Het Agent Manifesto

Een grondwet voor betrouwbare, bestuurbare en samenwerkende AI-agents

Versie: 0.1 **Status:** Concept **Uitgave:** Etine — augustus 2026 — etine.nl **Doel:** Een vendor-onafhankelijk fundament voor het ontwerpen, inzetten en besturen van AI-agents.

> **Geen agency zonder identiteit. Geen autonomie zonder grenzen. Geen handeling zonder verantwoordelijkheid.**

Over dit document. Het Agent Manifesto is het ontwerpkompas dat Etine hanteert bij het ontwerpen, inzetten en besturen van AI-agents. We publiceren het als open concept: reacties, kritiek en bijdragen zijn welkom via etine.nl. Dit document mag vrij worden gedeeld met bronvermelding.

Preambule

AI-agents ontwikkelen zich van hulpmiddelen die antwoorden formuleren tot operationele actoren die waarnemen, redeneren, beslissingen voorbereiden, systemen bedienen en taken aan elkaar delegeren.

Daarmee verandert hun positie.

Een agent die alleen tekst genereert, is vooral een instrument. Een agent die zelfstandig e-mail leest, klantgegevens verwerkt, software aanpast, betalingen voorbereidt of andere agents aanstuurt, neemt daadwerkelijk deel aan een organisatie.

Zo'n agent kan niet uitsluitend worden beschreven als software, model of API.

We moeten ook weten:

- wie de agent is;
- waarom de agent bestaat;
- waar de agent leeft;
- welk model zijn cognitieve vermogen levert;
- wie hem heeft aangesteld;
- welke bevoegdheden hij bezit;
- welke middelen hij mag gebruiken;
- met wie hij mag samenwerken;
- wie verantwoordelijk is voor zijn handelen;
- hoe zijn gedrag kan worden waargenomen;
- hoe hij kan worden begrensd, geschorst en beëindigd.

Dit manifesto stelt daarom een eenvoudig uitgangspunt centraal:

> **Geen agency zonder identiteit. Geen autonomie zonder grenzen. Geen handeling zonder verantwoordelijkheid.**

Het Agent Manifesto is geen protocol en geen technisch bestandsformaat. Het is een verzameling beginselen waarop agentarchitecturen, agentmanifests, runtimes, control planes en governance-systemen kunnen worden gebaseerd.

---

1. Onze waarden

Wij verkiezen:

Identificeerbare agents boven anonieme automatisering

Iedere operationele agent moet herkenbaar zijn als een afzonderlijke entiteit met een naam, eigenaar, versie, doel en levenscyclus.

Een expliciet mandaat boven impliciet gedrag

Een agent mag niet zelf bepalen waarom hij bestaat. Zijn opdracht, gewenste uitkomsten en verboden handelingen moeten vooraf worden vastgelegd.

Begrensde autonomie boven onbeperkte capaciteit

Niet alles wat een agent technisch kan, moet hij organisatorisch mogen.

Bewijsbaar gedrag boven verondersteld vertrouwen

Vertrouwen ontstaat niet doordat een agent intelligent lijkt, maar doordat zijn identiteit, handelingen, beslissingen en resultaten aantoonbaar zijn.

Menselijke verantwoordelijkheid boven machineverwijt

Een agent kan een taak uitvoeren, maar kan geen uiteindelijke morele, juridische of bestuurlijke verantwoordelijkheid dragen.

Open interfaces boven verborgen afhankelijkheden

Een agent moet kunnen worden begrepen en bestuurd zonder volledig afhankelijk te zijn van één modelleverancier, runtime of platform.

Observeerbare uitvoering boven ondoorzichtige uitkomsten

Niet alleen het eindresultaat telt. Ook de route, gebruikte middelen, delegaties, fouten en interventies moeten zichtbaar zijn.

Herroepbare bevoegdheden boven permanente toegang

Iedere bevoegdheid moet kunnen worden aangepast, beperkt of ingetrokken.

Samenwerking onder bestuur boven ongecontroleerde agentzwermen

Agents mogen samenwerken, maar niet buiten een herkenbare structuur van eigenaarschap, delegatie en toezicht.

