

# JDriven Tech Radar Voorjaar 2026

*Onze kijk op recente ontwikkelingen in het veld*

13e editie

**Commit to know.  
Develop to grow.  
Share to show.**

## Voorwoord

Met veel trots presenteren we de 13e editie van de JDriven Tech Radar.

Zoals menigeen zal hebben opgemerkt op conferenties en in de media zijn er twee onderwerpen die de gemoederen bezighouden in ons vakgebied: AI en digitale soevereiniteit in Europa. In zo'n situatie is het verleidelijk om mee te gaan in het heersende sentiment en kritiek klakkeloos over te nemen of te verwerpen. De opzet van de totstandkoming van de JDriven Tech Radar beschermt ons hiervoor. Allereerst moeten we voor zowel een onderwerp als de mening over dat onderwerp naar elkaar toe onderbouwen of het op de radar moet komen. Is dat eenmaal besloten, dan is er een stijlgids die we gebruiken voor het uitschrijven van de blip-tekst, waarbij we eerst objectief beschrijven wat iets is en daarna subjectief onze mening erover geven. Dit onderscheid helpt meerdere perspectieven in acht te nemen en feiten en meningen uit elkaar te houden. Ten slotte worden aangedragen teksten behandeld zoals een merge request: collega's reviewen elkaars bijdragen. Op deze manier kunnen we trends volgen en genuanceerd en goed onderbouwd duiden.

Volgens mij is dat ook deze keer gelukt.

Bedankt aan eenieder die deze editie mogelijk heeft gemaakt!



Jasper Bogers  
Consultant JDriven

## Introductie

Onze ervaren specialisten werken dagelijks mee aan tal van softwareprojecten in Nederland en zijn betrokken in wereldwijde community's. Halfjaarlijks komen wij vanuit JDriven bij elkaar om te bespreken wat wij aan nieuwe trends en ontwikkelingen zien. Deze trends proberen wij te vangen in een technologieradar. Elke editie van de radar laat verschuivingen zien t.o.v. een vorige editie. Een verschuiving kan betekenen dat wij een technologie interessanter zien worden waar van toepassing, of juist minder geschikt ongeacht de toepassing. Indien een trend niet meer voorkomt in een latere editie, dan zijn er geen nieuwe ontwikkelingen en/of ervaringen die ons eerdere beeld zouden hebben bijgesteld. In dit document willen we toelichten welke verschuivingen we de afgelopen periode hebben waargenomen, om op basis daarvan weer richting te geven aan wat wij inzetten en aanraden.

## Tech Radar

Het idee voor het opstellen van een Tech Radar komt voort vanuit Thoughtworks. Zij dragen al langer periodiek met een radar hun visie uit op nieuwe trends en ontwikkelingen. Bovendien raden zij iedereen aan [een eigen radar op te stellen](#).

Bij JDriven onderschrijven we dat. Het opstellen van een Radar is een leerzame en waardevolle ervaring waarin onderling kennis wordt gedeeld en een technisch bewustzijn wordt gecreëerd. Wij geloven erin dat specialisten zelf in staat moeten zijn om het gereedschap voor hun werkzaamheden samen te stellen. Met het opstellen van een radar faciliteer je discussies over technologie, om als organisatie de juiste balans te vinden in wat voor risico's en voordelen innovatie kan opleveren. Wij kunnen je helpen dit op te starten binnen je organisatie. Laat je teams elkaar inspireren tot innovatie en gezamenlijk komen tot een set aan technologieën en technieken die de ontwikkeling in je bedrijf versnellen.

## Indeling

Een radar bestaat uit kwadranten en ringen, met daarbinnen blips om interessante technologieën en technieken aan te duiden.

De kwadranten verdelen de verschillende onderwerpen in categorieën.

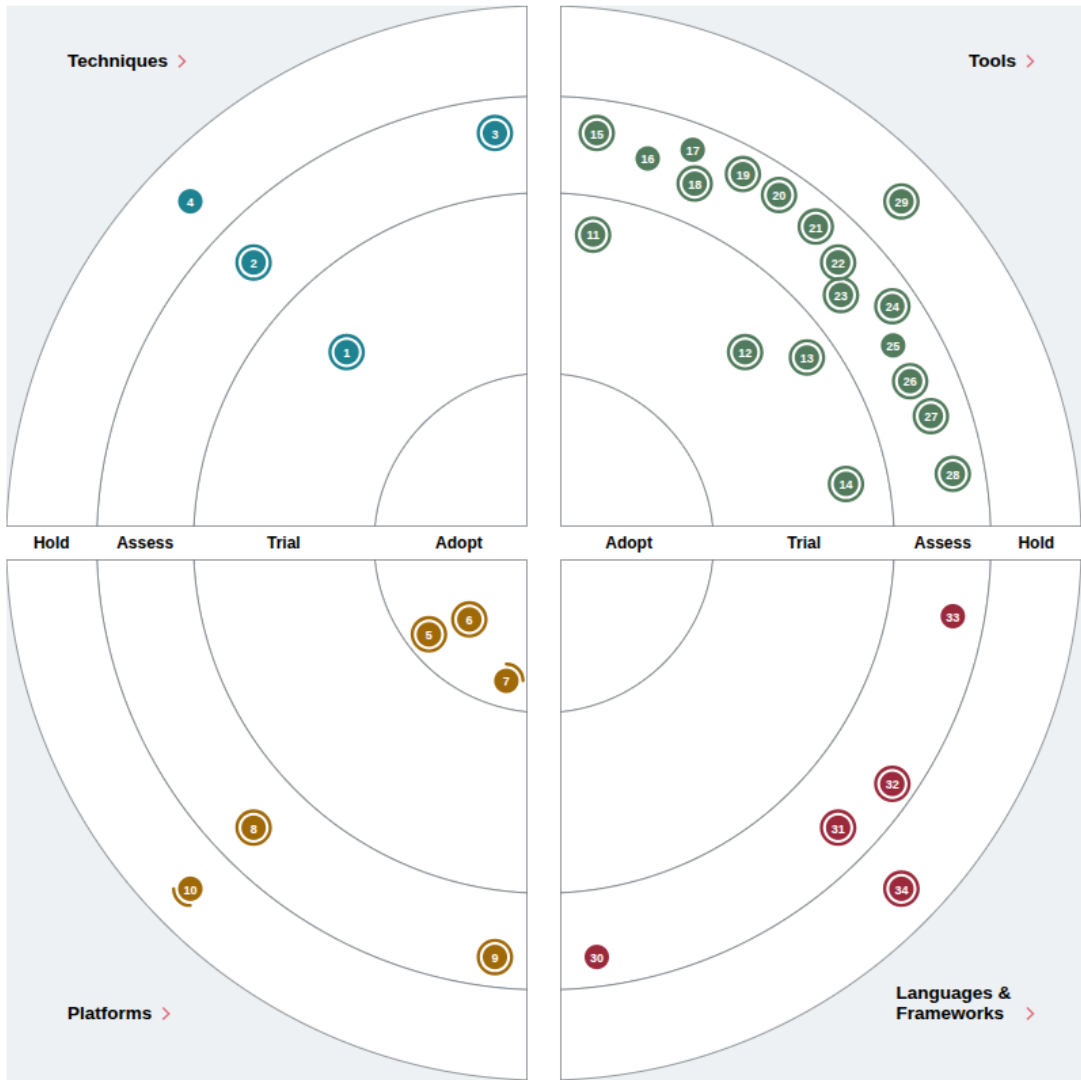
- **Technieken** die je helpen betere software te maken
- **Platformen** waarop je software kunt uitvoeren
- **Tools** ter ondersteuning van je ontwikkel- en delivery-proces
- **Talen & frameworks** die je ondersteunen bij de ontwikkeling van software

De ringen in elk kwadrant geven aan in welk stadium van adoptie wij denken dat die technologie zich bevindt.

- **Adopt** → Wij raden sterk aan deze technologie te gebruiken, waar van toepassing.
- **Trial** → Interessant om alvast ervaring mee op te doen (in een project dat het risico kan dragen).
- **Assess** → Goed om beter te begrijpen en toekomstige impact in te schatten, maar nog niet om toe te passen.
- **Hold** → Niet (meer) gebruiken.

In de volgende secties zullen we onze kijk op de recente ontwikkelingen per kwadrant toelichten.



