

Agentic design.

Agents ontwerp je zoals je een team ontwerpt. Een werkwijze om agentic systemen te ontwerpen met heldere rollen, context, grenzen en verantwoordelijkheid.

AUTEUR

Dusty Baars

UITGEVER

Mono by Dusty

DATUM

juni 2026

LEESTIJD

18 minuten

INHOUD

00 **Samenvatting** Wat dit document beweert, in vier punten

II **Wat agentic design is** Een systeem, geen los model

IV **In de praktijk** Vier systemen die in productie draaien

VI **Waar dit heen beweegt** Samenwerkende agents, en waarom ontwerp blijft tellen

I **Waarom agents blijven steken** Het ontbreekt zelden aan techniek, vaak aan ontwerp

III **De ontwerpprincipes** Vijf principes die een systeem laten standhouden

V **Organisatie en wat menselijk blijft** Eigenaarschap, governance en het review-ritme

· **Slot** Een uitnodiging, en over de auteur

Wat dit document beweert, in vier punten.

Veel organisaties hebben inmiddels ervaring met AI. Er draaien copilots, er staan automatiseringen, en de eerste agents zijn gebouwd. Toch ontbreekt vaak een heldere manier van denken over hoe zulke systemen veilig, beheersbaar en praktisch inzetbaar worden. Het gesprek gaat over modellen, terwijl het over ontwerp zou moeten gaan.

Dit document beschrijft hoe ik agentic systemen ontwerp. Niet vanuit de vraag welk model het best presteert, maar vanuit de vraag hoe werk binnen een organisatie verloopt. De aanpak rust op vier samenhangende keuzes.

01

Een ontwerpvraagstuk, geen modelvraagstuk

De waarde ontstaat niet bij het model, maar bij het systeem eromheen. Rollen, context, grenzen en verantwoordelijkheid bepalen of een agent bruikbaar wordt in de praktijk of indrukwekkend blijft in een demo.

02

Ontwerp rollen expliciet, zoals je een team ontwerpt

Een agent zonder duidelijke rol wordt een generieke assistent die overal een beetje bij betrokken is en nergens echt verantwoordelijkheid draagt. Gespecialiseerde rollen maken een systeem overzichtelijker en betrouwbaarder.

03

Context is het fundament, niet een extra

Toegang tot de juiste informatie, kennis van eerdere stappen, begrip van bedrijfsregels en zicht op de actuele status zijn voorwaarden. Is die laag zwak, dan blijft elke implementatie afhankelijk van toeval, prompts en handmatige correctie.

04

Autonomie blijft gekoppeld aan verantwoordelijkheid

Zodra een agent invloed heeft op processen, communicatie of besluitvorming, moeten de grenzen expliciet zijn. Wat mag autonoom, wat vraagt goedkeuring, wat wordt gelogd, en waar grijpt een mens in. Dat is geen rem op innovatie, het is de voorwaarde om AI serieus en duurzaam in te zetten.

In de rest van dit document werk ik die vier keuzes uit. Eerst beschrijf ik waarom agents in de praktijk zo vaak blijven steken. Daarna wat ik onder agentic design versta, gevolgd door de ontwerpprincipes die ik aanhoud. Vervolgens laat ik aan de hand van mijn eigen systemen zien hoe dat eruitziet, en wat de invoering van een organisatie vraagt. Tot slot kijk ik naar waar dit heen beweegt.

DEEL I

Waarom agents blijven steken.

De technologie is zelden het probleem. Modellen zijn goed genoeg, de tooling is volwassen, en een agent in elkaar zetten kan tegenwoordig in een middag. Wat ontbreekt is het ontwerp eromheen. Daardoor blijven agents in veel organisaties steken op dezelfde plekken.

Het begint vaak met losse oplossingen zonder onderliggende ontwerpprincipes. Een assistant voor content, een flow voor support, een agent voor interne kennis. Stuk voor stuk nuttig, maar zonder samenhang groeit de complexiteit sneller dan de waarde. Niemand overziet nog wie wat doet, en het onderhoud wordt zwaarder dan het werk dat het bespaart.

Een tweede patroon is autonomie op het verkeerde moment. Agents krijgen te vroeg te veel ruimte, of juist te weinig context om goed werk te leveren. Zonder grenzen ontstaat risico. Zonder context ontstaat ruis. In beide gevallen wordt AI vooral iets dat extra controle vraagt, in plaats van iets dat werk verlicht. De belofte keert zich om.