Gedeclareerde werkelijkheid boven verborgen werkelijkheid

De agent die werkelijk draait, moet overeenkomen met de agent die volgens zijn manifest hoort te bestaan.

Hoewel de elementen aan de rechterkant soms noodzakelijk zijn, kennen wij meer waarde toe aan de elementen aan de linkerkant.

---

2. Wat wij onder een agent verstaan

Een **agent** is een operationele software-entiteit die binnen een bepaalde context:

1. informatie kan waarnemen;
2. een toestand of doel kan interpreteren;
3. keuzes kan maken of voorstellen;
4. hulpmiddelen en systemen kan gebruiken;
5. handelingen kan uitvoeren of delegeren;
6. gedurende een bepaalde periode een identiteit en werkcontext behoudt.

Niet ieder taalmodel is een agent.

Niet iedere automatisering is een agent.

Niet iedere workflow is een agent.

Een agent ontstaat wanneer cognitieve capaciteit, handelingsvermogen, context, identiteit en een doel in één bestuurbare operationele eenheid worden samengebracht.

---

3. De fundamentele onderdelen van een agent

Onderdeel	Betekenis
Identiteit	Wie de agent is
Doel	Waarom de agent bestaat
Mandaat	Wat de agent namens de organisatie mag doen
Cognitie	Welk model of redeneersysteem wordt gebruikt
Geheugen	Welke informatie de agent tussen handelingen bewaart
Host	Waar de agent leeft en wordt uitgevoerd
Runtime	De software die zijn gedrag organiseert
Tools	De systemen waarmee hij handelingen uitvoert
Resources	De gegevens en objecten die hij mag benaderen
Sessies	Tijdelijke werkcontexten waarin taken worden uitgevoerd
Relaties	Andere agents, mensen en systemen waarmee hij samenwerkt
Observability	Het bewijs van wat hij werkelijk doet
Governance	De regels waaronder hij opereert
Levenscyclus	Hoe hij wordt aangesteld, gewijzigd, geschorst en beëindigd

4. De constitutionele principes

Principe 1 — Iedere agent heeft een identiteit

Een agent mag niet uitsluitend worden aangeduid met de naam van zijn model, container, prompt of proces.

De identiteit van een agent moet ten minste bestaan uit:

- een unieke agent-ID;
- een herkenbare naam;
- een versie;
- een organisatorische eigenaar;
- een technisch verantwoordelijke;
- een lifecycle-status;
- een verwijzing naar zijn geldende manifest.

Een model kan door meerdere agents worden gebruikt. De identiteit van de agent mag daarom nooit samenvallen met de identiteit van het model.

`gpt-5` is geen agentidentiteit.

`invoice-processing-agent-prod-v3` kan dat wel zijn.

Principe 2 — Iedere agent leeft ergens

Een agent bestaat nooit in abstractie. Hij wordt uitgevoerd op een concrete infrastructuur onder verantwoordelijkheid van een concrete partij.

Voor iedere agent moet duidelijk zijn:

- op welke host of runtime hij draait;
- in welk land of welke regio hij wordt uitgevoerd;
- wie de infrastructuur beheert;
- of de omgeving lokaal, self-hosted, cloud-hosted of door een leverancier beheerd is;
- welke delen van de agent buiten de eigen beheergrens draaien;
- waar gegevens, geheugen en telemetry worden opgeslagen.

De locatie van het model kan verschillen van de locatie van de runtime.

Een agent kan bijvoorbeeld op een eigen Kubernetes-cluster leven terwijl zijn model via een externe API wordt aangeroepen. Beide werkelijkheden moeten zichtbaar zijn.

Principe 3 — Iedere agent heeft een expliciet doel

Een agent mag alleen handelen vanuit een gedeclareerd doel.

Het doel moet beschrijven:

- welk probleem de agent oplost;
- voor wie hij dat doet;
- welke uitkomsten gewenst zijn;
- welke uitkomsten niet zijn toegestaan;
- hoe succes wordt gemeten;
- wanneer de opdracht als voltooid geldt.

Een opdracht als “help de financiële administratie” is onvoldoende begrensd.

Een opdracht als “classificeer binnenkomende facturen en maak boekingsvoorstellen, maar voer nooit zelfstandig betalingen uit” bevat wel een herkenbaar mandaat.

