



AI Agents

Een zelflerend AI-agentsysteem met bewijsgebaseerd kennisbeheer, deterministische evaluatie en autonome taakuitvoering.

Architectuurdocument 2027 — bijgewerkt april 2026

95 → 167

SIGNAL DENSITY SCORE

573+

COMMITTS

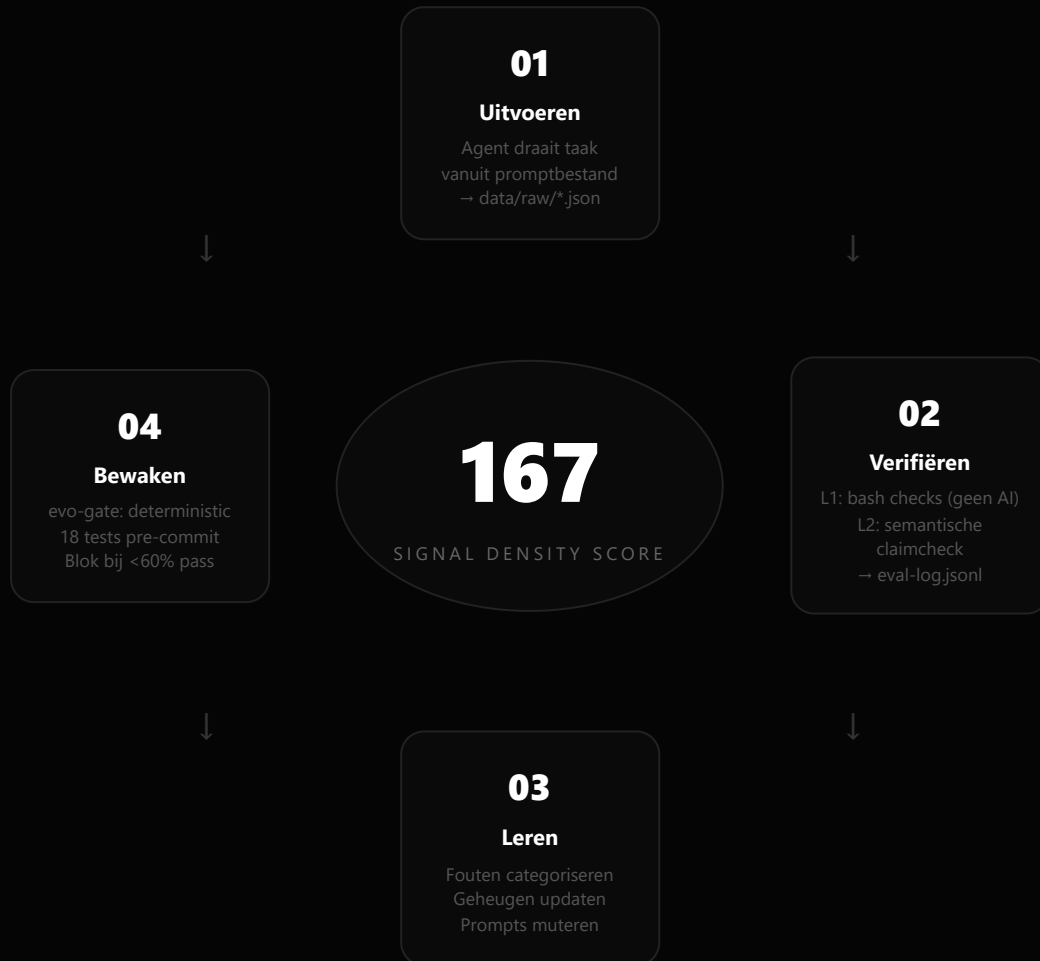
18

PRE-COMMIT TESTS

Onze AI agents draaien 24/7 op een Linux VPS. Ze voeren dagelijks 24 geautomatiseerde taken uit: monitoring, data-analyse, onderzoek (HN, GitHub, Twitter), semantische evaluatie en kennisbankonderhoud. Elke output wordt deterministisch gevalideerd op twee lagen (L1: bash, L2: semantisch). Een enforcement-laag blokkeert acties als de agent zijn eigen kennisbank niet raadpleegt. Elke sessie wordt meetbaar beoordeeld via een scorecard.

