Wat is een MCP?

En hoe je een AI veilig op een echt systeem laat werken. Het patroon, niet de code.

TICO VAN GERNER
Customer Experience + AI

Een korte gids · mei 2026

Waarom deze gids

Ik bouwde een MCP-server waarmee een AI-agent echt kan werken in een enterprise-systeem: lezen, configureren, records aanmaken via de officiële API. De vraag die ik daarna het vaakst kreeg was niet “mag ik je code” maar “hoe doe je dat veilig?”. Dat is de juiste vraag. Een AI loslaten op een live systeem van een klant kan namelijk gigantisch misgaan. Deze gids geeft het antwoord: niet mijn code, wel het patroon. De manier van denken die het verschil maakt tussen een agent die je vertrouwt en een ongeluk dat staat te wachten. Het is vendor-neutraal: het werkt voor SAP, voor elk CRM, en voor vrijwel elk systeem met een API.

Wat is een MCP, in gewone taal

MCP staat voor Model Context Protocol. Het is een open standaard die een AI-model gereedschap geeft. Normaal is een model als Claude goed in één ding: tekst genereren. Het kan niets ín je systemen. Een MCP verandert dat. Het is de brug tussen het model en een applicatie, en het levert de AI een set “tools”: afgebakende handelingen die hij echt mag uitvoeren, zoals een record opzoeken of een wijziging klaarzetten.

De vergelijking die werkt: zonder MCP is de AI een adviseur die naast je zit en zegt wat je moet doen. Met een MCP krijgt diezelfde adviseur handen, en kan hij het zelf doen. Dat is krachtig. En precies daarom moet je nadenken over wat die handen wel en niet mogen.

Van prater naar doener: wat er verandert

Zodra een AI handelingen kan verrichten, verschuift het risico. Een verkeerd antwoord in een chat is vervelend maar onschuldig. Een verkeerde actie in een live systeem raakt echte data, echte klanten, echte processen. De winst is enorm — werk dat normaal klik voor klik handmatig gebeurt, wordt reproduceerbaar en snel — maar de fout is dat ook. Het hele vraagstuk wordt daarmee: hoe geef je een agent genoeg vrijheid om nuttig te zijn, zonder dat één misverstand schade aanricht? Het antwoord zit niet in de slimheid van het model. Het zit in de discipline die je in het gereedschap zelf inbouwt.

De zes guardrails

Dit is het patroon. Zes principes die samen een veiligheidsriem vormen. Ze gelden voor elke AI-agent die in een echt systeem mag werken, niet alleen voor de mijne.

1. Preview-then-apply. De agent kan nooit direct schrijven. Elke wijziging gebeurt in twee stappen: eerst tonen wat er precies gaat gebeuren, een mens keurt het goed, en pas daarna voert de agent het uit. Dit ene principe vangt de meeste ongelukken af, omdat er altijd een menselijk moment tussen intentie en uitvoering zit.

2. Productie is een aparte drempel. Werken in een test- of acceptatieomgeving is één ding. Schrijven naar een live productieomgeving hoort een expliciete, extra bevestiging te vereisen. Maak het onmogelijk om er per ongeluk in te belanden.

3. Least privilege. De agent krijgt alleen toegang tot wat de taak nodig heeft, en niet meer. Net als bij autorisatierollen voor mensen: je geeft niet iedereen admin-rechten. Een agent die categorieën aanmaakt, hoeft geen contracten te kunnen verwijderen.

4. Een volledige audit-trail. Elke actie die de agent uitvoert, wordt vastgelegd: wat, wanneer, met welk resultaat. Zonder logboek weet je achteraf niet wat er is gebeurd, en kun je een fout niet terugvinden of herstellen.

5. Read-back verificatie. Vertrouw niet blind op “gelukt”. Laat de agent na een actie teruglezen wat er werkelijk in het systeem staat, en laat hem melden wanneer het systeem stilletjes iets heeft aangepast of laten vallen. Wat je denkt dat er gebeurde en wat er echt gebeurde, zijn niet altijd hetzelfde.

6. Een noodrem. Bouw een fail-safe in die ingrijpt als het misgaat — bijvoorbeeld stoppen na een reeks mislukte pogingen, in plaats van blind doorgaan. Een agent zonder noodrem maakt van een klein probleem een groot probleem.

Mens-in-de-lus: waarom dit het verschil maakt

De rode draad door alle zes is dezelfde: vraag het bij twijfel, en hou de mens in de lus. Niet omdat de AI dom is, maar omdat de kosten van een fout in een echt systeem te hoog zijn om volledig op automatisme te vertrouwen. Het mooie is dat deze discipline geen rem op de waarde is — het is juist wat de waarde mogelijk maakt. Organisaties die hun agents netjes inkaderen, durven ze méér vrijheid te geven, omdat ze weten dat de schade beperkt blijft als er iets misgaat. Vertrouwen bouw je niet door de agent los te laten, maar door de grenzen scherp te zetten.

In het kort

1. Een MCP geeft een AI gereedschap: van prater naar doener.
2. Zodra een AI kan handelen, verschuift het risico naar echte data en processen.
3. De veiligheid zit niet in het model, maar in het gereedschap: preview-then-apply, een aparte productiedrempel, least privilege, een audit-trail, read-back verificatie en een noodrem.
4. De rode draad: mens-in-de-lus, vraag het bij twijfel.

Tot slot

Verantwoorde AI is geen marketingterm. Het is een set concrete keuzes die je in je gereedschap inbouwt, vóórdat je een agent op een echt systeem loslaat. Wie dat goed doet, plukt de vruchten van automatisering zonder de nachtmerrie van een agent die op hol slaat.

Werk je aan AI-agents op je CRM of een ander enterprise-systeem, of denk je erover na? Daar help ik organisaties bij, en ik ben bezig met live trainingen over precies dit soort verantwoorde inzet. Connect gerust op LinkedIn of reageer op een van mijn posts. Ik wissel hier graag over uit.

Tico van Gerner
Customer Experience + AI