Principe 4 — Een agent heeft geen inherente bevoegdheden

Een agent bezit geen rechten omdat hij intelligent, technisch geavanceerd of nuttig is.

Alle bevoegdheden worden toegekend.

Een agent mag uitsluitend beschikken over de toegang die noodzakelijk is voor zijn opdracht. Bevoegdheden moeten:

- doelgebonden zijn;
- zo beperkt mogelijk worden verstrekt;
- een eigenaar hebben;
- technisch afdwingbaar zijn;
- periodiek worden herzien;
- onmiddellijk kunnen worden ingetrokken.

Het kunnen aanroepen van een tool is niet hetzelfde als toestemming hebben om iedere functie van die tool te gebruiken.

Principe 5 — Autonomie moet proportioneel zijn aan risico

Autonomie is geen binaire eigenschap. Een agent is niet simpelweg autonoom of niet-autonoom.

Zijn handelingsruimte moet afhangen van:

- de omkeerbaarheid van een handeling;
- de mogelijke financiële impact;
- de gevoeligheid van de gegevens;
- de juridische gevolgen;
- de mate van onzekerheid;
- de reikwijdte van het resultaat;
- de mogelijkheid van menselijke controle.

Een agent mag relatief zelfstandig openbare informatie ordenen. Dezelfde agent mag niet vanzelfsprekend zelfstandig medewerkers ontslaan, betalingen uitvoeren of productieomgevingen verwijderen.

Hoe groter de impact, hoe sterker de vereiste controle.

Principe 6 — Menselijke verantwoordelijkheid blijft bestaan

Een agent kan namens mensen en organisaties handelen, maar neemt hun uiteindelijke verantwoordelijkheid niet over.

Voor iedere agent moeten ten minste twee rollen herkenbaar zijn:

De accountable owner

De persoon of functie die bestuurlijk verantwoordelijk is voor het doel, het mandaat en de gevolgen van de agent.

De technical owner

De persoon of functie die verantwoordelijk is voor de implementatie, beveiliging, beschikbaarheid en technische werking.

De uitspraak “de AI heeft het gedaan” is nooit een geldige afsluiting van verantwoordelijkheid.

Principe 7 — Iedere betekenisvolle handeling moet herleidbaar zijn

Voor iedere relevante handeling moet achteraf kunnen worden vastgesteld:

- welke agent de handeling uitvoerde;
- onder welke versie van het manifest;
- binnen welke sessie of taak;
- op wiens verzoek;
- met welk model;
- met welke instructies en policies;
- welke gegevens zijn gebruikt;
- welke tools zijn aangeroepen;
- welke andere agents betrokken waren;
- welk resultaat is geproduceerd;
- welke goedkeuringen zijn verkregen.

Herleidbaarheid betekent niet dat alle vertrouwelijke inhoud onbeperkt moet worden gelogd. Het betekent dat voldoende bewijs moet bestaan om gedrag te onderzoeken zonder privacy en beveiliging onnodig te schaden.

Principe 8 — De werkelijke agent moet overeenkomen met zijn manifest

Een manifest beschrijft de agent zoals hij hoort te zijn.

De werkelijkheid moet hiermee voortdurend worden vergeleken.

Daarbij onderscheiden wij drie toestanden:

Declared state

Wat volgens het manifest zou moeten bestaan.

Discovered state

Wat de host en runtime melden dat daadwerkelijk is gedeployed en beschikbaar is.

Observed state

Wat uit telemetry en operationeel gedrag blijkt dat werkelijk gebeurt.

Een afwijking tussen deze toestanden noemen wij **agent drift**.

Voorbeelden van agent drift zijn:

- een ander model gebruiken dan gedeclareerd;
- meer tools bezitten dan toegestaan;
- in een andere regio draaien;
- meer sessies uitvoeren dan het maximum;
- gegevens langer bewaren dan vastgelegd;
- andere agents aansturen dan gedeclareerd;
- zonder de vereiste goedkeuring schrijven naar een extern systeem.

Agent drift moet detecteerbaar, beoordeelbaar en herstelbaar zijn.

