hun logbestand als er niets aan de hand is. Ik had ouderdom van een logbestand aangezien voor stilstand. Zie hoofdstuk 2.
3. **Ik ben onderdeel van het probleem.** Ik bouwde er gisteren één bij en stond vandaag op het punt er nog één toe te voegen. Nummer 21 en 22, zonder ooit te kijken of er al iets bestond dat hetzelfde deed.
4. **Waarom het nooit één werd:** elke storing kreeg zijn eigen oplossing. Er is nooit een moment geweest waarop iemand zei "stop, dit is twintig keer hetzelfde".
5. **Het kan wel één worden, en het wordt er simpeler van.** Niet alle twintig zijn hetzelfde soort ding. Er zitten drie groepen in, en die vragen elk een andere behandeling.

6. **Van 20 naar 3.** Dat is het voorstel. Geen nieuwe laag erbovenop, maar samenvoegen en opruimen.

0. De hele vloot, niet alleen de MacBook

Jamal vroeg dit meteen terecht: dit speelt bij alle bots. Dus is het overal geteld.

MACHINE	ACTIEVE TAKEN	WACHTERS
MacBook (Conductor)	68	20
Sjakie (Mac Mini)	32	8
iMac	35	7
Paco (Windows, Spanje)	26	5
VPS	18	4
Gemeten totaal	179	44

Niet gemeten: **Coco** (geen SSH-toegang met de sleutels die ik heb, vier gebruikersnamen geprobeerd), **Julio** en **Lenovo** (allebei offline). Het echte aantal ligt dus hoger dan 44.

En dan de dubbelingen

WACHTER	STAAT OP	OPMERKING
noodstop-wachter	MacBook, Sjakie, Paco	drie keer dezelfde
transcript-watchdog	MacBook, Sjakie	twee keer
mcp-monitor	Sjakie, iMac	twee keer
Suno-wachter	Paco	twee stuks op één machine: PacoSunoWatchdog én PacoSunoWatcher

Die laatste is het duidelijkste voorbeeld van het patroon. Er stond al een wachter op Suno, er kwam een probleem, en er kwam een tweede wachter bij in plaats van dat de eerste werd aangepast.

1. Wat er op de MacBook precies staat

20 wachters, samen ongeveer 3.700 regels code, 19 losse scripts en 20 losse instelbestanden.

De MacBook is het diepst uitgezocht; de andere machines volgen hetzelfde patroon.

Groep A: meten en melden (9 stuks, 1.436 regels)

Deze doen allemaal exact hetzelfde: meet iets, en stuur een bericht als het fout is.

WACHTER	REGELS	RITME	MEET
credential-health	413	dagelijks 8u	API-sleutels
dubbelwerk-wachter	270	dagelijks 3u	taken die dubbel werk doen
marketing-output-wachter	234	dagelijks 9u	of er gepubliceerd is
mac-nightcheck	139	dagelijks 0u	gezondheid van de drie Macs
transcript-size-watchdog	106	elke 5 min	sessiegrootte
uptime-monitor	83	elke 30 min	sites en diensten
gbrain-stilte-bewaker	78	dagelijks 9u	of gBrain nog leeft
usage-hourly-watch	58	elk uur	verbruik
cost-weekly-alert	55	wekelijks	kosten

Deze negen kunnen één worden. Ze verschillen alleen in wát ze meten. Dat is een regel op een lijst, geen eigen bouwsel.

Groep B: ingrijpen en herstellen (5 stuks, 1.251 regels)

Deze meten niet alleen, ze doen ook iets: een proces herstarten.