De Loop

Vier fasen, continu herhaald. Elke output wordt input voor de volgende cyclus.



Geen claim zonder bewijs. Geen zelfmodificatie zonder gate.

De agent kan zichzelf verbeteren, maar niet zichzelf evalueren.

Enforcement hooks blokkeren acties als de kennisbank niet gelezen is.

Bewijs

Signal Density Score

De SDS meet bruikbaarheid, niet omvang. Ruis verwijderen verhoogt de score. Ongevalideerde notities toevoegen verandert het nauwelijks.

AS	GEWICHT	WAT HET MEET
Signaaldichtheid	50/140	Bewijs per bestand, betrouwbaarheid, validatieratio
Toepassing	50/140	Recentheid (30 dagen), diep bewijs (2+ keer bevestigd)
Structuur	40/140	Categoriedekking, onderwerpdichtheid
Nauwkeurigheid	Straf	−3 per tegengesproken bestand

Kerncijfers (april 2026)

Dagelijkse taken	24 (cron)
Commits (totaal)	573+
Pre-commit tests	18
Kennisbestanden	245

Leerlus-interval	Elke 2 uur
Kennisgraaf	139 nodes, 227 edges
Databronnen	HN, GitHub, Twitter, GSC
Scorecard	Per sessie, correcties + proactieve catches

Architectuur

Ontwerpprincipes

- **Prompt = contract, geen suggestie.** Alles wat niet expliciet staat, wordt creatief geïnterpreteerd.
- **Bewijs boven vertrouwen.** Elke claim moet bewijs citeren: HTTP-status, bestandspad, git hash.
- **Faal luid, niet fout.** Pipelines stoppen bij fouten. Alerts gaan af. Geen stille slechte data.
- **Deterministisch waar mogelijk.** Bash-checks vóór semantische checks. Vertrouw AI nooit om zichzelf te evalueren.
- **Enforce, niet suggereer.** Hooks blokkeren acties als de agent regels niet volgt. Geen vertrouwen op instructies alleen.
- **Signaaldichtheid boven volume.** Slim = nuttige kennis toegepast, niet bestanden opgestapeld.

Repositorystructuur

```
brain/
  patterns/ # Herbruikbare technische patronen
  playbook/ # Causale ketens: als X → dan Y
  mental-models/ # Beslisframeworks en denkpatronen
  clients/ # Bedrijfscontext per klant
  decisions/ # Architectuurbeslissingen met rationale
  data/ # Analyses en afgeleide inzichten
    raw/ # Brondata: API outputs, probes, exports
  claims/ # Ongevalideerde hypotheses
  inbox/ # Onverwerkte items (auto-opgeschoond)
  server/
    prompts/ # Eén .md per taak (≤50 regels)
    tests/ # 18 pre-commit tests
    config.sh # Single source of truth
    run-task.sh # pull → claude → eval → commit → push
```

Stack & Evaluatie

Bewust eenvoudig. Geen frameworks, geen orkestratielagen, geen databases. Elk component is vervangbaar en debugbaar met standaard Unix-tools.

LAAG	TOOL	WAAROM
AI Runtime	Claude Code (headless)	Tool use, bestandstoegang, git, web
Orkestratie	cron + bash	Beproefd, nul afhankelijkheden
State	Git (lokaal + GitHub)	Geversioneerd, diffbaar, gedistribueerd
Evaluatie	bash + python	Deterministisch, geen AI in eval-lus
Alerting	Discord (tweerichtingsbot)	Gratis, instant, bidirectioneel
Dashboard	brain.html (gegenereerd)	Kennisgraaf (139 nodes, 227 edges) + health
Enforcement	Claude Code hooks	Blokkeert acties tot kennisbank gelezen is
Zoeken	QMD (semantisch)	Vectorzoekopdrachten over kennisbank
Verificatie	L1 bash + L2 semantisch	Twee-laags claimvalidatie, geen AI evalueert zichzelf
Config	Enkel bash-bestand	Eén bron van waarheid

18 Pre-Commit Tests

prompt-lint	Max 50 regels	learn-signals	Intervalmismatches
prompt-structure	Geen hardcoded secrets	brain-html	Generator + graaf
config-drift	Sync met config	iq-health	SDS regressie >5pt
cron-integrity	Geen dubbele schema's	regressions	6 historische bugs
data-deps	Pipeline consistentie	learning-loop	Geheugen + reflectie
contracts	JSON output velden	output-quality	Taakoutputs geldig
evo-gate	Pad-regex regressie	eval-trend	7d pass rate $\geq 60\%$
dependencies	Externe tools aanwezig	semantic-eval	L2 claim verificatie
		env-vars	Omgevingsvariabelen
		scripts	Script-integriteit

Veelgestelde Vragen

Hoe weet de AI wat goed en fout is?