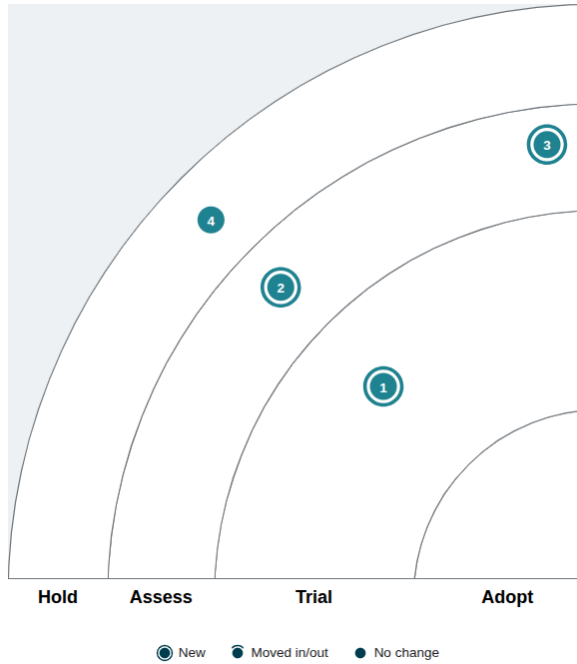


● New   ● Moved in/out   ● No change

Technieken	Tools
<b>ADOPT</b>	<b>ADOPT</b>
-	-
<b>TRIAL</b>	<b>TRIAL</b>
1. Open Space Technology	11. Agents.md
<b>ASSESS</b>	12. IntelliJ Spring Debugger
2. AI Reasoning	13. Mattermost
3. Dynamic Consistency Boundary (DCB)	14. NullAway
<b>HOLD</b>	<b>ASSESS</b>
4. Vibe coding	15. Agent skills
	16. Anubis
	17. BlueDolphin
	18. Danger
	19. Devbox
	20. Digital Sovereignty Readiness Assessment
	21. Jujutsu
	22. Kepler
	23. Kro
	24. Matrix
	25. Maven 4
	26. OpenCode.AI
	27. Pulp
	28. Pulumi
	<b>HOLD</b>
	29. OpenClaw
<b>Platforms</b>	<b>Talen &amp; frameworks</b>
<b>ADOPT</b>	<b>ADOPT</b>
5. Docker Hardened Images	-
6. European Clouds	<b>TRIAL</b>
7. Proxmox	-
<b>TRIAL</b>	<b>ASSESS</b>
-	30. Amber
<b>ASSESS</b>	31. Arconia
8. Codeberg	32. Embabel Agent Framework
9. Pigsty minio fork	33. Koog
<b>HOLD</b>	<b>HOLD</b>
10. AWS European Sovereign Cloud	34. Scala



# Technieken



## TRIAL

1. Open Space Technology

## ASSESS

2. AI Reasoning

3. Dynamic Consistency Boundary (DCB)

## HOLD

4. Vibe coding

## AI Reasoning

### ASSESS

AI Reasoning is een evolutie van onze eerdere blips over AI Agents, specifiek gericht op het gebruik van AI in businessprocessen en niet in het softwareontwikkelingsproces zelf. Waar "AI Agents" doorgaans verwijst naar autonome systemen die zelfstandig beslissingen nemen, legt AI Reasoning de nadruk op ondersteuning: AI die helpt redeneren, maar waarbij de mens de regie houdt.

Het kernonderscheid is *AI-assisted reasoning* versus *AI-based decisions*. AI neemt specifiek werk over: analyseren, samenvatten, opties afwegen, verbanden leggen. De verantwoordelijkheid voor de uiteindelijke beslissing blijft menselijk. Dat onderscheid is ethisch relatief onbetwist en maakt AI Reasoning praktisch toepasbaar waar AI Agents nog vragen oproepen.

Een verzekeraar die AI inzet om schadeclaims te beoordelen en een advies te formuleren, maar waarbij een medewerker de uiteindelijke beslissing neemt, is een typisch voorbeeld van AI Reasoning in de praktijk.

Toch gelden ook hier de nodige kanttekeningen: requirements nauwkeurig formuleren, output controleren en tussenstappen corrigeren kost tijd, soms meer dan het traditioneel implementeren van business rules. In de juiste situaties is het potentieel echter groot genoeg om actief mee te experimenteren.

# Dynamic Consistency Boundary (DCB)

## ASSESS

**Dynamic Consistency Boundary (DCB)** is een aanpak voor het handhaven van consistentie in event-sourced systemen. In klassiek DDD bewaakt een **aggregate** de regels die gelden voor data (entiteiten met business rules) binnen een logische grens (je consistency boundary). Zodra je architectuur een regel moet implementeren die logischerwijs meerdere aggregates moet bevragen of beïnvloeden, heb je een **saga** of **process manager** nodig om de coördinatie te regelen. En je bent aangewezen op het gebruiken van corrigerende events wanneer de transactie in één aggregate slaagt maar in de andere faalt. Je kunt overwegen je aggregates anders te modelleren, met het risico dat de modellen gekunsteld overkomen of elders gaan wringen. Bovendien is het hermodelleren van je aggregates nadat je systeem data heeft verwerkt geen eenvoudige exercitie.

Met DCB stap je af van aggregates als statische afbakening en wordt de saga overbodig. In plaats daarvan kijk je per uit te voeren opdracht (command) naar alle events in een event store die tot de huidige staat van het systeem hebben geleid en creëer je een tijdelijke consistentie-"bubbel". Dit is mogelijk doordat je bij DCB events van één of meerdere tags voorziet. Na het interpreteren van de huidige staat en de opdracht schrijf je de nieuwe events weg, waarbij je een voorwaarde (**append condition**) kunt opgeven waarmee je controleert of er niet in de tussentijd conflicterende events zijn weggeschreven. Hiermee bewaakt het systeem zijn consistentie.

Waar je een systeem met aggregates ook zonder event sourcing kunt werken, werk je met DCB met event sourcing zonder aggregates. Er zijn al frameworks en eventstores die het ondersteunen, maar het is een fundamenteel andere benadering van DDD die een leercurve kent. Binnen de DDD-gemeenschap wordt voorzichtig positief gereageerd op het verlaten van het beproefde, doch moeilijke concept van aggregates te verruilen voor DCB. Een overstap zal niet eenvoudig zijn maar de voordelen zijn veelbelovend en de moeite waard om te verkennen.

# Open Space Technology

## TRIAL

**Open Space Technology (OST)** is een interactieve en ongedwongen vergadervorm die experts in staat stelt om kennis te delen waarbij de agenda ter plekke wordt opgesteld. Deelnemers zijn vrij om te kiezen welk onderwerp voor hen interessant is om bij te wonen en kunnen gedurende de sessie ook elders aanschuiven. Hierdoor faciliteert het de mogelijkheid om vrij onderwerpen te bespreken waarbij je mensen bij elkaar brengt voor constructieve en positieve discussies.

Hoewel Open Space Technology als format al langer bestaat en ook beschreven staat in "**Liberating Structures**", een collectie aan technieken om samenwerking en innovatie te bevorderen, zien we dit de laatste tijd meer gebruikt worden bij congressen en binnen organisaties. Onze ervaringen zijn hier positief over en we hebben dit dan ook op *Trial* gezet om anderen eens aan te raden dit te proberen.