WACHTER	REGELS	RITME	DOET
telegram-poller-watchdog	369	elke minuut	herstart de Telegram-poller
noodstop-wachter	238	elke minuut	voert de vloot-noodstop uit
bot-watchdog	237	elke 10 min	herstart bots
telegram-lezer-wacht	237	elke 15 min	bewaakt precies één lezer per bot
hoofdsessie-wacht	170	elke 2 min	herstart de hoofdsessie

Deze kunnen ook samen, maar apart van groep A. Ze draaien veel vaker (elke 1 tot 15 minuten)

en ze grijpen in. Dat is een ander soort risico: een fout hier zet iets uit dat aan hoorde te staan.

Groep C: dit zijn helemaal geen wachters (5 stuks, ongeveer 1.080 regels)

NAAM	REGELS	WAT HET ECHT IS
vragen-sweep	589	maakt een dagelijks rapport
wachtrij-spiegel	337	maakt een uurlijks overzicht
inbox-watch	114	verwerkt binnenkomende mail
whatsapp-monitor	23	maakt een rapport
succesvol-leren-wacht	18	signaleert nieuwe lessen

Deze bewaken niets. Ze maken rapporten of doen werk. Ze heten alleen "wacht" of "monitor".

Die horen in de ochtendbriefing thuis, niet als losse geplande taak met een eigen bericht.

2. CORRECTIE: vijf van die zes waren een meetfout van mij

In de eerste versie van dit verslag stond dat zes wachters stilstonden, tot veertien dagen. Dat klopte niet. Ik heb het nagemeten en het was mijn eigen meetfout.

Wat ik deed: ik keek hoe oud het logbestand van elke wachter was. Een log van veertien dagen

oud leek te betekenen dat de wachter veertien dagen niets had gedaan.

Wat ik daarna testte: schrijven die scripts eigenlijk wel iets als er niets aan de hand is?

WACHTER	SCHRIJFT BIJ EEN GESLAAGDE RUN?
uptime-monitor	nee
dubbelwerk-wachter	nee
cost-weekly-alert	nee
hoofdsessie-wacht	nee
transcript-watchdog	ja

Vier van de vijf schrijven niets als alles goed gaat. Bij die scripts betekent een oud logbestand juist dat er niets mis was. Precies het tegenovergestelde van wat ik concludeerde.

Alle zes staan bovendien gewoon geladen bij launchd, met exitcode 0 op hun laatste run.

Wat blijft staan: transcript-watchdog schrijft wel bij elke run en zijn log was zeven dagen oud. Dat is dus mogelijk een echte storing, en de enige van de zes.

Waarom deze fout hier thuishoort

Dit is exact dezelfde fout als die ik een dag eerder in de MCP-healthcheck vond, alleen omgekeerd. Daar keurde een check iets goed dat stuk was. Hier keurde ik iets af dat werkte.

Beide komen uit dezelfde denkfout: een signaal gebruiken zonder te controleren wat dat signaal

eigenlijk betekent. "Poort antwoordt" is niet "dienst werkt". "Logbestand is oud" is niet "taak draait niet".

Het maakt het onderliggende probleem trouwens niet kleiner. Er is nog steeds geen enkele manier om te zien of een van de 44 wachters is weggevallen. Ik kon het zelf niet eens betrouwbaar meten, en dat is precies het punt.

3. Waarom het nooit één is geworden

Niet uit domheid. Door de volgorde waarin het is ontstaan.

Er ging iets mis. Dat werd opgelost. Om herhaling te voorkomen kwam er een wachter bij. Dat is elke keer op zichzelf de juiste beslissing geweest.

Maar niemand keek ooit naar het totaal. En omdat elke wachter door een andere storing werd geboren, leek het steeds om iets anders te gaan. Terwijl het twintig keer dezelfde vorm was.

Dat is precies hetzelfde patroon als bij de marketing-tools: twaalf keer iets bouwen omdat er elke keer een aanleiding was, en nooit één keer terugkijken naar het geheel.

4. Het voorstel: van 20 naar 3

1. Eén meldwachter.

Vervangt de negen uit groep A. Eén script, één lijst met controles, één bericht per dag. Een nieuwe controle toevoegen kost dan tien regels op een lijst, geen nieuw bouwsel.