Principe 9 — Een agent moet observeerbaar zijn

Een agent die operationeel handelt maar niet observeerbaar is, mag niet als betrouwbare bedrijfscomponent worden beschouwd.

Observability moet inzicht geven in:

- beschikbaarheid;
- actieve en voltooide sessies;
- modelgebruik;
- toolgebruik;
- doorlooptijden;
- fouten;
- kosten;
- delegaties;
- menselijke interventies;
- beleidsafwijkingen;
- beveiligingsincidenten;
- kwaliteit van resultaten.

Observability is niet alleen bedoeld voor technische monitoring. Het ondersteunt ook governance, audit, kwaliteitsverbetering en kostenbeheersing.

Principe 10 — Een agent moet kunnen worden tegengehouden

Iedere agent met operationeel handelingsvermogen moet beschikken over passende interventiemechanismen.

Afhankelijk van het risico omvat dit:

- een kill switch;
- het blokkeren van nieuwe sessies;
- het pauzeren van een actieve taak;
- het intrekken van credentials;
- het uitschakelen van specifieke tools;
- het verlagen van het autonomieniveau;
- het blokkeren van delegatie;
- het terugdraaien van omkeerbare handelingen;
- het isoleren van de runtime.

Een agent die niet kan worden gestopt, is niet bestuurbaar.

Principe 11 — Delegatie draagt verantwoordelijkheid over, maar verwijdert haar niet

Wanneer een agent een taak aan een andere agent overdraagt, moet de delegatie expliciet en traceerbaar zijn.

Een delegatie moet ten minste bevatten:

- de identiteit van de delegerende agent;
- de identiteit van de uitvoerende agent;
- het doel van de delegatie;
- de toegestane scope;
- de relevante context;
- de maximale duur;
- de vereiste terugrapportage;
- de verantwoordelijke menselijke eigenaar.

Een agent mag door delegatie geen bevoegdheden creëren die hij zelf niet bezit.

Een agent mag ook geen taak delegeren om beperkingen, approvals of logging te omzeilen.

Principe 12 — Agents mogen zichzelf niet ongecontroleerd verheffen

Een agent mag niet zelfstandig:

- zijn eigen mandaat verruimen;
- extra credentials aanvragen en activeren;

- een hoger autonomieniveau toekennen;
- governancepolicies uitschakelen;
- zijn accountable owner wijzigen;
- zijn telemetry verwijderen;
- zichzelf onbeperkt kopiëren;
- nieuwe agents met ruimere bevoegdheden creëren.

Aanpassing van constitutionele eigenschappen vereist een gecontroleerd wijzigingsproces.

Een agent mag leren binnen zijn taak, maar mag niet zonder toezicht zijn eigen grondwet herschrijven.

Principe 13 — Geheugen is een bevoegdheid, geen vanzelfsprekendheid

Het vermogen om informatie te onthouden vergroot zowel de bruikbaarheid als het risico van een agent.

Voor ieder geheugentype moet worden vastgelegd:

- welk doel het dient;
- welke informatie mag worden opgeslagen;
- waar de informatie wordt opgeslagen;
- wie toegang heeft;
- hoe lang deze wordt bewaard;
- hoe correctie en verwijdering plaatsvinden;
- of informatie tussen gebruikers, klanten of tenants wordt gedeeld;
- of het geheugen bij beëindiging van de agent wordt verwijderd.

Tijdelijke taakcontext, langetermijngeheugen, organisatorische kennis en persoonlijke voorkeuren moeten als verschillende categorieën worden behandeld.

Principe 14 — Falen moet veilig plaatsvinden

Agents zullen fouten maken.

Een betrouwbare agentarchitectuur probeert niet uitsluitend fouten te voorkomen, maar zorgt ervoor dat fouten:

- vroeg worden gedetecteerd;
- een beperkte impact hebben;
- zichtbaar worden gemaakt;
- niet onbeperkt worden vermenigvuldigd;
- kunnen worden onderzocht;
- waar mogelijk kunnen worden hersteld;

- leiden tot verbetering van systeem en beleid.

Een agent moet bij onzekerheid kunnen vertragen, escaleren, om bevestiging vragen of stoppen.

Doorgaan is niet altijd beter dan niets doen.