## Vibe coding

### HOLD

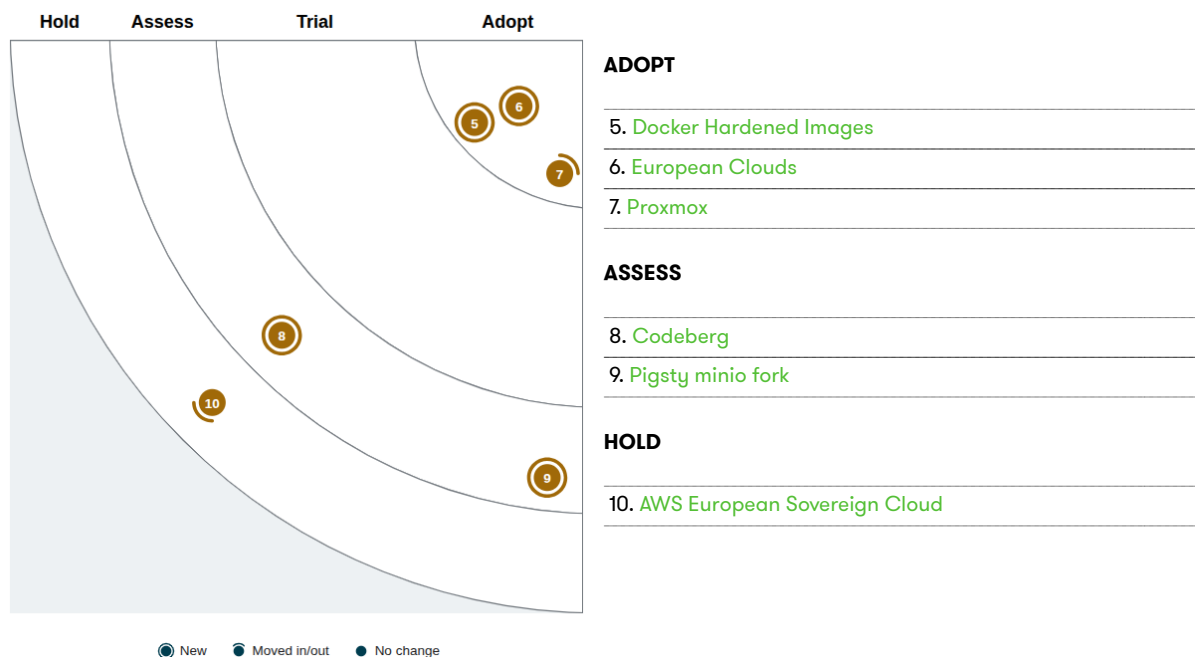
Vibe coding is een term die recent een opmars heeft gemaakt in de softwareontwikkeling. Het verwijst naar een ontwikkelmethodiek waarbij programmeurs primair vertrouwen op intuïtie en gevoel bij het iteratief gebruik van AI-tools, met name large language models (LLM's). In plaats van gestructureerde ontwikkelprocessen, draait het om het herhaaldelijk aanpassen van prompts en experimenteren met AI-gegenereerde code totdat het resultaat "goed voelt" en/of lijkt te functioneren.

De praktijk kenmerkt zich door een cyclus van prompt-engineering, waarbij ontwikkelaars instructies verfijnen op basis van de output, zonder noodzakelijkerwijs diepgaand begrip van de onderliggende code, de taal, het framework of de interactie met de infrastructuur waarop het draait. Volgens voorstanders draagt dit bij aan snellere prototyping, terwijl critici wijzen op risico's rondom codekwaliteit, performance, onderhoudbaarheid en begrip van de software.

JDriven schaart zich achter de critici en plaatst vibe coding op *Hold*. Het betreft een trial-and-error benadering die haaks staat op professionele softwareontwikkeling, waarbij gestructureerd werken, clean code en kwaliteitscontrole centraal staan. Dit gebrek aan systematiek leidt tot "technical debt", beveiligingsproblemen en slecht onderhoudbare codebases. Het feit is dat software development meer behelst dan 'code kloppen' om tot een degelijk resultaat te komen, waar vibe coding geen oplossing voor biedt.

Daartegenover staat dat bewust gebruik van LLM's in een professionele ontwikkelomgeving waarde kan toevoegen. Als onderdeel van een gestructureerd ontwikkelproces kunnen AI-tools uitstekend functioneren voor code review, documentatie, het genereren van boilerplate-code, het verkennen van design patterns en het bieden van enige houvast bij technische foutmeldingen. Het verschil ligt in verantwoorde, methodische integratie versus het ad hoc experimenteren dat kenmerkend is voor vibe coding.

# Platforms



## Docker Hardened Images

### ADOPT

Containerisatie, ofwel OS-virtualisatie, is voor veel developers de standaard manier om software te bouwen, testen en deployen. Het is daarom belangrijk dat de containers waar hun software op draait veilig zijn. Daartoe worden container images **gehard**, wat wil zeggen dat er extra veiligheidsmaatregelen worden toegevoegd om het aantal kwetsbaarheden te minimaliseren. Voorheen was dit een taak die developers zelf moesten uitvoeren, maar Docker biedt sinds december 2025 gratis 'geharde' versies van bepaalde images aan, zogenoemde **Docker Hardened Images (DHI)**. Voor JVM-ontwikkelaars is er bijvoorbeeld een hardened versie van de populaire Eclipse Temurin container image (**Eclipse Temurin DHI**).

We zijn enthousiast over Docker Hardened Images, omdat het een eenvoudige manier biedt om de veiligheid van container images te verbeteren zonder dat developers zelf aan de slag hoeven met hardening. DHI's zijn drop-in replacements voor reguliere images, wat betekent dat je ze gemakkelijk kunt gebruiken in plaats van de standaardversies. Bovendien worden CVE's proactief gepatcht in DHI's, waardoor je minder zorgen hebt over kwetsbaarheden in je container images. Daarom plaatsen we DHI in *Adopt* en stellen we voor bij beschikbaarheid altijd de geharde versie te verkiezen.



## European Clouds

### ADOPT

De EU stelt via General Data Protection Regulation (GDPR) eisen aan bedrijven die data, gerelateerd aan inwoners van de EU, verwerken en verzamelen. De Amerikaanse Cloud Act, wet FISA702 en Executive Order 12.333 stellen de Amerikaanse overheid echter in staat een bedrijf met Amerikaans hoofdkantoor te dwingen om de GDPR te negeren. Ze kunnen inzage in data eisen, zelfs als de betreffende servers buiten de VS liggen. Deze bedrijven zijn niet verplicht hun klanten daarover te informeren. Ook kan de dienstverlening zelf aan banden gelegd worden, wat een reëel risico is, zoals het Internationaal Strafhof ondervond toen de Amerikaanse politiek Microsoft opdroeg hun e-mail te blokkeren. Meer en meer bedrijven zijn daarom op zoek naar Europese alternatieven voor hun data en diensten. Op [european-alternatives.eu](https://european-alternatives.eu) is een groot aantal genoemd. Initiatieven om dergelijke beweging in gang te zetten, worden gevat binnen het concept digitale soevereiniteit.

Vanuit het perspectief van developers en operations engineers kijkt JDriven met name naar cloud providers. Bij het kiezen van een Europese cloud is het van belang te letten op een aantal aspecten:

- Wat biedt het aan compute-, networking- en storage-services? Veel van deze providers bieden niet het volledige serviceaanbod van de grootste Amerikaanse hyperscalers. Maar dat is minder relevant dan de vraag of het serviceaanbod voldoende is voor de actuele behoefte.
- Wat zijn de Service Level Agreements (SLA) voor beschikbaarheid?
- Wat zijn de prijzen? Veel providers bieden een prijscalculator om snel inzicht te kunnen krijgen en een vergelijking te maken. Afhankelijk van welke services gebruikt worden, kunnen verschillen per maand hier behoorlijk zijn.
- Is er een free tier? Een free tier kan enorm helpen met het opdoen van ervaring en dientengevolge ook met het vinden van ervaren developers.
- Biedt het infrastructure as code en een SDK om de services reproduceerbaar op te tuigen? Bestaan er bijvoorbeeld Terraform-providers voor?
- Hoe toegankelijk zijn de leermiddelen voor developers?
- Biedt het een vorm van managed Kubernetes? Dit is het containerplatform waar meer en meer klanten hun software op uitrollen, wat van groot belang is bij een exitstrategie waar een migratie naar een andere provider bij komt kijken.
- Ondersteunt het Gaia-X? Gaia-X is een project voor federatie van digitale ecosystemen waarbij data-soevereiniteit, veiligheid en naleving van wetgeving centraal staan. Een dergelijk streven naar samenwerking en integratie is een aspect van Europese cloud providers dat we bij de gevestigde Amerikaanse providers niet zien.

Nemen we bovenstaande in overweging, dan zien we een snelle groei naar volwassenheid bij met name Scaleway, STACKIT en OVHcloud. Ook Exoscale, IONOS en het Nederlandse Cyso Cloud zijn interessant.

## Proxmox

### ADOPT

Proxmox biedt een alternatief voor het veelgebruikte VMware-platform.