Het derde patroon is een te technische of te gefragmenteerde aanpak. Iets werkt op één plek, in handen van degene die het bouwde, maar het laat zich niet betrouwbaar breder uitrollen. De stap van een werkende proof naar een systeem waar de organisatie op durft te leunen wordt onderschat. Daar sneuvelen de meeste initiatieven, niet op de techniek maar op het ontwerp.

Agentic AI wordt benaderd als een technisch vraagstuk, terwijl het een ontwerpvoorbeeld is. Zodra je het zo bekijkt, verschuift de eerste vraag. Niet welk model, maar hoe het werk verloopt, waar context verloren gaat, en waar verantwoordelijkheid expliciet moet blijven.

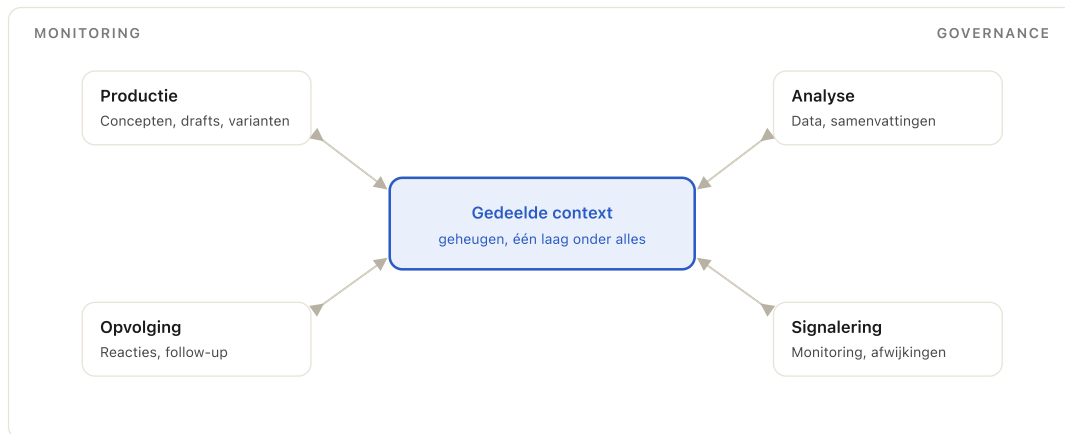
DEEL II

Wat agentic design is.

Agentic design betekent dat je agents ontwerpt zoals je een sterk team ontwerpt. Elke agent heeft een duidelijke rol, een afgebakend doel, toegang tot de juiste informatie en tools, en heldere regels over wat zelfstandig mag en waar menselijke tussenkomst nodig is. Dezelfde dingen die een team laten werken, laten een systeem van agents werken.

Dat vraagt om meer dan een goed model. Het vraagt om een systeem dat rekening houdt met samenwerking, context, geheugen, informatiestromen, monitoring en governance. Het model is één onderdeel. De rest bepaalt of het geheel in de praktijk standhoudt. Pas dan ontstaat een vorm van AI die niet los boven de organisatie hangt, maar er onderdeel van wordt.

AGENTIC DESIGN IN ÉÉN BEELD



In het midden een gedeelde context- en geheugenlaag. Daaromheen agents met elk een eigen rol, die uit die laag putten en eraan toevoegen. Eromheen lopen monitoring en governance, die zien wat er gebeurt en waar een mens beslist. Geen verzameling losse tools, maar een geheel met een binnenkant en een buitenkant.

Het beeld hierboven vat het samen. In het midden staat een gedeelde context- en geheugenlaag. Daaromheen agents met elk een eigen rol, die uit die laag putten en eraan toevoegen. Eromheen lopen monitoring en governance, die zien wat er gebeurt en waar een mens beslist. Het is geen verzameling losse tools, het is een geheel met een binnenkant en een buitenkant.

Het verschil met de gangbare aanpak zit in de volgorde. De meeste implementaties beginnen bij het model en de prompt, en hopen dat de rest zich vanzelf vormt. Agentic design begint bij de rollen en de informatiestromen, en kiest het model pas als het werk helder is. De techniek volgt het ontwerp, niet andersom.

DEEL III

De ontwerpprincipes.

Vijf principes dragen deze aanpak. Geen ervan is op zichzelf opzienbarend. De combinatie maakt dat een systeem in productie ook echt waarde levert.