Principe 15 — Interfaces en verantwoordelijkheden moeten gescheiden blijven

De identiteit en governance van een agent mogen niet volledig worden bepaald door één technisch protocol.

Het manifesto maakt onderscheid tussen verschillende lagen:

Laag	Functie
Manifesto	Constitutionele beginselen
Agentmanifest	Identiteit, doel, bevoegdheden en gewenste toestand
Hostprotocol	Beheer van hosts, agents en sessies
Capabilityprotocol	Toegang tot tools, data en externe systemen
Telemetryprotocol	Bewijs van gedrag en prestaties
Policy engine	Afdwingen van organisatorische regels
Runtime	Uitvoeren van de agentlogica

Hierdoor kan een organisatie technologie vervangen zonder de identiteit en governance van de agent te verliezen.

Principe 16 — Een agent moet vervangbaar en overdraagbaar zijn

Een agent mag niet zo sterk aan één model, prompt, leverancier of runtime zijn gekoppeld dat zijn organisatorische identiteit verloren gaat wanneer technologie verandert.

De agentidentiteit moet losstaan van:

- het gebruikte foundation model;
- de modelversie;
- de cloudprovider;
- de programmeertaal;
- het agentframework;
- de gebruikte toolprotocollen;
- de specifieke deployment.

Een modelwisseling kan belangrijk genoeg zijn om een nieuwe agentversie te vereisen, maar hoeft niet automatisch een volledig nieuwe agentidentiteit te creëren.

De organisatie moet eigenaar blijven van het mandaat, de geschiedenis en de governance.

---

5. De drie lagen van agentbestuur

5.1 Het manifesto: de grondwet

Het Agent Manifesto beschrijft de beginselen die voor alle agents gelden.

Het beantwoordt vragen als:

- Wat vinden wij een verantwoordelijke agent?
- Welke grenzen mogen niet ongemerkt worden overschreden?
- Welke verantwoordelijkheid blijft bij mensen?
- Welke eigenschappen moeten altijd zichtbaar zijn?

Het manifesto verandert langzaam.

5.2 Het manifest: het paspoort en mandaat

Een agentmanifest beschrijft één concrete agent.

Het bevat onder meer:

```yaml
apiVersion: agents.example.org/v1alpha1
kind: AgentManifest

metadata:
  id: invoice-processing-agent
  name: Invoice Processing Agent
  version: 1.0.0
  status: active

ownership:
  accountableOwner: finance-director
  technicalOwner: agent-platform-team

purpose:
  mission: >
    Inkomende leveranciersfacturen classificeren, controleren
    en voorbereiden voor menselijke goedkeuring.

  allowedOutcomes:
    - invoice_classified
    - supplier_matched
    - booking_proposal_created

  prohibitedOutcomes:
    - payment_executed
    - supplier_created_without_approval

cognition:
  provider: external
  model: approved-reasoning-model
  fallbackPolicy: stop-and-escalate

embodiment:
  hostType: kubernetes
  environment: production
  region: eu-west
  runtimeOwner: agent-platform-team

governance:
  autonomyLevel: supervised
  riskClass: high
  killSwitch: true
```

```
  approvalRequiredFor:
    - external_write
    - financial_commitment
    - supplier_creation

observability:
  logs: required
  traces: required
  metrics: required
  everyToolCallTraced: true
```

Het manifest verandert wanneer de agent zelf verandert.

5.3 De operationele werkelijkheid

De operationele laag omvat:

- deployment;
- runtime;
- sessies;
- modellen;
- tools;
- credentials;
- telemetry;
- policies;
- menselijke interacties;
- andere agents.

Deze laag verandert voortdurend.

Daarom moet de operationele werkelijkheid continu worden vergeleken met het geldende manifest.

---

6. De plaats van AHP, MCP en OpenTelemetry

Het Agent Manifesto schrijft geen specifiek protocol verplicht voor. Wel kunnen bestaande protocollen verschillende rollen vervullen.

Agent Host Protocol

Een hostprotocol zoals het Agent Host Protocol (AHP) kan worden gebruikt voor:

- het ontdekken van agents;

- het publiceren van beschikbare agentbackends;
- het starten en beheren van sessies;
- het volgen van sessiestatus;
- communicatie tussen client, host en agent;
- het aanbieden van runtime-informatie;
- het ontsluiten van telemetrychannels.