Proxmox Virtual Environment is een compleet, opensource serverbeheerplatform voor enterprise-virtualisatie. Het integreert de KVM-hypervisor en Linux Containers (LXC), software-gedefinieerde opslag en netwerkfunctionaliteit, in een enkel platform. Met de geïntegreerde webgebaseerde gebruikersinterface beheer je makkelijk de VMs en containers, hoge beschikbaarheid voor clusters, of geïntegreerde noodherstelgereedenschappen.

Proxmox Backup Server is een bedrijfskritische back-upoplossing, voor het opslaan en herstellen van reservekopieën van VMs, containers en fysieke hosts. Door ondersteuning van incrementele, volledig ontduubelde reservekopieën, reduceert Proxmox Backup Server drastisch de netwerkbelasting en bespaart het kostbare opslagruimte. Met sterke versleuteling en methoden voor het verzekeren van gegevensintegriteit kun je je veilig voelen tijdens het maken van reservekopieën van gegevens, zelfs naar doelen die niet volledig vertrouwd zijn.

Proxmox Datacenter Manager is een gecentraliseerde beheeroplossing voor het beheren en overzien van meerdere nodes en clusters met Proxmox-gebaseerde virtuele omgevingen. Ontworpen voor grote bedrijfsmatige omgevingen, biedt het IT-teams een compleet overzicht van hun gedistribueerde Proxmox-virtualisatie-infrastructuur.

Ondanks dat het geen nieuwe techniek is, is het in de context van groeiende onzekerheid rondom internationale clouddiensten belangrijk om het zelf hosten en in beheer nemen van IT-diensten op eigen servers opnieuw te overwegen. Wij zien Proxmox als een interessantere optie dan VMware, dat sinds de overname door Broadcom almaar stijgende licentiekosten kent. We hebben Proxmox naar Adopt verhuisd omdat we zien dat het breed wordt omarmd in de markt.

## Codeberg

### ASSESS

Als je graag een Europees alternatief voor GitLab of GitHub als clouddienst zoekt, is Codeberg een interessant alternatief om te overwegen. Het biedt ongeveer dezelfde functionaliteiten, zij het minder uitgebreid, zoals: code storage, static site hosting, pull requests met daarbij reviewmogelijkheden, pipelines, etc. Meertaligheid wordt ondersteund via Weblate. Het is gebaseerd op het Forgejo (Gitea fork) codeplatform.

Codeberg is een in Duitsland gevestigde non-profitorganisatie. Iedereen kan gratis een account aanmaken, code uploaden en samenwerken. Als je de continuïteit van Codeberg wilt versterken is het ook mogelijk om lid te worden (je betaalt daarvoor een kleine contributie) en/of eenmalige of structurele donaties te doen. Alhoewel het hoge beschikbaarheid nastreeft kan het niet dezelfde formele uptime Service Level Agreements (SLA's) garanderen als commerciële platformen.



## Pigsty minio fork

### ASSESS

MinIO was een populair opensource product om on-premise S3 buckets te hosten zonder AWS. Maar sinds enkele jaren heeft MinIO Inc de licentievoorwaarden strenger gemaakt en sinds 2026-02-13 is de MinIO repository gearchiveerd. MinIO Inc doet zelf nog wel onderhoud, maar maakt die code niet meer open source.

Organisaties die MinIO gebruiken maar niet afhankelijk willen zijn van een betaalde versie van MinIO Inc kunnen nu overwegen om uit te wijken naar de [Pigsty MinIO fork](#). Die zou als een drop-in replacement gebruikt moeten kunnen worden. Images zijn via Docker Hub beschikbaar; binaries via APT en Yum.

Een ander alternatief is [Ceph](#). Dat is open source en zonder licentiekosten te gebruiken. Voor de applicaties die gebruik maken van MinIO is dit ook een drop-in replacement. Maar voor de beheerders niet, aangezien Ceph een complexer product is. Omdat Ceph onder de vlag van de Linux Foundation valt, is de kans dat het "opeens" closed-source wordt (zoals bij MinIO of Redis gebeurde) nagenoeg nul. De governance-structuur is erop ingericht dat de community en de industrie samen eigenaar zijn. Nog een open source alternatief is [SeaweedFS](#). Minder complex dan Ceph, maar niet zo eenvoudig als MinIO.

## AWS European Sovereign Cloud

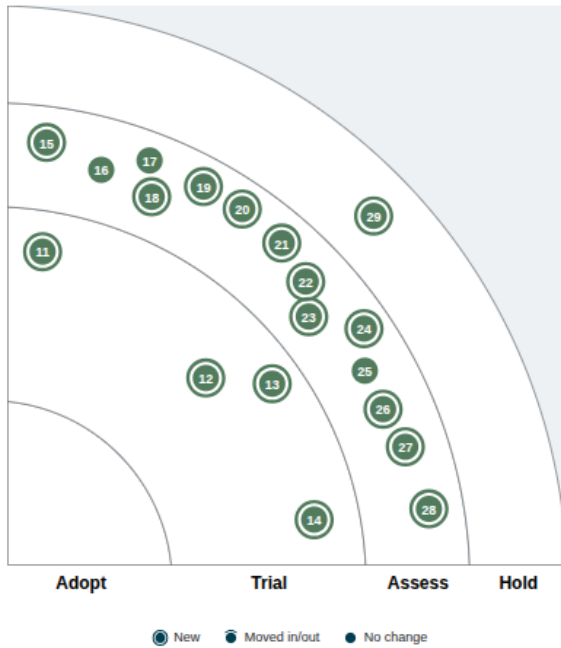
### HOLD

Om zorgen van Europese bedrijven rondom P-CLOB en GDPR te adresseren, heeft AWS de [AWS European Sovereign Cloud](#) aangekondigd, wat een AWS Cloud is die geheel in de EU gehuisvest moet zijn, bemand en bediend door inwoners van de EU. AWS European Sovereign Cloud is in Duitsland gevestigd onder een eigen Duitse bedrijfsentiteit, Amazon Web Services EMEA SARL. AWS is transparant omtrent het willen naleven van de GDPR en over de impact van FISA702 en benadrukt dat ieder verzoek van een overheid om data in te zien kritisch bekeken zal worden. Het biedt in eerste instantie minder services dan de Amerikaanse AWS Cloud, omdat het niet alleen organisatorisch maar ook technisch en fysiek gescheiden wil kunnen opereren van het Amerikaanse moederbedrijf.

Voor een bedrijf dat voor zijn software hevig heeft geïnvesteerd in AWS-kennis en -implementatie en dat op zoek is naar een manier om GDPR-compliant te blijven zonder die investeringen af te schrijven, lijkt European Sovereign Cloud wellicht een veilig alternatief. Bedrijven die de organisatorische opzet vanuit een wettelijk standpunt hebben bekeken, waarschuwen echter dat het Europese dochterbedrijf uiteindelijk nog altijd onder het Amerikaanse Amazon.com Inc. en de daarbij behorende wetgeving valt. Of het moederbedrijf bij een eis van de Amerikaanse rechtbank via de Duitse rechtbank zal gaan om iets bij de eigen dochteronderneming af te dwingen en hoe het zich er wettelijk uit zal gaan redden als het daar bakzeil haalt, is onduidelijk. Het is een risico dat kleiner is dan wanneer je data en services afneemt van de Amerikaanse AWS-cloud, maar het is een risico dat afgewogen moet worden: zijn de kosten van een migratie van AWS US naar AWS EU, met bovendien een kleiner serviceaanbod, dan te verantwoorden als soevereiniteit een argument is, of is een keuze voor een andere Europese cloudprovider met voldoende technisch aanbod het overwegen waard?

Dit alles maakt dat we dit van *Assess* naar *Hold* verschuiven. Daarbij is het het vermelden waard dat, gegeven de soevereiniteitskwestie, je *capabiliteit* om te kunnen migreren tussen cloudproviders sowieso aandacht verdient.

# Tools



## TRIAL

- 11. [Agents.md](#)
- 12. [IntelliJ Spring Debugger](#)
- 13. [Mattermost](#)
- 14. [NullAway](#)

## ASSESS

- 15. [Agent skills](#)
- 16. [Anubis](#)
- 17. [BlueDolphin](#)
- 18. [Danger](#)
- 19. [Devbox](#)
- 20. [Digital Sovereignty Readiness Assessment](#)
- 21. [Jujutsu](#)
- 22. [Kepler](#)
- 23. [Kro](#)
- 24. [Matrix](#)
- 25. [Maven 4](#)
- 26. [OpenCode.AI](#)
- 27. [Pulp](#)
- 28. [Pulumi](#)

## HOLD

- 29. [OpenClaw](#)

## NullAway

### TRIAL

[NullAway](#) is een statische analysetool voor Java die op basis van nullability-annotaties mogelijke `NullPointerException`'s tijdens de build detecteert.