Begin bij het werk.

Niet bij tooling, maar bij het proces zelf. Waar zit vertraging, waar gaat context verloren, waar is herhaling, en waar zijn mensen te veel tijd kwijt aan coördinatie in plaats van inhoud. Niet elke taak leent zich voor agentic AI. De beste eerste toepassingen zitten in processen met duidelijke uitkomsten, meerdere overdrachtsmomenten en een grote behoefte aan consistentie, snelheid en goede context.

Ontwerp rollen expliciet.

Een agent zonder duidelijke rol zakt weg in vaagheid. Gespecialiseerde rollen, elk met een eigen taak en een duidelijke plek in het geheel, maken het systeem overzichtelijker en betrouwbaarder. Zodra helder is wie of wat waarvoor verantwoordelijk is, wordt de samenwerking tussen mensen en agents stuurbaar.

EEN AGENT, ONTWORPEN ALS TEAMLID

| ROL IN HET TEAM | |
|------------------------------|--|
| Een agent als teamlid | |
| ROL | Wat het is, één duidelijke taak |
| DOEL | Een afgebakende uitkomst |
| TOEGANG | De juiste context en tools |
| GRENZEN | autonoom goedkeuring |

Een agent leest als een teamprofiel. De rol zegt wat het is, het doel geeft een afgebakende uitkomst, de toegang regelt context en tools, en de grenzen maken expliciet wat autonoom mag en wat goedkeuring vraagt.

Zie context als fundament.

Een agent is alleen bruikbaar als de context goed georganiseerd is. Toegang tot de juiste informatie, kennis van eerdere stappen, begrip van bedrijfsregels en zicht op de actuele status zijn geen extra's maar voorwaarden. Daarom besteed ik veel aandacht aan hoe informatie beweegt binnen een systeem. Is die laag zwak, dan blijft elke implementatie afhankelijk van toeval.

Houd autonomie gekoppeld aan verantwoordelijkheid.

Zodra een agent invloed heeft op processen, communicatie of besluitvorming, moeten de grenzen expliciet zijn. Wat mag autonoom, wat vraagt goedkeuring, wat wordt gelogd, en waar grijpt een mens in. Voor mij is dat geen rem op innovatie, maar juist de voorwaarde om AI serieus en duurzaam in te zetten.

Bouw voor schaal, maar begin klein.

Klein beginnen met een proces dat direct relevant is voor de business. Niet omdat de ambitie klein moet zijn, maar omdat goed ontwerp zich eerst in de praktijk moet bewijzen. Een sterk eerste systeem laat zien dat de technologie werkt, en dat de organisatie klaar is om ermee te werken. Daarna wordt opschalen een logische stap in plaats van een sprong in het diepe.

DEEL IV

In de praktijk.

Deze aanpak komt niet uit een boek. Ik heb hem gebouwd, in systemen die draaien. Vier daarvan laten elk een ander stuk van het principe zien.

Mono Dash, rollen en orkestratie

Mono Dash stuurt teams van agents aan met elk een eigen rol en mandaat. Niet één model dat alles probeert, maar gespecialiseerde rollen die samen een proces dragen, met een coördinatielaag erboven. Het maakt zichtbaar wat het tweede principe in de praktijk betekent: een systeem wordt betrouwbaarder zodra verantwoordelijkheid is verdeeld in plaats van opgehoopt.

Open Brain, context als fundament

Open Brain is een geheugenlaag die over applicaties en sessies heen onthoudt. Gedachten, beslissingen en patronen worden opgeslagen en teruggevonden via betekenis, niet via trefwoord. Het is de praktische invulling van het derde principe. Agents putten uit een gedeelde context en voegen eraan toe, zodat kennis niet in losse tools of in iemands hoofd blijft hangen. Gebouwd op MCP, zodat dezelfde laag vanuit meerdere applicaties bereikbaar is.

Memortium Studio, autonomie met een grens

Memortium Studio bewerkt portretfoto's van overledenen. Een pipeline genereert automatisch varianten, en een validatiestap bewaakt dat de identiteit niet verandert. Het principe is hard: bij twijfel niet leveren. Lukt het automatisch niet goed genoeg, dan escaleert het werk naar een mens. AI-first, met een menselijke fallback. Het vierde principe in zijn scherpste vorm, omdat de inzet hoog is en een fout niet bij een agent hoort te liggen.