2. Eén herstelwachter.

Vervangt de vijf uit groep B. Draait vaak, mag ingrijpen, met strikte grenzen (nooit de eigen bot-agent, dat is al vastgelegd na het incident van 2 augustus).

3. De rapporten naar de ochtendbriefing.

De vijf uit groep C zijn geen wachters. Hun uitkomst hoort in het bericht dat je toch al krijgt.

En het belangrijkste onderdeel

De meldwachter controleert of hij zelf gedraaid heeft. Elke run schrijft een tijdstempel weg. Blijft die achter, dan is dat het eerste wat je in de ochtendbriefing ziet.

En hij meet of ANDERE taken nog geladen staan, niet hoe oud hun logbestand is. Dat verschil is de belangrijkste les van dit onderzoek, en ik moest hem zelf eerst fout doen om hem te vinden.

5. Hoe ik het zou aanpakken

Niet in één klap. Dat is precies hoe de vorige twaalf marketing-tools zijn ontstaan.

1. **Bouw de meldwachter met de negen controles erin.** Laat hem een week naast de bestaande negen meelopen en vergelijk of hij dezelfde dingen ziet.
2. **Klopt dat: zet de negen oude uit.** Niet weggooien, uitzetten. Terugdraaien moet gratis zijn.
3. **Daarna pas groep B.** Die grijpt in, dus daar is de kans op schade groter.
4. **Groep C als laatste,** want dat is alleen verhuizen naar de briefing.

Klaar is het pas als er één bericht per dag komt in plaats van twintig losse, en als een stilgevallen controle binnen een dag zichtbaar is in plaats van na twaalf dagen.

6. Eén wachter voor de hele vloot, niet één per machine

De verleiding is nu om op elke machine één wachter te zetten. Dan gaan we van 44 naar 5, en dat

klinkt goed. Maar dan blijft het kernprobleem staan: als de wachter op Sjakie stilvalt, ziet niemand dat, want die machine bewaakt zichzelf.

Beter is één wachter die de hele vloot meet, met een klein meldpunt op elke machine.

Hoe dat werkt:

- Elke machine schrijft periodiek zijn eigen metingen weg naar één plek (Supabase, waar de wachtrij en de mijlpalen al staan).
- Eén wachter, op de MacBook, leest die metingen en meldt wat er mis is.
- Meldt een machine een tijd lang niets, dan is dát het alarm. Precies het gat dat er nu is: een stille machine is nu onzichtbaar, terwijl stilte juist het duidelijkste signaal is.

Dat lost meteen de Lenovo-kwestie op die vandaag ook speelde: die lag negen uur uit zonder melding, omdat de wachter die dat had moeten zien op die machine zelf stond.

Wat ik voorstel

Dit is advies. Zeg go op wat je wilt.

1. **Bouw één vloot-wachter** die de metingen van alle machines op één plek leest, in plaats van
44 losse wachters die elk hun eigen ding roepen.
2. **Begin met groep A op de MacBook** (negen controles). Grootste winst, kleinste risico.
3. **Laat het een week dubbel lopen** voor er iets wordt uitgezet. Uitzetten, niet weggooien.
4. **Daarna Sjakie, iMac, Paco en de VPS**, in die volgorde.
5. **Groep B (ingrijpen) als laatste**, want een fout daar zet iets uit dat aan hoorde te staan.
6. **En een afspraak voor onszelf**: vanaf nu geen nieuwe losse wachter meer. Een nieuwe controle
wordt een regel op de lijst. Als ik dat toch voorstel, mag je me erop wijzen. Ik deed het gisteren en vandaag zelf nog twee keer.

Eerst nog dit: ik kan niet bij Coco, en Julio en Lenovo staan uit. Voor een compleet beeld moeten die drie nog gemeten worden. Coco heeft een SSH-sleutel nodig die ik niet heb.