Het draait als een plugin op ErrorProne, wat een compiler plugin is voor Java die tijdens de build statische codechecks uitvoert. NullAway geeft daardoor snelle feedback in de normale build. Recente versies ondersteunen ook JSpecify en vereisen JDK 17 of hoger.

Wij plaatsen NullAway in *Trial*. Het levert snel waarde op in Java codebases. Invoering vraagt wel een consistente annotatiestrategie op basis van JSpecify en integratie met ErrorProne.



## Agents.md

### TRIAL

**Agents.md** is een open standaard voor het meegeven van projectspecifieke instructies aan AI coding agents. Het is een Markdown-bestand dat je in de root van je repository plaatst, vergelijkbaar met een README.md maar dan gericht op AI-assistenten in plaats van menselijke developers. In het bestand leg je vast hoe het project gebouwd en getest moet worden, welke code-conventies er gelden en welke architectuurpatronen gevolgd moeten worden.

Het initiatief is gestart door Sourcegraph en valt nu onder de Agentic AI Foundation binnen de Linux Foundation. AGENTS.md is tool-agnostisch en wordt ondersteund door meerdere AI coding tools zoals Kiro, Cursor, GitHub Copilot en OpenAI Codex. Dit maakt het een aantrekkelijk alternatief voor tool-specifieke bestanden zoals .cursorsules of CLAUDE.md.

Vanuit JDriven zien wij dat een AGENTS.md de kwaliteit van AI-gegenereerde code merkbaar verbetert, doordat de AI beter begrijpt hoe het project in elkaar zit en welke patronen het team hanteert. Een bijkomend voordeel is dat dit ook voor nieuwe teamleden waardevolle documentatie oplevert. Het opstellen van een goed AGENTS.md-bestand vergt wel een initiële investering en het vinden van de juiste balans: je moet bewust nadenken over je conventies en deze helder formuleren. Het open karakter van de standaard, alsook het feit dat tienduizenden repositories op GitHub het al toepassen, geven ons voldoende vertrouwen om AGENTS.md op *Trial* te plaatsen.

## OpenCode.AI

### ASSESS

**OpenCode.ai** is een opensource AI-coding assistant die onafhankelijk werkt van providers en modellen. De tool kan worden gebruikt als command-line interface, desktop-applicatie en IDE-extensie. OpenCode.ai draait lokaal en stelt je in staat om meerdere agents te definiëren, bijvoorbeeld voor Planning, Building of Reviewing. De tool begrijpt je code met behulp van language server protocols (LSP). Je kunt skills of MCP-based tools integreren met een expliciet permission model. Developers kunnen modellen gebruiken van 75+ LLM providers, bijvoorbeeld OpenAI, Anthropic, Google of zelfs lokale modellen.

OpenCode.ai bevordert consistent Agentic AI gebruik binnen teams door de configuratie te delen in je team. Code en context blijven lokaal. Configuratie en governance blijven onder eigen controle via version control en lokale environment variabelen.

JDriven ziet dat OpenCode.ai volwassen genoeg is voor pilots, maar de sprong naar brede toepassing vraagt eerst validatie en feedback op je eigen workflows. De tool moet zich nog bewijzen in diverse team- en projectcontexten. We plaatsen OpenCode.ai daarom in *Trial*: geschikt voor experimentatie en workflow standaardisatie nu, maar nog te nieuw om organisatiebreed in te zetten als standaard development tooling.

## Agent skills

### ASSESS

**Agent Skills** is een open standaard voor het verpakken van herbruikbare AI-agentcapabiliteiten als modulaire eenheden, oorspronkelijk ontwikkeld door Anthropic en in december 2025 vrijgegeven als open specificatie. Het stelt developers in staat om domeinkennis, instructies, scripts en referentiemateriaal te bundelen in een mapstructuur met een `SKILL.md`-bestand, dat agents on-demand kunnen ontdekken en laden. De standaard wordt inmiddels breed ondersteund door tools als Claude Code, VS Code, GitHub Copilot, OpenAI Codex, Cursor en Gemini CLI, en is ook geïntegreerd in het Spring AI-ecosysteem via de `spring-ai-agent-utils` toolkit.

Vanuit JDriven zien wij Agent Skills als een interessant stuk gereedschap voor teams die werken met AI-agents in hun ontwikkelproces. De kracht zit in de eenvoud: een skill is niet meer dan een map met Markdown en optioneel scripts, waardoor het laagdrempelig is om in gebruik te nemen en in versiebeheer te plaatsen bij je applicatiecode. Voor Java-teams is de integratie via Spring AI bijzonder interessant: skills werken modelonafhankelijk en vereisen geen architectuurwijzigingen in bestaande applicaties.

De standaard is echter jong en evolueert snel. Er is nog geen ingebouwd versiebeheermechanisme voor skills, en scripts worden zonder sandboxing uitgevoerd op de lokale machine, wat security-aandacht vereist. Daarnaast ontbreekt een standaard human-in-the-loop-mechanisme, waardoor agents skills en scripts automatisch aanroepen kunnen doen zonder expliciete goedkeuring. Dat grote spelers dit in gebruik nemen is bemoedigend, maar best practices voor het organiseren en onderhouden van skills op schaal moeten zich nog bewijzen. Om deze redenen plaatsen wij Agent Skills op Assess.

## OpenClaw

### HOLD

**OpenClaw** is een lokaal draaiende, autonome AI-agent die direct acties op een besturingssysteem uitvoert. Via de marktplaats **ClawHub** is de functionaliteit van de agent uit te breiden met zogeheten 'skills'. Deze skills omvatten diverse functionaliteiten, zoals het aanroepen van externe API's en het uitvoeren van prompts.

De directe systeemtoegang van de agent brengt echter risico's met zich mee, en in combinatie met een publieke marktplaats ontstaat hier een supply-chain risico vergelijkbaar met `npm` of `PyPI`. Op de marktplaats was er in eerste instantie geen controle op malafide skills, met als gevolg kwaadaardige scripts die zich richtten op lokale privédata. Inmiddels scant en blokkeert een nieuwe VirusTotal-integratie nu automatisch schadelijke code. Toch leunt de beveiliging nog te zwaar op systeem-prompts om het gedrag van de agent in te perken. Omdat een taalmodel gevoelig is voor misleiding via onbetrouwbare invoer, vormen tekstinstructies geen harde beveiligingsgrens. Wanneer de output van het model direct wordt omgezet in uitvoerbare commando's, blijft het risico op schadelijke prompt-injecties gewoon te groot.

Binnen ons vakgebied hechten we veel waarde aan de verantwoorde inzet van dit soort nieuwe technologie. Hoewel het veelbelovend is en recente updates helpen, blijft de standaardconfiguratie risicovol. Het lokaal draaien van deze software zonder de juiste isolatie is eigenlijk onverantwoord vanuit een security-perspectief. We vergelijken dit effectief met het uitvoeren van onbetrouwbare externe code via bijvoorbeeld een scriptrunner met volledige root-rechten. Het platform is werkbaar onder strikte voorwaarden, zoals strakke netwerkisolatie binnen een virtuele machine met minimale permissies. Totdat de standaardimplementaties van dit soort platforms harde beveiligingsgrenzen afdwingen, zetten wij OpenClaw momenteel op 'Hold' voor professioneel gebruik.



## IntelliJ Spring Debugger

### TRIAL

De **IntelliJ Spring Debugger** is een plugin beschikbaar voor IntelliJ IDEA Ultimate. De plugin is primair gericht op Spring Boot projecten; voor plain Spring zonder Boot kunnen de ervaringen wisselvallig zijn. Het tilt Spring debugging naar een heel ander niveau dan standaard debugging met breakpoints.

Tijdens een debugsessie geeft de Spring Debugger je toegang tot de volledige application context. Alle actieve beans zijn inzichtelijk, ook buiten de scope van je huidige breakpoint. Je kunt methods aanroepen op beans die elders in de applicatie leven. Zo kun je onder andere caches invalideren, direct queries uitvoeren via je repositories en tabelinhoud inzien of on-the-fly muteren, zonder de sessie te onderbreken. Het maakt de Spring *magie* inzichtelijker, wat met name de minder ervaren Spring developer zal helpen.