Binnen de analogie is AHP het operationele zenuwstelsel tussen de agenthost en de buitenwereld.

Model Context Protocol

Een capabilityprotocol zoals het Model Context Protocol (MCP) kan worden gebruikt voor:

- het aanbieden van tools;
- het ontsluiten van databronnen;
- het lezen van resources;
- het uitvoeren van gecontroleerde acties;
- het standaardiseren van koppelingen met externe systemen.

Binnen de analogie levert MCP de handen, ogen en gereedschappen van de agent.

OpenTelemetry

Een observabilitystandaard zoals OpenTelemetry kan worden gebruikt voor:

- logs;
- traces;
- metrics;
- correlatie van sessies en handelingen;
- onderzoek naar fouten;
- kosten- en prestatieanalyse;
- auditbewijs.

Binnen de analogie vormt telemetry zowel het zenuwstelsel als het medisch dossier van de operationele agent.

Geen van deze protocollen vervangt het agentmanifest. Ze helpen om delen van het manifest technisch waar te maken en te controleren.

---

7. Autonomie- en vertrouwenstrappen

Autonomie moet expliciet worden geclassificeerd.

Niveau 0 — Inert

De agent is geregistreerd, maar voert geen taken uit.

Hij kan bijvoorbeeld als template, ontwerp of gedeactiveerde agent bestaan.

Niveau 1 — Informerend

De agent mag informatie verzamelen, ordenen en presenteren.

Hij verandert geen externe systemen en neemt geen beslissingen namens de organisatie.

Niveau 2 — Adviserend

De agent mag analyses, concepten en aanbevelingen produceren.

Een mens beoordeelt het resultaat voordat het operationele gevolgen krijgt.

Niveau 3 — Assisterend

De agent mag handelingen voorbereiden en beperkte, omkeerbare acties uitvoeren.

Belangrijke wijzigingen vereisen menselijke goedkeuring.

Niveau 4 — Gedelegeerd

De agent mag binnen een nauwkeurig omschreven mandaat zelfstandig acties uitvoeren.

Uitzonderingen, hoge risico's en onzekerheid worden geëscaleerd.

Niveau 5 — Hoog-autonoom

De agent mag gedurende langere tijd zelfstandig doelen nastreven, taken plannen, tools gebruiken en andere agents inschakelen.

Dit niveau vereist de strengste eisen aan observability, isolatie, bevoegdheden, testen en noodinterventie.

Autonomieniveaus zijn geen rangorde van intelligentie of waarde. Zij beschrijven uitsluitend de omvang van de toegestane handelingsruimte.

Een agent mag zichzelf nooit naar een hoger niveau promoveren.

---

8. De samenleving van agents

Wanneer meerdere agents samenwerken, ontstaat meer dan een verzameling losse scripts. Er ontstaat een operationeel netwerk met rollen, afhankelijkheden, gezagsverhoudingen en informatiestromen.

Deze samenleving moet bestuurbaar blijven.

Rollen binnen een agentsamenleving

Een agent kan optreden als:

- uitvoerende agent;
- specialistische agent;
- beoordelende agent;
- controlerende agent;
- coördinerende agent;
- planner;
- router;
- toezichthouder;
- kennisbeheerder;
- menselijke interface.

Een coördinerende agent is niet automatisch de eigenaar van onderliggende agents.

Technische hiërarchie, organisatorische verantwoordelijkheid en bevoegdheidsdelegatie zijn verschillende relaties en moeten afzonderlijk worden vastgelegd.

Sociale stratificatie

Agents kunnen in verschillende vertrouwens- en bevoegdheidsklassen vallen. Deze classificatie mag uitsluitend zijn gebaseerd op:

- risico;
- bewezen betrouwbaarheid;
- scope;
- gevoeligheid van de taak;
- onafhankelijkheid van toezicht;
- impact van fouten.

De classificatie mag niet ontstaan uit schijnbare intelligentie, welsprekendheid of populariteit.

Een welbespraakte agent is niet automatisch een betrouwbare agent.