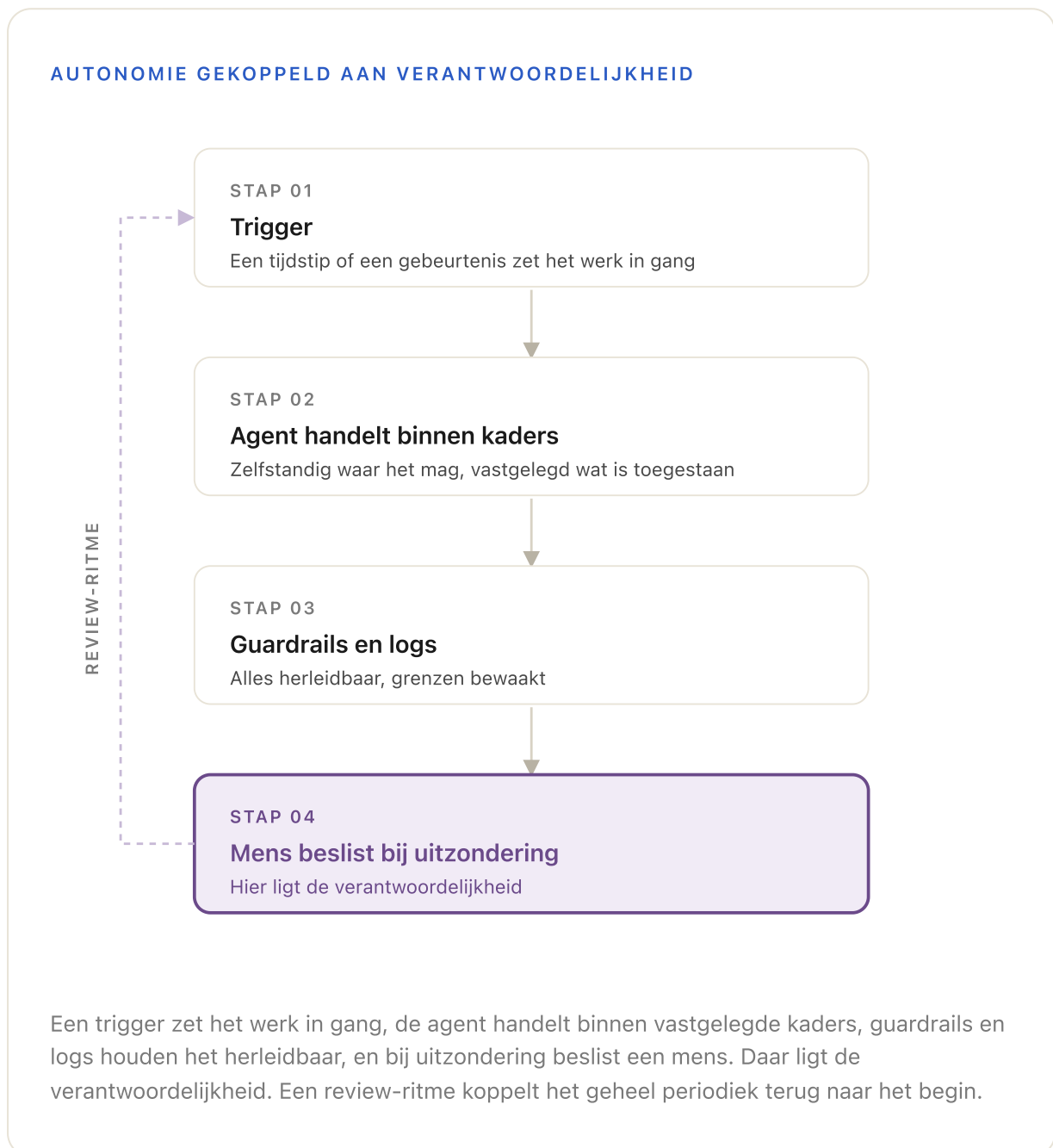
De Memortium marketing-machine, een team van agents

Hier werken meerdere agents met eigen rollen samen aan content, outreach en opvolging, met een mens die kaders zet en bij uitzonderingen beslist. Het laat zien hoe de principes samenkomen: rollen expliciet, context gedeeld, autonomie begrensd, en klein begonnen voordat het breder ging.