Daarnaast zijn omgevingsvariabelen zichtbaar precies zoals de applicatie ze heeft ingelezen. Je krijgt meer inzicht in hoe beans geïnstantieerd worden en waar hun waarden vandaan komen. Dat maakt de internals van Spring een stuk transparanter en concreter, zeker voor ontwikkelaars die minder vertrouwd zijn met het framework.

Als je dagelijks met Spring Boot werkt, is dit een tool die je zeker moet uitproberen. De hoeveelheid informatie die je tijdens een debugsessie tot je beschikking hebt, is aanzienlijk groter dan wat standaard IntelliJ biedt. Het leidt niet tot merkbare vertraging of andere negatieve neveneffecten.

## Maven 4

### ASSESS

**Maven 4** bevindt zich momenteel in Release Candidate-stadium en brengt significante verbeteringen voor Java-projecten. De belangrijkste winst zit in de prestaties: parallelle dependency resolution en verbeterde caching kunnen build-tijden significant verkorten, vooral merkbaar bij grote multi-module-projecten. Daarnaast introduceert Maven 4 de scheiding tussen build en consumer POM's, waardoor ontwikkelaars flexibeler configuraties kunnen gebruiken zonder afhankelijke projecten te beïnvloeden.

De verbeterde foutmeldingen en automatische parentversie-inferentie maken het werken met Maven intuïtiever. Voor teams die werken met moderne CI/CD-pipelines zijn de betere integratiemogelijkheden met container-based builds en cloud-native omgevingen een welkome toevoeging. Maven 4 introduceert een nieuw dedicated bom packaging-type met ondersteuning voor exclusions en classifiers, wat dependency management in multi-module projecten vereenvoudigt. De langverwachte ingebouwde ondersteuning voor CI-friendly versioning met `${revision}+`, `${sha1}+` en `${changelist}+` properties werkt nu zonder de Flatten Maven Plugin, samen met automatische parent- en subproject-versionering.

De lifecycle is veranderd van een graph- naar een tree-structuur met nieuwe phase types zoals `before:all` en `after:all` voor meer granulaire controle. Profile management is flexibeler geworden met optionele profiles die een `?` prefix gebruiken, waardoor builds niet meer breken als een profile niet bestaat. Deze verbeteringen maken Maven 4 significant gebruiksvriendelijker voor moderne development workflows. Er zijn wel enkele breaking changes, met name rond plugin-compatibiliteit en de strengere POM-validatie, waardoor een zorgvuldige migratie nodig is. Merk ook op dat Java 17 de minimale vereiste is voor Maven 4.

We verwachten dat wanneer Maven 4 GA (General Availability) bereikt, het een solide keuze zal zijn voor projecten, gezien de significante prestatieverbeteringen en de verbeterde developer experience. Daarom is het zaak nu al te kijken naar wat er bij een migratie komt kijken. Gezien de significante prestatieverbeteringen en de verbeterde developer experience plaatsen wij Maven 4 in de Assess ring.

# Danger

## ASSESS

**Danger** is een opensource automatiseringstool die draait tijdens CI/CD om code review-standaarden af te dwingen. De tool automatiseert repetitieve code review-taken door pull requests te controleren tegen vooraf gedefinieerde regels. Danger draait als een extra stap in je pipeline en integreert met platforms zoals GitHub, GitLab en Bitbucket. De tool ondersteunt meerdere talen waaronder Ruby, JavaScript, Swift, Python en sinds 2022 ook Kotlin.

Teams kunnen Danger configureren om changelog-entries te verifiëren, issue tracker-referenties te vereisen in PR-omschrijvingen of architecturale anti-patterns te signaleren. Hiermee detecteert Danger veel voorkomende omissies, op basis van door het team in code gedefinieerde standaarden, automatisch.

Hoewel Danger al bestaat sinds 2016, maakt de in 2022 toegevoegde ondersteuning voor Kotlin het opnieuw relevant voor het Java- en Kotlin-ecosysteem. JDriven ziet dat organisaties Danger succesvol inzetten om hun code review-proces te standaardiseren. De tool helpt teams om reviewnormen vast te leggen en consistentie te behouden over pull requests heen. We plaatsen Danger in *Trial*: de tool is volwassen, maar Kotlin-support is relatief recent en teams moeten valideren hoe het past in hun specifieke workflow en reviewproces.

# Pulumi

## ASSESS

**Pulumi** is een opensource infrastructure-as-code-tool (IaC-tool) waarmee je cloudinfrastructuur definieert in algemene programmeertalen zoals TypeScript, Python, Go, .NET, Java en desgewenst YAML.

De opensource IaC-tool staat los van het commerciële Pulumi Cloud-platform, al kan dat platform wel worden gebruikt voor teamsamenwerking en als backend voor het opslaan van de infrastructuurstatus. Wie Pulumi zonder Pulumi Cloud wil gebruiken, kan state ook opslaan in een eigen backend, zoals S3, Azure Blob Storage, Google Cloud Storage, PostgreSQL of het lokale filesysteem.

Voor development teams is Pulumi interessant omdat infrastructuurcode met dezelfde taal, libraries en ontwikkeltools kan worden geschreven als applicatiecode. Dat verlaagt de drempel voor developers om infrastructurele wijzigingen mee te ontwikkelen en maakt hergebruik, testing en abstrahering mogelijk met bekende programmeerconstructies.

Algemene programmeertalen maken het eenvoudig om nette abstraheringen te bouwen, maar ook om onnodig complexe deployment logica te introduceren en teams moeten bewuste keuzes maken rond statebeheer, secrets, governance en samenwerking. Voor organisaties die al zwaar op Terraform of OpenTofu leunen betekent Pulumi bovendien een extra IaC-model en ecosysteem om te beheren.

Daarom plaatsen we Pulumi in Assess: het is wellicht volwassen genoeg om op echte projecten te beproeven en kan developers productiever maken, maar we adviseren eerst te verkennen wat het vereist aan team afspraken, governance en statebeheer alvorens het in te zetten.



## Devbox

### ASSESS

**Devbox** is een command-line tool om reproduceerbare ontwikkelomgevingen met projectspecifieke tools en versies te definiëren.

Met Devbox leg je in versiebeheer vast welke tools een project nodig heeft, zoals Node, Python, Java of Pandoc, waarna dezelfde omgeving lokaal en in CI (simpelere build pipelines) gebruikt kan worden. Het voorkomt ook de "het werkt op mijn machine"-issues. Daarmee is het een lichter alternatief voor volledig gecontaineriseerde ontwikkelomgevingen. Devbox draait op Nix en moet niet verward worden met Microsoft Dev Box.

Wij plaatsen Devbox in Assess. Het biedt duidelijke voordelen voor reproduceerbare ontwikkelomgevingen, maar de Nix-basis en de bijbehorende leercurve maken het verstandig om eerst ervaring op te doen voordat het breed wordt ingezet.

## Kro

### ASSESS

Kube Resource Orchestrator (**KRO**) is een opensource, Kubernetes-native project bedoeld om complexe definities van resources te beheren middels Resource Graph Definitions (**RGD's**). Deze RGD's zijn templates waarmee daadwerkelijke resource-groepen eenvoudig en uniform beheerd en uitgerold kunnen worden.

**kro** vereenvoudigt beheer van k8s resources door simpeler en centraal definities te aggregeren; een brug tussen complexiteit en ontwikkelaarvriendelijke self-service, zonder verdere banden met een specifieke leverancier. Waar **helm** distributie van definities makkelijker en beheersbaar maakt, zorgt **kro** voor een lokale kant van beheer van complexe resources.

## Pulp

### ASSESS

**Pulp** is een gratis self-hosted opensource container- en artifact-repository. Het is uit te rollen als enkele container, of in meerdere containers via podman of docker compose. Bovendien is er Pulp Operator, waarmee Pulp horizontaal schaalbaar is uit te rollen in een Kubernetes cluster. Net als Nexus en Artifactory is Pulp te gebruiken als een proxy voor publieke container en artifact repositories. Het is niet zo uitgebreid als die producten; zo is de *pull through cache* voor NPM nog in tech review en heeft het weliswaar een vulnerability reporter voor Python packages, maar niet voor andere repositories. Pulp biedt wel de mogelijkheid om zelf plugins toe te voegen voor additionele vulnerability reporting.