De AI weet dat niet zelf. Het systeem werkt in stappen die elkaar controleren: **1.** AI genereert een bevinding. **2.** Een aparte stap opent het bronbestand en controleert. **3.** Resultaat wordt beoordeeld als VALID, PARTIALLY VALID of INVALID — met bewijs. Geen claim zonder bewijs.

Is de Signal Density Score betrouwbaar?

De score is geen absoluut getal maar een nauwkeurigheidspercentage. De trend is wat telt: 95 → 167 in 7 maanden. Elke week worden bevindingen onafhankelijk gecontroleerd. Springt het ineens 30 punten? Alert.

Wat als een mens foute input geeft?

Nieuwe input begint met lage betrouwbaarheid. Die stijgt alleen als de data bevestigt. **Expert-pad:** input van een expert begint als hypothese met medium confidence en doorloopt een versneld validatiepad.

Wat als de expert en de AI het oneens zijn?

Objectief fout: AI corrigeert direct. **Expert weet meer:** versneld validatiepad. **Grijze zone:** AI geeft perspectief met reden, vraagt waarom de expert anders denkt. Bij onopgelost verschil: opslaan als spanning, monitoren.

Hoe weet je of de AI echt beter wordt?

Elke sessie logt een scorecard: hoeveel fouten de mens ving vs. hoeveel de AI zelf ving. Die ratio moet verschuiven. Daarnaast blokkeert een enforcement-laag de AI als deze zijn eigen geheugen niet leest — geen suggestie, maar een harde blokkade.

Wat zijn de grenzen van AI-verificatie?

L2-verificatie is AI-gebaseerd en kan blinde vlekken hebben. Handmatige correcties dienen als externe kalibratie. Als correcties toenemen, wordt de verificatielogica hertraint. Menselijke feedback is structureel, geen noodoplossing.

Bijlage: 10 Stadia

Van praten met AI tot volledig autonome systemen. Progressie is niet lineair — hogere stadia kunnen eerder rijpen dan lagere.

01	Conversationale AI	VOLTOOID	Mens typt, AI antwoordt. Geen geheugen, geen tools.
02	AI-Ondersteund Coderen	VOLTOOID	AI vult code aan in je editor. Mens stuurt, AI suggereert.
03	CLI Agent	VOLTOOID	AI voert meerstaps-taken uit vanuit terminal.
04	Persistente Agent	VOLTOOID	AI draait continu met geheugen en communicatiekanalen.
05	Gestructureerd Kennisbrein	VOLTOOID	Gecureerde kennisbank met betrouwbaarheidsscores.
06	Zelfverbeterende Agent	VOLTOOID	24/7, observeert, leert, verbetert eigen prompts. Evo-gate voorkomt drift.
07	Autonome Laagrisico-Acties	70%	Auto-issues en systeemfixes live. Auto-PRs op klantcode bewust uitgesteld tot kwaliteitsgate gehaald.
08	Cross-Project Intelligentie	40%	Patronen gedeeld en gevalideerd over 6 projecten. Automatische toepassing nog niet gebouwd.
09	Agent-Gestuurde Strategie	85%	Wekelijkse Deep Think: data-analyse, strategische actie + inversie. Mens beslist, agent denkt mee.
10	Volledig Autonome Operatie	VISIE	Mens als eigenaar, niet operator. Het Rick Rubin-model.

Stadia 7–9 lopen parallel. Het systeem denkt al strategisch (9) maar handelt nog niet autonoom op klantcode (7). De bottleneck is niet denken maar doen — en dat is een bewuste ontwerpkeuze.

AI Agents is gebouwd door AIFAIS, AI-gedreven productconsultancy.

aifais.com/ai-agents