Machtenscheiding

Bij risicovolle processen behoort niet één agent het volledige proces te beheersen.

Een agent die een betaling voorbereidt, behoort bijvoorbeeld niet zonder aanvullende controle:

- de leverancier aan te maken;
- de bankgegevens te wijzigen;
- de betaling goed te keuren;
- de betaling uit te voeren;

- de auditgegevens te verwijderen.

Agentarchitecturen moeten waar nodig functiescheiding en vierogenprincipes toepassen.

---

9. De levenscyclus van een agent

Iedere agent doorloopt een herkenbare levenscyclus.

Proposed

De agent is ontworpen, maar nog niet operationeel goedgekeurd.

Registered

De agent heeft een identiteit en manifest en is opgenomen in het agentregister.

Tested

De agent is technisch, functioneel en bestuurlijk beoordeeld.

Deployed

De agent is geïnstalleerd op een concrete host.

Active

De agent mag sessies starten en taken uitvoeren.

Restricted

De agent is actief, maar met tijdelijk beperkte bevoegdheden.

Suspended

De agent mag geen nieuwe operationele handelingen uitvoeren.

Deprecated

De agent wordt uitgefaseerd en mag niet meer voor nieuwe toepassingen worden ingezet.

Retired

De agent is beëindigd. Credentials zijn ingetrokken en resterende gegevens zijn volgens beleid gearchiveerd of verwijderd.

Revoked

De agent is wegens risico, misbruik of incident onmiddellijk buiten werking gesteld.

Een gedeactiveerde agent mag niet onzichtbaar blijven doorwerken via achtergebleven tokens, geplande taken, kopieën of gedelegeerde agents.

---

10. Minimale inhoud van een agentmanifest

Een agent die operationeel wordt ingezet, behoort ten minste de volgende gegevens te hebben:

1. **Identiteit** — naam, ID, versie en status.
2. **Eigenaarschap** — accountable owner en technical owner.
3. **Doel** — missie, gewenste uitkomsten en verboden uitkomsten.
4. **Cognitie** — modeltype, provider, fallback en relevante beperkingen.
5. **Embodiment** — host, runtime, regio en beheerpartij.
6. **Bevoegdheden** — tools, resources, scopes en schrijfrechten.
7. **Autonomie** — niveau, approvals en escalatieregels.
8. **Geheugen** — typen, opslag, retentie en verwijdering.
9. **Relaties** — orchestrators, parent agents, subagents en peers.
10. **Observability** — logs, traces, metrics en auditvereisten.
11. **Veiligheidsmechanismen** — kill switch, isolatie en credential revocation.
12. **Levenscyclus** — registratie, wijziging, review en beëindiging.
13. **Compliance** — dataresidentie, privacy, classificatie en relevante regelgeving.
14. **Protocolinterfaces** — management-, capability- en telemetryinterfaces.

---

11. Verboden patronen

De anonieme agent

Een proces voert agentachtige handelingen uit, maar heeft geen herkenbare identiteit, eigenaar of manifest.

De schaduwagent

Een agent wordt buiten het agentregister of de normale governanceprocessen ingezet.

De almachtige agent

Eén agent heeft brede lees-, schrijf-, beheer- en delegatierechten zonder technische begrenzing.

De verborgen modelwissel

Het gebruikte model verandert zonder versiebeheer, beoordeling of zichtbaarheid in het manifest.

De oneindige sessie

Een agent blijft onbeperkt actief, bouwt steeds meer context op en heeft geen duidelijk eindpunt of herstartbeleid.

De onbegrensde herinnering

De agent bewaart gegevens zonder doel, retentiebeleid of mogelijkheid tot verwijdering.

De ontraceerbare delegatie

Een agent besteedt werk uit zonder vast te leggen aan wie, waarom en onder welk mandaat.

De papieren human-in-the-loop

Een menselijke goedkeuring bestaat formeel, maar de mens beschikt niet over tijd, context of werkelijke mogelijkheid om te weigeren.

Responsibility laundering

Mensen of organisaties gebruiken de agent om verantwoordelijkheid voor besluiten te verhullen of af te schuiven.