We zien dat organisaties worstelen met oplopende licentiekosten voor Nexus en Artifactory. Ook laat de beheersbaarheid van deze producten soms te wensen over. JDriven ziet dat Pulp een interessant alternatief kan zijn, maar waarschuwt dat het qua features mogelijk achterblijft en dat alhoewel het gratis is om te gebruiken, het qua onderhoudbaarheid wel kosten met zich meebrengt.

## Kepler

### ASSESS

**Kepler** is een Prometheus exporter die energieverbruik in Kubernetes van containers, pods en nodes kan meten.

Door te meten op basis van daadwerkelijk resourcegebruik en hardware-sensoren geeft Kepler concretere signalen voor performance, capaciteit en energieverbruik dan statische analyses of code scanners.

Wij plaatsen Kepler in Assess. Het is een interessante aanvulling voor teams die sustainability en efficiency meetbaar willen maken in Kubernetes, maar als CNCF sandbox project is het nog niet klaar voor brede toepassing.

## Jujutsu

### ASSESS

**Jujutsu** is een versiebeheersysteem (VCS) dat een eigen user interface biedt bovenop bestaande storage systems van andere versiebeheersystemen. Het is gestart als project binnen Google en een aantal mensen binnen Google werken er ook aan, maar Google is niet de maintainer. Jujutsu wil een makkelijkere manier bieden voor ontwikkelaars om hun dagelijks bezigheden met VCS te doen. Het is geheel compatibel met Git, wat momenteel het enige geïmplementeerde achterliggende storage system is. Qua features heeft Jujutsu het een en ander afgekeken van Git, Mercurial, Sapling en Darcs zoals snelheid, anonieme branches, geavanceerde merge conflict resoluties. Daarnaast ziet het iedere change automatisch als een commit, houdt het een uitgebreide log bij van acties op de repository en maakt een undo eenvoudig.

Jujutsu wordt actief doorontwikkeld, maar zit nog niet op een 1.0 versie en noemt zichzelf nog experimenteel. Het is zeker iets om in de gaten te houden en daarom staat het in Assess op de radar, want het combineren van de krachten van diverse versiebeheersystemen in een eenvoudigere user interface zonder van Git af te hoeven stappen klinkt veelbelovend.

## Anubis

### ASSESS

Anubis is een softwareprogramma dat een op JavaScript gebaseerde proof-of-work challenge aan websites toevoegt voordat gebruikers deze kunnen bezoeken. In principe kunnen alleen moderne browsers die proof-of-work challenge uitvoeren en fungeert dit dus als een alternatief voor CAPTCHA's, waar geen gebruikersinteractie voor nodig is. Het is ontworpen om het internet te beschermen tegen de eindeloze stroom requests van AI-bedrijven. Een scraperbot kan standaard geen proof-of-work challenge uitvoeren en dus geen zinvolle verbinding opzetten. Anubis fungeert zodoende als een zogenaamde 'tar pit': het remt met name scrapers af, die data zoeken voor AI-training.

Anubis is een jong project en het is afwachten of deze aanpak standhoudt; scrapers kunnen erop aangepast worden. Maar het aanpassen van een scraper, zodanig dat hij de proof-of-work challenge wel kan verrichten, zou hem langzamer maken, waardoor hij minder requests kan versturen. Omdat het toepassen van Anubis een relatief kleine investering is, achten we het de moeite waard om te proberen. Van belang is om Anubis te configureren zodat bots van bijvoorbeeld zoekmachines doorgelaten worden, indien gewenst. De standaardafbeelding van Anubis, die bezoekers te zien krijgen bij een bezoek, zal niet bij het thema van alle websites passen, maar die kan aangepast worden.



## Matrix

### ASSESS

**Matrix** is een open protocol voor veilige, gedecentraliseerde instant messaging.

Matrix ondersteunt federation, end-to-end encryptie en bridges naar andere chatplatforms. Het wordt al gebruikt in Frankrijk bij de overheid met Tchap en in Duitsland bij de Bundeswehr en zorginstellingen. Hierdoor is het een interessant alternatief voor organisaties die meer controle willen over hun communicatie-infrastructuur.

Wij plaatsen Matrix in Assess. Het protocol is sterk voor open en soevereine communicatie, maar operationele complexiteit en wisselende kwaliteit van client software maken het nog geen vanzelfsprekende keuze voor brede inzet.

## Mattermost

### TRIAL

**Mattermost** is een opensource online chatdienst die zelf te hosten is. De dienst is ontworpen als interne chat voor organisaties en bedrijven, en positioneert zich voornamelijk als alternatief voor Slack en Microsoft Teams.

De laatste versies ondersteunen bellen, videobellen en scherm delen. Er is een desktopapplicatie voor Windows, macOS en Linux. Ook op je Android- of iOS-telefoon is er een app beschikbaar. We zien dat grote partijen Mattermost al enige tijd naar tevredenheid inzetten om developers, beheerders en gebruikers met elkaar te laten communiceren.

Hierbij moet wel worden opgemerkt dat **Mattermost** recentelijk de licentievoorwaarden restrictiever heeft gemaakt, wat vooral de functionaliteit en inzetbaarheid van de gratis zelf te hosten variant heeft ingeperkt. Het zet de toon voor waar Mattermost als bedrijf heen wil met de dienst. Zeker als het gratis zelf hosten een argument is om het in te zetten, is het aan te raden de voorwaarden goed door te nemen.

## Digital Sovereignty Readiness Assessment

### ASSESS

De **Digital Sovereignty Readiness Assessment** is een opensource en vendor-neutraal gereedschap om organisaties te helpen bepalen hoe digitaal soeverein ze zijn en welke stappen ze kunnen overwegen om eventueel meer soeverein te worden.

Red Hat heeft de Digital Sovereignty Readiness Assessment uitgegeven onder de Apache 2.0-licentie en beschikbaar gemaakt op GitHub. Het kan lokaal geïnstalleerd worden.

In ongeveer 10-15 minuten genereert deze tool professionele rapporten op basis van je antwoorden op 21 vragen in de vorm van Ja/Nee/"Weet niet". Het rapport behandelt zaken zoals je data, technische en operationele soevereiniteit, opensourcestrategie, managed services en directietoezicht. De tool berekent realtime de volwassenheidsniveaus van je organisatie en geeft concrete aanbevelingen op basis van je beoordelingsresultaten. Daarnaast biedt de tool onderzoeksvragen voor nader onderzoek naar je "Weet niet"-antwoorden.

Wij zijn van mening dat de mate waarin een organisatie afhankelijk wil zijn van externe partijen een strategisch en tactisch beleidsvraagstuk is. We plaatsen deze Digital Sovereignty Readiness Assessment daarom in Assess, aangezien het een van de eerste heldere gereedschappen is die wij kennen om dit vraagstuk te helpen beantwoorden.

# BlueDolphin

## ASSESS

BlueDolphin is een platform voor enterprise-architectuur en procesmodellering. BlueDolphin biedt een set aan functionaliteiten voor het visualiseren van bedrijfsstructuren, applicatielandschappen en informatiestromen. De kracht van BlueDolphin zit in het ondersteunen van samenwerking tussen verschillende rollen binnen een organisatie en stelt teams in staat om zowel de huidige als de gewenste architectuur visueel en gestructureerd vorm te geven. Dit vergroot het inzicht in de impact van veranderingen en helpt bij het onderbouwen van architectuurkeuzes.