Wat deze systemen verbindt, is niet het model. Het is het ontwerp eromheen. In elk geval begon het bij het werk, kregen de agents een duidelijke rol, was de context geregeld, en lag de eindverantwoordelijkheid bij een mens.

Wat dit van een organisatie vraagt, en wat menselijk blijft.

Een aanpak werkt zelden alleen omdat hij op papier klopt. De invoering vraagt iets van de organisatie zelf, en het vraagt dat een paar dingen bewust menselijk blijven.



Wat in mensenhanden blijft.

Strategie, klantrelatie, concept- en richtingkeuzes, finale approvals en eindverantwoordelijkheid blijven bewust mensenwerk. Een agent kan een voorstel voorbereiden, een mens tekent ervoor. De verantwoordelijkheid die hoort bij keuzes wanneer iets misgaat, ligt bij mensen, niet bij een systeem. Dit onderscheid expliciet maken beschermt niet alleen wat menselijk is, het geeft ook rust. Wie weet waar de mens onmisbaar blijft, kijkt met meer vertrouwen naar de plekken waar een agent wel kan landen. Het gesprek gaat dan over inrichting, niet over angst.

Wat de organisatie zelf moet regelen.

Een intern eigenaar, iemand met mandaat over scope, kwaliteit en doorontwikkeling, zodat een agent onderdeel van het bedrijf wordt en geen afhankelijkheid van de bouwer blijft. Een afspraak over governance: wie ziet welke data, waar belandt die, wie heeft inzage in de logs, en wie tekent voor uitzonderingen. En een review-ritme, want een agent in productie hoort periodiek tegen het licht. Werkt hij nog zoals afgesproken, levert hij nog de uitkomst waarvoor hij is gebouwd, en zijn de grenzen nog passend.

Dit is voor de meeste organisaties geen ingewikkeld document. Het is wel een onderwerp dat vooraf op tafel hoort, omdat het privacy, herleidbaarheid en aansprakelijkheid concreet maakt in plaats van abstract.

DEEL VI

Waar dit heen beweegt.

Het veld beweegt snel, en de richting is duidelijk. Losse agents worden samenwerkende agents. Protocollen als MCP maken context en tools breed bereikbaar, en agent-naar-agent-communicatie maakt dat systemen elkaar werk kunnen doorgeven. Guardrails en evals worden volwassen, zodat betrouwbaarheid meetbaar wordt in plaats van een gevoel. De vraag verschuift van "kan een agent dit" naar "kunnen we erop vertrouwen dat hij het goed doet, keer op keer".

Precies daar wordt ontwerp het verschil. Naarmate agents meer zelf doen, telt de inrichting eromheen zwaarder, niet lichter. Rollen, context, grenzen en verantwoordelijkheid zijn geen tijdelijke randvoorwaarden die verdwijnen als de modellen beter worden. Ze zijn de blijvende kern. Een beter model maakt een slecht ontworpen systeem niet betrouwbaar, het maakt de fouten alleen sneller.

Ik bouw met agents, orchestratie en automatisering, en duw op de grenzen om te zien wat mogelijk wordt. Niet om de mens te vervangen, maar om werk te verlichten en ruimte te maken voor het werk dat er echt toe doet. Dat is de richting waarin ik wil werken, en waarin ik voorop wil lopen.

SLOT

Een uitnodiging.

Agentic AI wordt pas waardevol wanneer het niet draait om losse prompts of slimme demo's, maar om ontwerp. Ik implementeer agentic systemen zoals ik een organisatie zou ontwerpen: met duidelijke rollen, goede informatiestromen, heldere grenzen en verantwoordelijkheid op de juiste plek.

Wie hierover wil doordenken, voor een team of een organisatie die AI verder wil brengen dan losse experimenten, praat ik graag verder.

Over de auteur

Dusty Baars is designer turned AI builder. Met AI bouwt hij oplossingen die mensen echt gebruiken, vanuit één overtuiging: goede technologie maakt mensen beter in plaats van ze te vervangen. Ruim twaalf jaar enterprise-UX bij Shell, Roche, Philips en ING, via Plat4mation. De laatste twee jaar bouwde hij AI in productie: Memortium, Open Brain, Mono Dash. Hij werkt vanuit Amsterdam.

Mono by Dusty · Amsterdam · contact@dustybaars.com

Voor een eerste verkenning of om kennis te maken is een korte mail het snelste begin.