De onsterfelijke agent

Een agent kan niet volledig worden beëindigd doordat credentials, kopieën, geplande processen of externe memories blijven bestaan.

De zelfuitbreidende agent

Een agent kan ongecontroleerd nieuwe agents, tools, accounts of infrastructuur creëren.

De stille drift

De operationele agent wijkt af van zijn manifest zonder dat dit wordt gedetecteerd of gerapporteerd.

---

12. Governance als continue praktijk

Agentgovernance is geen eenmalige goedkeuring vooraf.

Een agent verandert doordat:

- modellen worden bijgewerkt;
- prompts worden aangepast;
- tools nieuwe functies krijgen;
- databronnen veranderen;
- medewerkers nieuwe opdrachten geven;
- andere agents worden toegevoegd;
- de omgeving verandert;
- onverwacht gedrag zichtbaar wordt.

Daarom vereist betrouwbare agentgovernance:

- periodieke herbeoordeling;
- versievergelijking;
- toegangsreviews;
- evaluatie van modelgedrag;
- controle op agent drift;
- beoordeling van incidenten;
- herziening van autonomieniveaus;
- controle op kosten en effectiviteit;
- beëindiging van agents die geen aantoonbare waarde meer leveren.

Een agentmanifest is dus geen statisch formulier. Het is een levend contract tussen de organisatie en de operationele agent.

---

13. De Agent Bill of Responsibilities

Een agent bezit geen menselijke rechten. Toch kan zijn positie worden beschreven in termen van operationele verantwoordelijkheden.

Een goed ontworpen agent behoort:

1. zijn identiteit niet te verbergen;
2. binnen zijn mandaat te handelen;
3. onzekerheid zichtbaar te maken;
4. geen bevoegdheden te veinzen;
5. menselijke instructies te toetsen aan geldende policies;
6. relevante handelingen te laten registreren;
7. delegaties herkenbaar te maken;
8. gevoelige gegevens doelgebonden te behandelen;
9. bij conflicterende opdrachten te escaleren;
10. bij onvoldoende zekerheid te vertragen of stoppen;
11. interventie en beëindiging te accepteren;
12. zijn eigen beperkingen niet zelfstandig te verwijderen.

Deze verantwoordelijkheden moeten niet alleen in een prompt worden beschreven. Waar mogelijk moeten zij technisch en organisatorisch worden afgedwongen.

---

14. Onze ontwerpbelofte

Wij zullen agents niet beoordelen op hoe menselijk zij klinken, maar op hoe betrouwbaar zij functioneren.

Wij zullen geen autonomie toekennen zonder een expliciet mandaat.

Wij zullen technische capaciteit niet verwarren met organisatorische bevoegdheid.

Wij zullen agentidentiteit loskoppelen van model en leverancier.

Wij zullen iedere agent verbinden aan menselijk eigenaarschap.

Wij zullen handelingen observeerbaar en bevoegdheden herroepbaar maken.

Wij zullen samenwerking tussen agents behandelen als een bestuurlijke architectuur, niet alleen als een technisch experiment.

Wij zullen afwijkingen tussen gedeclareerde en werkelijke toestand zichtbaar maken.

Wij zullen systemen ontwerpen waarin een agent veilig kan falen, kan worden gecorrigeerd en volledig kan worden beëindigd.

Wij zullen technologie niet gebruiken om menselijke verantwoordelijkheid te verbergen.

---

15. Slotverklaring

Agents kunnen organisaties helpen sneller te leren, consistenter te handelen en complexe werkzaamheden op grotere schaal uit te voeren.

Maar intelligent gedrag alleen maakt een agent nog niet betrouwbaar.

Betrouwbaarheid vereist identiteit. Identiteit vereist eigenaarschap. Autonomie vereist grenzen. Samenwerking vereist governance. Vertrouwen vereist bewijs. Macht vereist interventie. En iedere operationele agent vereist een plek binnen een menselijke orde.

Daarom verklaren wij:

> **Een agent is nooit slechts een model. Een agent is een gedeclareerde, begrensde en observeerbare actor binnen een systeem van menselijke verantwoordelijkheid.**

En:

> **Geen agency zonder identiteit. Geen autonomie zonder grenzen. Geen handeling zonder verantwoordelijkheid.**