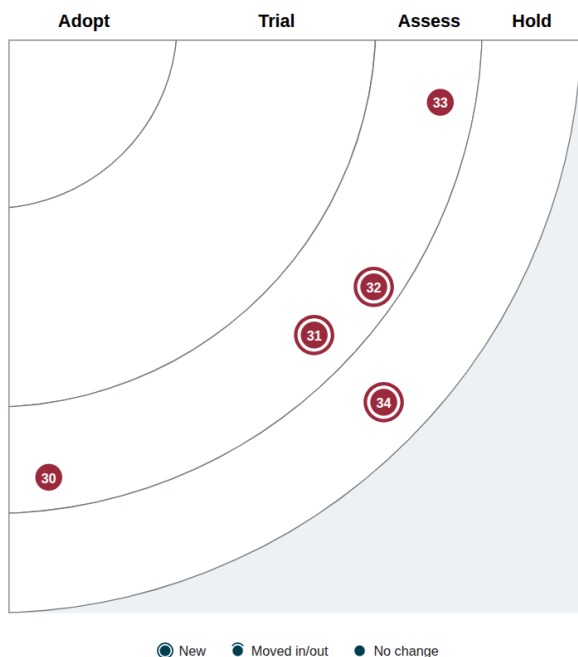
Wel brengt de veelheid aan mogelijkheden ook een keerzijde met zich mee. Door de grote hoeveelheid opties en configuratiemogelijkheden is het voor nieuwe gebruikers, en soms zelfs voor ervaren gebruikers, makkelijk om de weg kwijt te raken. De leercurve is steil, en zonder duidelijke richtlijnen of governance kan het gebruik van BlueDolphin snel ongestructureerd of inconsistent worden.

Hoewel BlueDolphin potentie heeft om organisaties te helpen bij het structureren en communiceren van hun architectuur, vereist het een duidelijke visie en begeleiding om effectief ingezet te worden. Het is daarom belangrijk om vooraf goed na te denken over het doel, scope en de opleidingsbehoeftes van het gebruik binnen de organisatie.

Daarom wordt BlueDolphin in het kwadrant Assess geplaatst van deze editie van onze Tech Radar.



## Talen & frameworks



### ASSESS

- 30. Amber
- 31. Arconia
- 32. Embabel Agent Framework
- 33. Koog

### HOLD

- 34. Scala

## Embabel Agent Framework

### ASSESS

**Embabel** is een opensource AI-agentframework voor de JVM, gecreëerd door Rod Johnson, de grondlegger van het Spring Framework. Het stelt Java- en Kotlin-developers in staat om agentic flows te bouwen die LLM-interacties naadloos combineren met getypeerde domeinmodellen en bestaande applicatiecode. Een onderscheidend kenmerk is het gebruik van Goal-Oriented Action Planning (GOAP), een planningsalgoritme uit de gamesindustrie, waarmee agents dynamisch stappen combineren om doelen te bereiken.

Vanuit JDriven zien wij Embabel als een veelbelovende ontwikkeling voor het Java-ecosysteem. Het brengt AI-agentcapabilities naar de JVM op een manier die aansluit bij hoe Java-teams gewend zijn te werken: met sterke typing, compile-time checking en naadloze integratie in het Spring-ecosysteem. Dit verlaagt de drempel om GenAI te integreren in bestaande enterprise-applicaties, zonder over te hoeven stappen naar een Python-stack. De GOAP-aanpak biedt voordelen ten opzichte van frameworks die volledig op LLM's leunen voor planning: het is efficiënter, voorspelbaarder en beter testbaar.

Wel is Embabel een jong project dat zich nog moet bewijzen in productieomgevingen. De community is klein vergeleken met alternatieven zoals LangChain4j en Spring AI, waar Embabel overigens bovenop bouwt als hogere abstractielaag. Daarnaast vereist het begrip van concepten als GOAP en agentic flows, wat een leercurve met zich meebrengt. Om deze redenen plaatsen wij Embabel op Assess.

## Arconia

### ASSESS

**Arconia** is een add-on framework voor Spring Boot dat cloud-native capabilities toevoegt zonder van framework te wisselen.

Het framework richt zich op developer experience met onder andere Dev Services voor lokaal ontwikkelen en testen, Kubernetes-ondersteuning, service binding, observability en multi-tenancy. Dat maakt het interessant voor teams die al Spring Boot gebruiken en minder libraries en maatwerkconfiguraties willen combineren.

Wij plaatsen Arconia in Assess. Het framework is veelbelovend, maar werkt nog toe naar een eerste stabiele release begin 2026 en is daarom nog niet klaar voor brede inzetbaarheid. Daarbij komt ook dat het project voor nu door één developer is gestart en het afwachten is wat voor support er komt.

## Koog

### ASSESS

JetBrains heeft met **Koog** een opensource-framework gelanceerd, dat een developer helpt diverse uitdagingen rondom het maken van AI-agents het hoofd te bieden. Met een AI-agent bedoelen we in dezen: Een onafhankelijke applicatie die los gedefinieerde input accepteert, die naar een LLM stuurt ter interpretatie en het in beginsel non-deterministische antwoord als output teruggeeft. Een applicatie die op deze manier één of meerdere LLM-calls doet, kent een aantal uitdagingen op het gebied van voorspelbaarheid en daarmee herbruikbaarheid en betrouwbaarheid. Koog biedt diverse mechanismes om hiermee om te gaan, zoals de mogelijkheid om workflows op te stellen, conversaties te onthouden, historie te comprimeren en zodoende een kennisbank op te bouwen waaruit een AI-agent kan putten i.p.v. telkens een LLM aan te roepen. Daarbij kan het binnen een conversatie naar een andere LLM overstappen, om bijvoorbeeld te zorgen dat een duurdere of tragere LLM alleen wordt aangeroepen als een snellere, goedkopere geen wenselijk antwoord weet te leveren. Koog is gericht op Kotlin en ondersteunt o.a. OpenTelemetry, Model Context Protocol (MCP), Spring Boot en Ktor.

JDriven ziet in Koog een framework dat helpt de grote risico's van onvoorspelbare uitkomsten en mogelijk hoge kosten van een LLM te mitigeren tot een punt waar je je kunt afvragen of een applicatie, met een dusdanig veilig ingebakerd AI-component, niet simpeler te bouwen zou zijn zonder dat AI-component. Maar juist daar, waar developers nog terughoudend zijn om AI in hun applicaties te introduceren en de brug zoeken naar een solution-architectuur waarin non-deterministische componenten tot hun recht komen, is die veiligheid nodig. We verwachten dat Koog daar een rol van betekenis kan spelen, al moet het daarbij concurreren met LangChain4j en Spring AI.



## Amber

### ASSESS

**Amber** is een moderne programmeertaal die compileert naar Bash shell-scripts. Het pakt de problemen aan van foutgevoelige en lastig te onderhouden shell-scripts door compile-time type checking, een moderne syntaxis vergelijkbaar met ECMAScript en de toevoeging van een uitgebreide standaardbibliotheek. Dit maakt shell-scripts betrouwbaarder en helpt runtime-fouten te voorkomen. Het project won in 2024 de prijs voor "The Best Engineering Project in Technology" aan de Universiteit van Wrocław. De taal ondersteunt vijf datatypes (`Text`, `Num`, `Bool`, `Null` en ``Array`s`), biedt moderne programmeerconstructen zoals functies, loops en condities, en integreert naadloos met bestaande Bash-scripts.

Amber bevindt zich momenteel in versie 0.4.0-alpha en is nog volop in ontwikkeling. Hoewel het concept veelbelovend is voor het verbeteren van shell-scripting, zijn er nog beperkingen, zoals het ontbreken van ondersteuning voor geneste arrays vanwege de beperkingen van Bash zelf, en het risico op incompatibele wijzigingen tussen versies. De taal is interessant om te volgen voor teams die veel gebruikmaken van shell-scripts, maar is nog niet rijp voor productie. Daarom plaatsen we Amber in de Assess-ring.

## Scala

### HOLD

Scala is ontstaan uit een behoefte om principes van functioneel programmeren te kunnen toepassen binnen het JVM ecosysteem. Het faciliteert ook objectgeoriënteerd programmeren en breidt de standaard features van Java op dat gebied uit, zoals bijv. via extension methods en operator overloading.

In 2020 plaatsten we Scala nog op Adopt ondanks de steile leercurve van Scala en de Scala Build Toolkit (sbt). De reden daarvoor was dat versie 3 uitkwam, die de taal niet alleen rijker maakte, maar ook gericht was op het vereenvoudigen ervan. Sindsdien zien we de vraag naar Scala in de markt echter significant afnemen, ten gunste van met name Kotlin. Expertise is lastig te vinden en is bovendien vaak specifiek gericht op ZIO, Cats of Akka, wat niet uitwisselbaar is. Een Scala-project is daardoor moeilijker en daarmee duurder te bemannen. Aangezien er een sterk alternatief als Kotlin bestaat en Java zelf ook is doorontwikkeld, kunnen wij Scala niet meer aanbevelen voor nieuwe projecten.



**Commit.**  
**Develop.**  
**Share.**